User's Manual

32

Renesas RA2T1 Group

User's Manual: Hardware

32-Bit MCU
Renesas Advanced (RA) Family
Renesas RA2 Series

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

Renesas Electronics
www.renesas.com

Rev.1.00 Jun 2025

Notice

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Contact Information

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

Trademarks

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

© 2025 Renesas Electronics Corporation. All rights reserved.

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

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

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

Preface

1. About this document

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

2. Audience

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

3. Renesas Publications

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

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

4. Numbering Notation

The following numbering notation is used throughout this manual:

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

5. Typographic Notation

The following typographic notation is used throughout this manual:

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

6. Unit and Unit Prefix

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

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

7. Special Terms

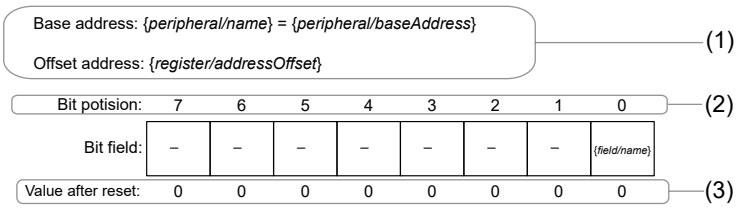
The following terms have special meanings.

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

8. Register Description

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

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



(4) (5) (6)

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

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

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

(2) Bit number

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

(3) Value after reset

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

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

(4) Symbol

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

(5) Function

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

(6) R/W

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

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

9. Abbreviations

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

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

10. Proprietary Notice

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

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

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

SuperFlash® is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

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

11. Feedback on the product

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

Contents

Features **32**
1. Overview **33**
1.1 Function Outline 33
1.2 Block Diagram 37
1.3 Part Numbering 37
1.4 Function Comparison 40
1.5 Pin Functions 41
1.6 Pin Assignments 44
1.7 Pin Lists 47
2. CPU **49**
2.1 Overview 49
2.1.1 CPU 49
2.1.2 Debug 49
2.1.3 Operating Frequency 49
2.1.4 Block Diagram 49
2.2 Implementation Options 50
2.3 SWD Interface 51
2.4 Debug Function 51
2.4.1 Debug Mode Definition 51
2.4.2 Debug Mode Effects 51
2.5 Programmers Model 52
2.5.1 Address Spaces 52
2.5.2 Cortex-M23 Peripheral Address Map 53
2.5.3 External Debug Address Map 53
2.5.4 CoreSight ROM Table 53
2.5.5 DBGREG Module 54
2.5.6 OCDREG Module 56
2.6 SysTick Timer 58
2.7 OCD Emulator Connection 58
2.7.1 Unlock ID Code 59
2.7.2 DBGEN 59
2.7.3 Restrictions on Connecting an OCD emulator 59
2.8 References 61
2.9 Usage Notes 61
3. Operating Modes **62**
3.1 Operating Mode Types and Selection 62
3.2 Details of Operating Modes 62
3.2.1 Single-Chip Mode 62
3.2.2 SCI Boot Mode 62
3.3 Operating Modes Transitions 62
3.3.1 Operating Mode Transitions as Determined by the Mode-Setting Pin 62
4. Address Space **63**
4.1 Address Space 63
5. Resets **64**

5.1 Overview 64
5.2 Register Descriptions 68
5.2.1 RSTSR0 : Reset Status Register 0 68
5.2.2 RSTSR1 : Reset Status Register 1 69
5.2.3 RSTSR2 : Reset Status Register 2 71
5.3 Operation 72
5.3.1 RES Pin Reset 72
5.3.2 Power-On Reset 72
5.3.3 Voltage Monitor Reset 73
5.3.4 Independent Watchdog Timer Reset 74
5.3.5 Watchdog Timer Reset 74
5.3.6 Software Reset 75
5.3.7 Determination of Cold/Warm Start 75
5.3.8 Determination of Reset Generation Source 75
5.4 Usage Notes 76
5.4.1 Note on RES pin reset 76
6. Option-Setting Memory 77
6.1 Overview 77
6.2 Register Descriptions 77
6.2.1 OFS0 : Option Function Select Register 0 77
6.2.2 OFS1 : Option Function Select Register 1 81
6.2.3 MPU Registers 82
6.2.4 AWS : Access Window Setting Register 83
6.2.5 OSIS : OCD/Serial Programmer ID Setting Register 84
6.3 Setting Option-Setting Memory 85
6.3.1 Allocation of Data in Option-Setting Memory 85
6.3.2 Setting Data for Programming Option-Setting Memory 85
6.4 Usage Notes 86
6.4.1 Data for Programming Reserved Areas and Reserved Bits in the Option-Setting Memory 86
6.4.2 Note on FSPR Bit 86
7. Low Voltage Detection (LVD) 87
7.1 Overview 87
7.2 Register Descriptions 89
7.2.1 LVCMPCR : Voltage Monitor Circuit Control Register 89
7.2.2 LVDLVLR : Voltage Detection Level Select Register 90
7.2.3 LVD1CR0 : Voltage Monitor 1 Circuit Control Register 0 90
7.2.4 LVD2CR0 : Voltage Monitor 2 Circuit Control Register 0 91
7.2.5 LVD1CR1 : Voltage Monitor 1 Circuit Control Register 1 92
7.2.6 LVD1SR : Voltage Monitor 1 Circuit Status Register 92
7.2.7 LVD2CR1 : Voltage Monitor 2 Circuit Control Register 1 93
7.2.8 LVD2SR : Voltage Monitor 2 Circuit Status Register 93
7.3 VCC Input Voltage Monitor 94
7.3.1 Monitoring V_{det0} 94
7.3.2 Monitoring V_{det1} 94
7.3.3 Monitoring V_{det2} 94
7.4 Reset from Voltage Monitor 0 95

7.5 Interrupt and Reset from Voltage Monitor 1 95
7.6 Interrupt and Reset from Voltage Monitor 2 97
7.7 Event Link Controller (ELC) Output 99
7.7.1 Interrupt Handling and Event Linking 99
8. Clock Generation Circuit 101
8.1 Overview 101
8.2 Register Descriptions 104
8.2.1 SCKDIVCR : System Clock Division Control Register 104
8.2.2 SCKSCR : System Clock Source Control Register 105
8.2.3 MEMWAIT : Memory Wait Cycle Control Register for Code Flash 105
8.2.4 FLDWAITR : Memory Wait Cycle Control Register for Data Flash 106
8.2.5 MOSCCR : Main Clock Oscillator Control Register 109
8.2.6 LOCOCR : Low-Speed On-Chip Oscillator Control Register 110
8.2.7 HOCOCR : High-Speed On-Chip Oscillator Control Register 111
8.2.8 MOCOCR : Middle-Speed On-Chip Oscillator Control Register 112
8.2.9 OSCSF : Oscillation Stabilization Flag Register 112
8.2.10 OSTDCR : Oscillation Stop Detection Control Register 113
8.2.11 OSTDSR : Oscillation Stop Detection Status Register 114
8.2.12 MOSCWTCR : Main Clock Oscillator Wait Control Register 115
8.2.13 HOCOWTCR : High-Speed On-Chip Oscillator Wait Control Register 116
8.2.14 MOMCR : Main Clock Oscillator Mode Oscillation Control Register 116
8.2.15 CKOCR : Clock Out Control Register 117
8.2.16 LOCOUTCR : LOCO User Trimming Control Register 118
8.2.17 MOCOUTCR : MOCO User Trimming Control Register 118
8.2.18 HOCOUTCR : HOCO User Trimming Control Register 119
8.3 Main Clock Oscillator 119
8.3.1 Connecting a Crystal Resonator 119
8.3.2 External Clock Input 120
8.3.3 Notes on External Clock Input 121
8.4 Oscillation Stop Detection Function 121
8.4.1 Oscillation Stop Detection and Operation after Detection 121
8.4.2 Oscillation Stop Detection Interrupts 122
8.5 Internal Clock 123
8.5.1 System Clock (ICLK) 123
8.5.2 Peripheral Module Clock (PCLKB, PCLKD) 124
8.5.3 CAC Clock (CACCLK) 124
8.5.4 IWDT-Dedicated Clock (IWDTCLK) 124
8.5.5 AGTW-Dedicated Clock (AGTLCLK) 124
8.5.6 SysTick Timer-Dedicated Clock (SYSTICCLK) 125
8.5.7 External Pin Output Clock (CLKOUT) 125
8.6 Usage Notes 125
8.6.1 Notes on Clock Generation Circuit 125
8.6.2 Notes on Board Design 125
8.6.3 Notes on Resonator Connect Pin 126
9. Clock Frequency Accuracy Measurement Circuit (CAC) 127
9.1 Overview 127

9.2 Register Descriptions 128
9.2.1 CACR0 : CAC Control Register 0 128
9.2.2 CACR1 : CAC Control Register 1 129
9.2.3 CACR2 : CAC Control Register 2 129
9.2.4 CAICR : CAC Interrupt Control Register 130
9.2.5 CASTR : CAC Status Register 131
9.2.6 CAULVR : CAC Upper-Limit Value Setting Register 132
9.2.7 CALLVR : CAC Lower-Limit Value Setting Register 132
9.2.8 CACNTBR : CAC Counter Buffer Register 133
9.3 Operation 133
9.3.1 Measuring Clock Frequency 133
9.3.2 Digital Filtering of Signals on CACREF Pin 135
9.4 Interrupt Requests 135
9.5 Usage Notes 135
9.5.1 Settings for the Module-Stop Function 135
10. Low Power Modes 136
10.1 Overview 136
10.2 Register Descriptions 139
10.2.1 SBYCR : Standby Control Register 139
10.2.2 MSTPCRA : Module Stop Control Register A 140
10.2.3 MSTPCRB : Module Stop Control Register B 140
10.2.4 MSTPCRC : Module Stop Control Register C 141
10.2.5 MSTPCRD : Module Stop Control Register D 142
10.2.6 OPCCR : Operating Power Control Register 143
10.2.7 SOPCCR : Sub Operating Power Control Register 143
10.2.8 SNZCR : Snooze Control Register 145
10.2.9 SNZEDCR0 : Snooze End Control Register 0 145
10.2.10 SNZREQCR0 : Snooze Request Control Register 0 147
10.2.11 SYOCDCR : System Control OCD Control Register 148
10.2.12 LSMRWDIS : Low Speed Module R/W Disable Control Register 148
10.2.13 LPOPT : Lower Power Operation Control Register 149
10.3 Reducing Power Consumption by Switching Clock Signals 150
10.4 Module-Stop Function 150
10.5 Function for Lower Operating Power Consumption 151
10.5.1 Setting Operating Power Control Mode 151
10.5.2 Operating Range 152
10.6 Sleep Mode 154
10.6.1 Transitioning to Sleep Mode 154
10.6.2 Canceling Sleep Mode 154
10.7 Software Standby Mode 155
10.7.1 Transition to Software Standby Mode 155
10.7.2 Canceling Software Standby Mode 155
10.7.3 Example of Software Standby Mode Application 156
10.8 Snooze Mode 157
10.8.1 Transition to Snooze Mode 157
10.8.2 Canceling Snooze Mode 158
10.8.3 Returning from Snooze Mode to Software Standby Mode 159

10.8.4 Snooze Operation Example 160
10.9 Usage Notes 164
10.9.1 Register Access 164
10.9.2 I/O Port pin states 165
10.9.3 Module-Stop State of DTC 165
10.9.4 Internal Interrupt Sources 165
10.9.5 Transitioning to Low Power Modes 165
10.9.6 Timing of WFI Instruction 165
10.9.7 Writing to the WDT/IWDT Registers by DTC in Sleep Mode or Snooze Mode 165
10.9.8 Oscillators in Snooze Mode 165
10.9.9 Snooze Mode Entry by RXD0 Falling Edge 166
10.9.10 Using UART of SCI0 in Snooze Mode 166
10.9.11 Conditions of A/D Conversion Start in Snooze Mode 166
10.9.12 ELC Events in Snooze Mode 166
10.9.13 Module-Stop Function for ADC120 166
10.9.14 Module-Stop Function for an Unused Circuit 166
10.9.15 Notes when Transition to Low Power Modes by Sleep-On-Exit 167
10.9.16 Applicable Condition and Notes 167
10.9.17 Workaround 168
11. Register Write Protection 170
11.1 Overview 170
11.2 Register Descriptions 170
11.2.1 PRCR : Protect Register 170
12. Interrupt Controller Unit (ICU) 171
12.1 Overview 171
12.2 Register Descriptions 172
12.2.1 IRQCRi : IRQ Control Register i (i = 0 to 7) 172
12.2.2 NMISR : Non-Maskable Interrupt Status Register 173
12.2.3 NMIER : Non-Maskable Interrupt Enable Register 176
12.2.4 NMICLR : Non-Maskable Interrupt Status Clear Register 177
12.2.5 NMICR : NMI Pin Interrupt Control Register 179
12.2.6 IELSRn : ICU Event Link Setting Register n (n = 0 to 31) 180
12.2.7 SELSR0 : SYS Event Link Setting Register 181
12.2.8 WUPEN : Wakeup Interrupt Enable Register 181
12.2.9 IELEN : ICU Event Enable Register 183
12.3 Vector Table 183
12.3.1 Interrupt Vector Table 183
12.3.2 Event Number 184
12.3.3 ICU and DTC Event Number 188
12.4 Interrupt Operation 193
12.4.1 Detecting Interrupts 193
12.5 Interrupt setting procedure 194
12.5.1 Enabling Interrupt Requests 194
12.5.2 Disabling Interrupt Requests 194
12.5.3 Polling for interrupts 194
12.5.4 Selecting Interrupt Request Destinations 194

12.5.5 Digital Filter 195
12.5.6 External Pin Interrupts 196
12.6 Non-Maskable Interrupt Operation 196
12.7 Return from Low Power Modes 197
12.7.1 Return from Sleep Mode 197
12.7.2 Return from Software Standby Mode 197
12.7.3 Return from Snooze Mode 197
12.8 Using the WFI Instruction with Non-Maskable Interrupts 198
12.9 Reference 198
13. Buses 199
13.1 Overview 199
13.2 Description of Buses 200
13.2.1 Main Buses 200
13.2.2 Slave Interface 200
13.2.3 Parallel Operations 200
13.2.4 Restriction on Endianness 200
13.2.5 Restriction on Exclusive Access 200
13.3 Register Descriptions 201
13.3.1 BUSMCNTx : Master Bus Control Register x (x = SYS, DMA) 201
13.3.2 BUSnERRADD : Bus Error Address Register n (n = 3, 4) 201
13.3.3 BUSnERRSTAT : Bus Error Status Register n (n = 3, 4) 202
13.4 Bus Error Monitoring Section 202
13.4.1 Error Type that Occurs by Bus 202
13.4.2 Operation when a Bus Error Occurs 203
13.4.3 Conditions for issuing illegal Address Access Errors 203
13.5 References 203
14. Memory Protection Unit (MPU) 204
14.1 Overview 204
14.2 CPU Stack Pointer Monitor 204
14.2.1 Protecting the Registers 207
14.2.2 Overflow and Underflow Errors 207
14.2.3 Register Descriptions 207
14.3 Arm MPU 211
14.4 Bus Master MPU 211
14.4.1 Register Descriptions 212
14.4.2 Operation 216
14.5 Bus Slave MPU 219
14.5.1 Register Descriptions 220
14.5.2 Functions 224
14.6 Security MPU 224
14.6.1 Register Descriptions (Option-Setting Memory) 225
14.6.2 Memory Protection 230
14.7 Usage Notes 231
14.7.1 Notes on the Use of a Debugger 231
14.8 References 231
15. Data Transfer Controller (DTC) 232

15.1 Overview ... 232
15.2 Register Descriptions ... 233
15.2.1 MRA : DTC Mode Register A ... 233
15.2.2 MRB : DTC Mode Register B ... 234
15.2.3 SAR : DTC Transfer Source Register ... 235
15.2.4 DAR : DTC Transfer Destination Register ... 236
15.2.5 CRA : DTC Transfer Count Register A ... 236
15.2.6 CRB : DTC Transfer Count Register B ... 237
15.2.7 DTCCR : DTC Control Register ... 237
15.2.8 DTCVBR : DTC Vector Base Register ... 237
15.2.9 DTCST : DTC Module Start Register ... 238
15.2.10 DTCSTS : DTC Status Register ... 238
15.3 Activation Sources ... 239
15.3.1 Allocating Transfer Information and DTC Vector Table ... 239
15.4 Operation ... 241
15.4.1 Transfer Information Read Skip Function ... 243
15.4.2 Transfer Information Write-Back Skip Function ... 243
15.4.3 Normal Transfer Mode ... 244
15.4.4 Repeat Transfer Mode ... 245
15.4.5 Block Transfer Mode ... 246
15.4.6 Chain Transfer ... 247
15.4.7 Operation Timing ... 248
15.4.8 Execution Cycles of DTC ... 250
15.4.9 DTC Bus Mastership Release Timing ... 251
15.5 DTC Setting Procedure ... 251
15.6 Examples of DTC Usage ... 252
15.6.1 Normal Transfer ... 252
15.6.2 Chain transfer ... 252
15.6.3 Chain Transfer when Counter = 0 ... 254
15.7 Interrupt ... 255
15.7.1 Interrupt Sources ... 255
15.8 Event Link ... 256
15.9 Low Power Consumption Function ... 256
15.10 Usage Notes ... 256
15.10.1 Transfer Information Start Address ... 256
16. Event Link Controller (ELC) ... 257
16.1 Overview ... 257
16.2 Register Descriptions ... 258
16.2.1 ELCR : Event Link Controller Register ... 258
16.2.2 ELSEGRn : Event Link Software Event Generation Register n (n = 0, 1) ... 258
16.2.3 ELSRn : Event Link Setting Register n (n = 0 to 9, 14, 15) ... 259
16.3 Operation ... 262
16.3.1 Relation between Interrupt Handling and Event Linking ... 262
16.3.2 Linking Events ... 263
16.3.3 Example of Procedure for Linking Events ... 263
16.4 Usage Notes ... 263
16.4.1 Linking DTC Transfer End Signals as Events ... 263

16.4.2 Setting Clocks 263
16.4.3 Module-Stop Function Setting 263
16.4.4 ELC Delay Time 263
17. I/O Ports 265
17.1 Overview 265
17.2 Register Descriptions 267
17.2.1 PCNTR1/PODR/PDR : Port Control Register 1 267
17.2.2 PCNTR2/EIDR/PIDR : Port Control Register 2 267
17.2.3 PCNTR3/PORR/POSR : Port Control Register 3 268
17.2.4 PCNTR4/EORR/EOSR : Port Control Register 4 269
17.2.5 PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 5, 9, n = 00 to 15) 270
17.2.6 PWPR : Write-Protect Register 272
17.2.7 PRWCNTR : Port Read Wait Control Register 272
17.3 Operation 273
17.3.1 General I/O Ports 273
17.3.2 Port Function Select 273
17.3.3 Port Group Function for ELC 273
17.3.4 Wait Function for Port Read 275
17.4 Handling of Unused Pins 275
17.5 Usage Notes 276
17.5.1 Procedure for Specifying the Pin Functions 276
17.5.2 Procedure for Using Port Group Input 276
17.5.3 Port Output Data Register (PODR) Summary 276
17.5.4 Notes on Using Analog Functions 277
17.6 Peripheral Select Settings for Each Product 277
18. Key Interrupt Function (KINT) 280
18.1 Overview 280
18.2 Register Descriptions 280
18.2.1 KRCTL : Key Return Control Register 280
18.2.2 KRF : Key Return Flag Register 281
18.2.3 KRM : Key Return Mode Register 281
18.3 Operation 281
18.3.1 Operation When Not Using the Key Interrupt Flags (KRCTL.KRMD = 0) 281
18.3.2 Operation When Using the Key Interrupt Flags (KRCTL.KRMD = 1) 282
18.4 Usage Notes 284
19. Port Output Enable for GPT (POEG) 285
19.1 Overview 285
19.2 Register Descriptions 287
19.2.1 POEGGn : POEG Group n Setting Register (n = A, B) 287
19.2.2 GTONCWPn : GPT Output Stopping Control Group n Write Protection Register (n = A, B) 288
19.2.3 GTONCCRn : GPT Output Stopping Control Group n Controlling Register (n = A, B) 288
19.3 Operation 289
19.3.1 Request to Stop Output in Response to Detection of Input Level on the Corresponding GTETRGn Pin (n = A, B) 289
19.3.2 Requests to Stop Output in Response to Detection of Output Stopping from GPT 290

19.3.3 Request to Stop Output in Response to Comparator Detection 290
19.3.4 Requests to Stop Output by Oscillation Stop Detection 290
19.3.5 Requests to Stop Output by a Register 291
19.3.6 Cancelling Requests to Stop Output 291
19.3.7 Requests to Stop Output in Response to Detected Signals and Cancelling the Requests 292
19.4 Interrupt Sources 294
19.5 External Trigger Output to GPT 295
19.6 Usage Notes 295
19.6.1 Transitions to Low-Power Modes 295
19.6.2 Setting the Function for Stopping the Module 295
19.6.3 Duplication of Requests to Stop Output 295
20. General PWM Timer (GPT) 296
20.1 Overview 296
20.2 Register Descriptions 302
20.2.1 GTWP : General PWM Timer Write-Protection Register 302
20.2.2 GTSTR : General PWM Timer Software Start Register 304
20.2.3 GTSTP : General PWM Timer Software Stop Register 305
20.2.4 GTCLR : General PWM Timer Software Clear Register 305
20.2.5 GTSSR : General PWM Timer Start Source Select Register 306
20.2.6 GTPSR : General PWM Timer Stop Source Select Register 309
20.2.7 GTCSR : General PWM Timer Clear Source Select Register 312
20.2.8 GTUPSR : General PWM Timer Up Count Source Select Register 315
20.2.9 GTDNSR : General PWM Timer Down Count Source Select Register 318
20.2.10 GTICASR : General PWM Timer Input Capture Source Select Register A 322
20.2.11 GTICBSR : General PWM Timer Input Capture Source Select Register B 325
20.2.12 GTCR : General PWM Timer Control Register 328
20.2.13 GTUDDTYC : General PWM Timer Count Direction and Duty Setting Register 330
20.2.14 GTIOR : General PWM Timer I/O Control Register 333
20.2.15 GTINTAD : General PWM Timer Interrupt Output Setting Register 338
20.2.16 GTST : General PWM Timer Status Register 341
20.2.17 GTBER : General PWM Timer Buffer Enable Register 347
20.2.18 GTITC : General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register 350
20.2.19 GTCNT : General PWM Timer Counter 352
20.2.20 GTCCRk : General PWM Timer Compare Capture Register k (k = A to F) 353
20.2.21 GTPR : General PWM Timer Cycle Setting Register 353
20.2.22 GTPBR : General PWM Timer Cycle Setting Buffer Register 353
20.2.23 GTPDBR : General PWM Timer Cycle Setting Double Buffer Register 354
20.2.24 GTADTRk : A/D Conversion Start Request Timing Register k (k = A, B) 354
20.2.25 GTADTBRk : A/D Conversion Start Request Timing Buffer Register k (k = A, B) 355
20.2.26 GTADTDBRk : A/D Conversion Start Request Timing Double Buffer Register k (k = A, B) 355
20.2.27 GTDTCR : General PWM Timer Dead Time Control Register 355
20.2.28 GTDVk : General PWM Timer Dead Time Value Register k (k = U, D) 357
20.2.29 GTDBk : General PWM Timer Dead Time Buffer Register k (k = U, D) 357
20.2.30 GTSOS : General PWM Timer Output Protection Function Status Register 358
20.2.31 GTSOTR : General PWM Timer Output Protection Function Temporary Release Register 358

20.2.32 GTADSMR : General PWM Timer A/D Conversion Start Request Signal Monitoring Register 359
20.2.33 GTEITC : General PWM Timer Extended Interrupt Skipping Counter Control Register 360
20.2.34 GTEITLI1 : General PWM Timer Extended Interrupt Skipping Setting Register 1 362
20.2.35 GTEITLI2 : General PWM Timer Extended Interrupt Skipping Setting Register 2 364
20.2.36 GTEITLB : General PWM Timer Extended Buffer Transfer Skipping Setting Register 365
20.2.37 GTPC : General PWM Timer Period Count Register 367
20.2.38 GTADCMSC : General PWM Timer A/D Conversion Start Request Compare Match Skipping Control Register 368
20.2.39 GTADCMSS : General PWM Timer A/D Conversion Start Request Compare Match Skipping Setting Register 370
20.2.40 GTSECSR : General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register 372
20.2.41 GTSECR : General PWM Timer Operation Enable Bit Simultaneous Control Register 373
20.2.42 GTBER2 : General PWM Timer Buffer Enable Register 2 376
20.2.43 GTOLBR : General PWM Timer Output Level Buffer Register 380
20.2.44 GTICCR : General PWM Timer Inter Channel Cooperation Input Capture Control Register 381
20.2.45 OPSCR : Output Phase Switching Control Register 384
20.3 Operation 387
20.3.1 Basic Operation 387
20.3.2 Buffer Operation 398
20.3.3 PWM Output Operating Mode 416
20.3.4 Automatic Dead Time Setting Function 468
20.3.5 Count Direction Changing Function 475
20.3.6 Function of Output Duty 0% and 100% 476
20.3.7 Hardware Count Start/Count Stop and Clear Operation 478
20.3.8 Synchronized Operation 489
20.3.9 PWM Output Operation Examples 498
20.3.10 Period Count Function 503
20.3.11 Phase Counting Function 504
20.3.12 External Pulse Width Measuring Function 514
20.3.13 Output Phase Switching (GPT_OPS) 515
20.3.14 GTIOCnA/B Input/Output Pin Polarity Reversal Function 522
20.4 Interrupt Sources 522
20.4.1 Interrupt Sources 522
20.4.2 DTC Activation 525
20.4.3 Interrupt and A/D Conversion Start Request Skipping Function 525
20.5 A/D Conversion Start Request 544
20.6 Operations Linked by ELC 547
20.6.1 Event Signal Output to ELC 547
20.6.2 Event Signal Inputs from ELC 547
20.7 Noise Filter Function 548
20.8 Protection Function 548
20.8.1 Write-Protection for Registers 548
20.8.2 Disabling of Buffer Operation 548
20.8.3 GTIOCnm Pin Output Negate Control (n = 0 to 3, m = A, B) 555
20.8.4 Output Protection Function for GTIOCnm Pin Output (n = 0 to 2, m = A, B) 556

20.9 Initialization Method of Output Pins 563
20.9.1 Pin Settings after Reset 563
20.9.2 Pin Initialization Due to Error during Operation 564
20.10 Usage Notes 564
20.10.1 Module-Stop Function Setting 564
20.10.2 GTCCRn Settings During Compare Match Operation (n = A to F) 564
20.10.3 Setting Range for GTCNT Counter 566
20.10.4 Setting Range of GTPBR and GTPDBR in Complementary PWM Mode 566
20.10.5 Starting and Stopping the GTCNT Counter 566
20.10.6 Priority Order of Each Event 566
20.10.7 Counter Clear Operation in Complementary PWM Mode 567
20.10.8 Setting Range of the Compare Match Registers when GTIOR.CPSCIR = 1 567
20.10.9 Prohibit Invalid Register Setting 567
21. Low Power Asynchronous General Purpose Timer (AGTW) 568
21.1 Overview 568
21.2 Register Descriptions 569
21.2.1 AGT : AGT Counter Register 569
21.2.2 AGTCMA : AGT Compare Match A Register 570
21.2.3 AGTCMB : AGT Compare Match B Register 570
21.2.4 AGTCR : AGT Control Register 571
21.2.5 AGTMR1 : AGT Mode Register 1 572
21.2.6 AGTMR2 : AGT Mode Register 2 573
21.2.7 AGTIOC : AGT I/O Control Register 574
21.2.8 AGTISR : AGT Event Pin Select Register 576
21.2.9 AGTCMSR : AGT Compare Match Function Select Register 576
21.2.10 AGTIOSEL : AGT Pin Select Register 577
21.3 Operation 577
21.3.1 Reload Register and Counter Rewrite Operation 577
21.3.2 Reload Register and AGT Compare Match A/B Register Rewrite Operation 579
21.3.3 Timer Mode 580
21.3.4 Pulse Output Mode 581
21.3.5 Event Counter Mode 582
21.3.6 Pulse Width Measurement Mode 583
21.3.7 Pulse Period Measurement Mode 584
21.3.8 Compare Match function 585
21.3.9 Output Settings for Each Mode 586
21.3.10 Standby Mode 588
21.3.11 Interrupt Sources 588
21.3.12 Event Signal Output to ELC 589
21.4 Usage Notes 589
21.4.1 Count Operation Start and Stop Control 589
21.4.2 Access to Counter Register 589
21.4.3 When Changing Mode 589
21.4.4 Output pin setting 590
21.4.5 Digital Filter 590
21.4.6 How to Calculate Event Number, Pulse Width, and Pulse Period 590
21.4.7 When Count is Forcibly Stopped by TSTOP Bit 590

21.4.8 When Selecting AGTW0 Underflow as the Count Source 590
21.4.9 Module-stop function 591
22. Watchdog Timer (WDT) 592
22.1 Overview 592
22.2 Register Descriptions 593
22.2.1 WDTRR : WDT Refresh Register 593
22.2.2 WDTCR : WDT Control Register 594
22.2.3 WDTSR : WDT Status Register 596
22.2.4 WDTRCR : WDT Reset Control Register 597
22.2.5 WDTCSTPR : WDT Count Stop Control Register 598
22.2.6 OFS0 : Option Function Select Register 0 598
22.3 Operation 598
22.3.1 Count Operation in each Start Mode 598
22.3.2 Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers 602
22.3.3 Refresh Operation 602
22.3.4 Status Flags 603
22.3.5 Reset Output 603
22.3.6 Interrupt Sources 603
22.3.7 Reading the Down-Counter Value 604
22.3.8 Association between Option Function Select Register 0 (OFS0) and WDT Registers 604
22.4 Output to the Event Link Controller (ELC) 605
22.5 Usage Notes 605
22.5.1 ICU Event Link Setting Register n (IELSRn) Setting 605
23. Independent Watchdog Timer (IWDT) 606
23.1 Overview 606
23.2 Register Descriptions 607
23.2.1 IWDTRR : IWDT Refresh Register 607
23.2.2 IWDTSR : IWDT Status Register 608
23.2.3 OFS0 : Option Function Select Register 0 609
23.3 Operation 611
23.3.1 Auto Start Mode 611
23.3.2 Refresh Operation 612
23.3.3 Status Flags 613
23.3.4 Reset Output 614
23.3.5 Interrupt Sources 614
23.3.6 Reading the Down-Counter Value 614
23.4 Output to the Event Link Controller (ELC) 615
23.5 Usage Notes 615
23.5.1 Refresh Operations 615
23.5.2 Clock Division Ratio Setting 615
23.5.3 Constraints on the ICU Event Link Setting Register n (IELSRn) Setting 615
24. Serial Communications Interface (SCI) 616
24.1 Overview 616
24.2 Register Descriptions 619
24.2.1 RSR : Receive Shift Register 619
24.2.2 RDR : Receive Data Register 619

24.2.3 RDRHL : Receive Data Register 619
24.2.4 FRDRHL/FRDRH/FRDRL : Receive FIFO Data Register 620
24.2.5 TDR : Transmit Data Register 621
24.2.6 TDRHL : Transmit Data Register 621
24.2.7 FTDRHL/FTDRH/FTDRL : Transmit FIFO Data Register 622
24.2.8 TSR : Transmit Shift Register 622
24.2.9 SMR : Serial Mode Register for Non-Smart Card Interface Mode (SCMR.SMIF = 0) 623
24.2.10 SMR_SMCI : Serial Mode Register for Smart Card Interface Mode (SCMR.SMIF = 1) 624
24.2.11 SCR : Serial Control Register for Non-Smart Card Interface Mode (SCMR.SMIF = 0) 626
24.2.12 SCR_SMCI : Serial Control Register for Smart Card Interface Mode (SCMR.SMIF = 1) 627
24.2.13 SSR : Serial Status Register for Non-Smart Card Interface and Non-FIFO Mode (SCMR.SMIF = 0, FCR.FM = 0) 629
24.2.14 SSR_FIFO : Serial Status Register for Non-Smart Card Interface and FIFO Mode (SCMR.SMIF = 0, FCR.FM = 1) 631
24.2.15 SSR_SMCI : Serial Status Register for Smart Card Interface Mode (SCMR.SMIF = 1) 634
24.2.16 SCMR : Smart Card Mode Register 636
24.2.17 BRR : Bit Rate Register 638
24.2.18 MDDR : Modulation Duty Register 643
24.2.19 SEMR : Serial Extended Mode Register 646
24.2.20 SNFR : Noise Filter Setting Register 647
24.2.21 SIMR1 : IIC Mode Register 1 648
24.2.22 SIMR2 : IIC Mode Register 2 649
24.2.23 SIMR3 : IIC Mode Register 3 649
24.2.24 SISR : IIC Status Register 651
24.2.25 SPMR : SPI Mode Register 652
24.2.26 FCR : FIFO Control Register 653
24.2.27 FDR : FIFO Data Count Register 654
24.2.28 LSR : Line Status Register 655
24.2.29 CDR : Compare Match Data Register 656
24.2.30 DCCR : Data Compare Match Control Register 656
24.2.31 SPTR : Serial Port Register 658
24.3 Operation in Asynchronous Mode 658
24.3.1 Serial Data Transfer Format 659
24.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode 660
24.3.3 Clock 661
24.3.4 Double-Speed Operation and Frequency of 6 Times the Bit Rate 661
24.3.5 CTS and RTS Functions 662
24.3.6 Address Match (Receive Data Match Detection) Function 662
24.3.7 SCI Initialization in Asynchronous Mode 664
24.3.8 Serial Data Transmission in Asynchronous Mode 666
24.3.9 Serial Data Reception in Asynchronous Mode 671
24.4 Multi-Processor Communication Function 684
24.4.1 Multi-Processor Serial Data Transmission 686
24.4.2 Multi-Processor Serial Data Reception 689
24.5 Operation in Clock Synchronous Mode 694
24.5.1 Clock 695
24.5.2 CTS and RTS Functions 695

24.5.3 SCI Initialization in Clock Synchronous Mode ... 696
24.5.4 Serial Data Transmission in Clock Synchronous Mode ... 697
24.5.5 Serial Data Reception in Clock Synchronous Mode ... 701
24.5.6 Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode ... 706
24.6 Operation in Smart Card Interface Mode ... 708
24.6.1 Example Connection ... 709
24.6.2 Data Format (Except in Block Transfer Mode) ... 709
24.6.3 Block Transfer Mode ... 711
24.6.4 Receive Data Sampling Timing and Reception Margin ... 711
24.6.5 SCI Initialization (Smart Card Interface Mode) ... 712
24.6.6 Serial Data Transmission (Except in Block Transfer Mode) ... 713
24.6.7 Serial Data Reception (Except in Block Transfer Mode) ... 715
24.6.8 Clock Output Control ... 717
24.7 Operation in Simple IIC Mode ... 718
24.7.1 Generation of Start, Restart, and Stop Conditions ... 719
24.7.2 Clock Synchronization ... 720
24.7.3 SDAn Output Delay ... 721
24.7.4 SCI Initialization in Simple IIC Mode ... 721
24.7.5 Operation in Master Transmission in Simple IIC Mode ... 722
24.7.6 Master Reception in Simple IIC Mode ... 725
24.8 Operation in Simple SPI Mode ... 727
24.8.1 States of Pins in Master and Slave Modes ... 727
24.8.2 SS Function in Master Mode ... 728
24.8.3 SS Function in Slave Mode ... 728
24.8.4 Relationship between Clock and Transmit/Receive Data ... 728
24.8.5 SCI Initialization in Simple SPI Mode ... 729
24.8.6 Transmission and Reception of Serial Data in Simple SPI Mode ... 729
24.9 Bit Rate Modulation Function ... 729
24.10 Interrupt Sources ... 730
24.10.1 Buffer Operation for SCIn_TXI and SCIn_RXI Interrupts (Non-FIFO Selected) ... 730
24.10.2 Buffer Operation for SCIn_TXI and SCIn_RXI Interrupts (FIFO Selected) ... 730
24.10.3 Interrupts in Asynchronous, Clock Synchronous, and Simple SPI Modes ... 730
24.10.4 Interrupts in Smart Card Interface Mode ... 732
24.10.5 Interrupts in Simple IIC Mode ... 733
24.11 Event Linking ... 733
24.12 Address Non-match Event Output (SCI0_DCUF) ... 734
24.13 Noise Cancellation Function ... 734
24.14 Usage Notes ... 735
24.14.1 Settings for the Module-Stop Function ... 735
24.14.2 SCI Operation during Low Power State ... 735
24.14.3 Break Detection and Processing ... 740
24.14.4 Mark State and Production of Breaks ... 741
24.14.5 Receive Error Flags and Transmit Operation in Clock Synchronous Mode and Simple SPI Mode ... 741
24.14.6 Restrictions on Clock Synchronous Transmission in Clock Synchronous Mode and Simple SPI Mode ... 741
24.14.7 Restrictions on Using DTC ... 742

24.14.8 Notes on Starting Transfer 743
24.14.9 External Clock Input in Clock Synchronous Mode and Simple SPI Mode 743
24.14.10 Limitations on Simple SPI Mode 743
24.14.11 Notes on Transmit Enable Bit (SCR.TE) 744
24.14.12 Note on Stopping Reception When Using the RTS Function in Asynchronous Mode 744
25. I²C Bus Interface (IIC) 745
25.1 Overview 745
25.2 Register Descriptions 747
25.2.1 ICCR1 : I²C Bus Control Register 1 747
25.2.2 ICCR2 : I²C Bus Control Register 2 749
25.2.3 ICMR1 : I²C Bus Mode Register 1 752
25.2.4 ICMR2 : I²C Bus Mode Register 2 753
25.2.5 ICMR3 : I²C Bus Mode Register 3 754
25.2.6 ICFER : I²C Bus Function Enable Register 756
25.2.7 ICSER : I²C Bus Status Enable Register 757
25.2.8 ICIER : I²C Bus Interrupt Enable Register 758
25.2.9 ICSR1 : I²C Bus Status Register 1 760
25.2.10 ICSR2 : I²C Bus Status Register 2 762
25.2.11 ICWUR : I²C Bus Wakeup Unit Register 765
25.2.12 ICWUR2 : I²C Bus Wakeup Unit Register 2 766
25.2.13 SARLy : Slave Address Register L y (y = 0 to 2) 767
25.2.14 SARUy : Slave Address Register U y (y = 0 to 2) 768
25.2.15 ICBRL : I²C Bus Bit Rate Low-Level Register 768
25.2.16 ICBRH : I²C Bus Bit Rate High-Level Register 769
25.2.17 ICDRT : I²C Bus Transmit Data Register 770
25.2.18 ICDRR : I²C Bus Receive Data Register 770
25.2.19 ICDRS : I²C Bus Shift Register 771
25.3 Operation 771
25.3.1 Communication Data Format 771
25.3.2 Initial Settings 772
25.3.3 Master Transmit Operation 773
25.3.4 Master Receive Operation 777
25.3.5 Slave Transmit Operation 782
25.3.6 Slave Receive Operation 784
25.4 SCL Synchronization Circuit 786
25.5 SDA Output Delay Function 787
25.6 Digital Noise Filter Circuits 788
25.7 Address Match Detection 789
25.7.1 Slave-Address Match Detection 789
25.7.2 Detection of General Call Address 792
25.7.3 Device-ID Address Detection 793
25.7.4 Host Address Detection 794
25.8 Wakeup Function 795
25.8.1 Normal Wakeup Mode 1 796
25.8.2 Normal Wakeup Mode 2 799

25.8.3 Command Recovery Mode and EEP Response Mode (Special Wakeup Modes) 801
25.9 Automatic Low-Hold Function for SCL 804
25.9.1 Function to Prevent Wrong Transmission of Transmit Data 804
25.9.2 NACK Reception Transfer Suspension Function 805
25.9.3 Function to Prevent Failure to Receive Data 806
25.10 Arbitration-Lost Detection Functions 807
25.10.1 Master Arbitration-Lost Detection (MALE Bit) 807
25.10.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit) 809
25.10.3 Slave Arbitration-Lost Detection (SALE Bit) 810
25.11 Start, Restart, and Stop Condition Issuing Function 811
25.11.1 Issuing a Start Condition 811
25.11.2 Issuing a Restart Condition 811
25.11.3 Issuing a Stop Condition 813
25.12 Bus Hanging 813
25.12.1 Timeout Function 813
25.12.2 Extra SCL Clock Cycle Output Function 814
25.12.3 IIC Reset and Internal Reset 815
25.13 SMBus Operation 815
25.13.1 SMBus Timeout Measurement 816
25.13.2 Packet Error Code (PEC) 817
25.13.3 SMBus Host Notification Protocol (Notify ARP Master Command) 817
25.14 Interrupt Sources 817
25.14.1 Buffer Operation for IICn_TXI and IICn_RXI Interrupts 818
25.15 State of Registers When Issuing Each Condition 818
25.16 Event Link Output 819
25.16.1 Interrupt Handling and Event Linking 819
25.17 Usage Notes 820
25.17.1 Settings for the Module-Stop Function 820
25.17.2 Notes on Starting Transfer 820
26. Serial Peripheral Interface (SPI) 821
26.1 Overview 821
26.2 Register Descriptions 823
26.2.1 SPCR : SPI Control Register 823
26.2.2 SSLP : SPI Slave Select Polarity Register 824
26.2.3 SPPCR : SPI Pin Control Register 825
26.2.4 SPSR : SPI Status Register 826
26.2.5 SPDR/SPDR_HA/SPDR_BY : SPI Data Register 828
26.2.6 SPBR : SPI Bit Rate Register 830
26.2.7 SPDCR : SPI Data Control Register 831
26.2.8 SPCKD : SPI Clock Delay Register 832
26.2.9 SSLND : SPI Slave Select Negation Delay Register 833
26.2.10 SPND : SPI Next-Access Delay Register 833
26.2.11 SPCR2 : SPI Control Register 2 834
26.2.12 SPCMD0 : SPI Command Register 0 835
26.3 Operation 837
26.3.1 Overview of SPI Operation 837
26.3.2 Controlling the SPI Pins 838

26.3.3 SPI System Configuration Examples 839
26.3.4 Data Formats 843
26.3.5 Transfer Formats 852
26.3.6 Data Transfer Modes 854
26.3.7 Transmit Buffer Empty and Receive Buffer Full Interrupts 855
26.3.8 Error Detection 857
26.3.9 Initializing the SPI 862
26.3.10 SPI Operation 862
26.3.11 Clock Synchronous Operation 873
26.3.12 Loopback Mode 877
26.3.13 Self-Diagnosis of Parity Bit Function 877
26.3.14 Interrupt Sources 878
26.4 Event Link Controller Event Output 879
26.4.1 Receive Buffer Full Event Output 879
26.4.2 Transmit Buffer Empty Event Output 880
26.4.3 Mode-Fault, Underrun, Overrun, or Parity Error Event Output 880
26.4.4 SPI Idle Event Output 880
26.4.5 Transmission-Completed Event Output 880
26.5 Usage Notes 881
26.5.1 Settings for the Module-Stop State 881
26.5.2 Constraint on Low-Power Functions 881
26.5.3 Constraints on Starting Transfer 881
26.5.4 Constraints on Mode-Fault, Underrun, Overrun, or Parity Error Event Output 881
26.5.5 Constraints on the SPSR.SPRF and SPSR.SPTEF Flags 881
27. Cyclic Redundancy Check (CRC) 882
27.1 Overview 882
27.2 Register Descriptions 883
27.2.1 CRCCR0 : CRC Control Register 0 883
27.2.2 CRCCR1 : CRC Control Register 1 884
27.2.3 CRCDIR/CRCDIR_BY : CRC Data Input Register 884
27.2.4 CRCDOR/CRCDOR_HA/CRCDOR_BY : CRC Data Output Register 885
27.2.5 CRCSAR : Snoop Address Register 885
27.3 Operation 885
27.3.1 Basic Operation 885
27.3.2 CRC Snoop Function 888
27.4 Usage Notes 889
27.4.1 Settings for the Module-Stop State 889
27.4.2 Note on Transmission 889
28. High-Speed Analog Comparator (ACMPHS) 890
28.1 Overview 890
28.2 Register Descriptions 891
28.2.1 COMPMDR0 : ACMPHS Mode Setting Register 0 891
28.2.2 COMPFIR0 : ACMPHS Filter Control Register 0 892
28.2.3 COMPOCR0 : ACMPHS Output Control Register 0 893
28.2.4 CMP0SEL : Comparator 0 Input Signal Selection Control Register 894
28.2.5 CMP1SEL : Comparator 1 Input Signal Selection Control Register 894

28.3 Operation 894
28.4 Noise Filter 896
28.5 ACMPHS Interrupts 897
28.6 ELC Event Output 897
28.7 Interrupt Handling and ELC Linking 897
28.8 Comparator Pin Output 897
28.9 Usage Notes 897
28.9.1 Module-Stop Function Settings 897
29. 12-Bit A/D Converter (ADC12) 898
29.1 Overview 898
29.2 Register Descriptions 902
29.2.1 ADDRn : A/D Data Registers n (n = 0 to 2, 5 to 10, 19 to 22) 902
29.2.2 ADDBLDR : A/D Data Duplexing Register 903
29.2.3 ADSHCR : A/D S&H Circuit Control Register 904
29.2.4 ADDBLDRn : A/D Data Duplexing Register n (n = A, B) 905
29.2.5 ADTSDR : A/D Temperature Sensor Data Register 906
29.2.6 ADOCDR : A/D Internal Reference Voltage Data Register 907
29.2.7 ADRD : A/D Self-Diagnosis Data Register 909
29.2.8 ADCSR : A/D Control Register 909
29.2.9 ADANSA0 : A/D Channel Select Register A0 913
29.2.10 ADANSA1 : A/D Channel Select Register A1 914
29.2.11 ADANSB0 : A/D Channel Select Register B0 914
29.2.12 ADANSB1 : A/D Channel Select Register B1 915
29.2.13 ADADS0 : A/D-Converted Value Addition/Average Channel Select Register 0 915
29.2.14 ADADS1 : A/D-Converted Value Addition/Average Channel Select Register 1 917
29.2.15 ADADC : A/D-Converted Value Addition/Average Count Select Register 918
29.2.16 ADCER : A/D Control Extended Register 918
29.2.17 ADSTRGR : A/D Conversion Start Trigger Select Register 920
29.2.18 ADEXICR : A/D Conversion Extended Input Control Register 922
29.2.19 ADSSTRn/ADSSTRL/ADSSTRT/ADSSTRO : A/D Sampling State Register n (n = 0 to 2, 5 to 10) 923
29.2.20 ADDISCR : A/D Disconnection Detection Control Register 924
29.2.21 ADACSR : A/D Conversion Operation Mode Select Register 925
29.2.22 ADGSPCR : A/D Group Scan Priority Control Register 925
29.2.23 ADCMPCR : A/D Compare Function Control Register 926
29.2.24 ADCMPANSR0 : A/D Compare Function Window A Channel Select Register 0 928
29.2.25 ADCMPANSR1 : A/D Compare Function Window A Channel Select Register 1 928
29.2.26 ADCMPANSER : A/D Compare Function Window A Extended Input Select Register 929
29.2.27 ADCMPLR0 : A/D Compare Function Window A Comparison Condition Setting Register 0 929
29.2.28 ADCMPLR1 : A/D Compare Function Window A Comparison Condition Setting Register 1 931
29.2.29 ADCMPLER : A/D Compare Function Window A Extended Input Comparison Condition Setting Register 931
29.2.30 ADCMPDRn : A/D Compare Function Window A Lower-Side/Upper-Side Level Setting Register n (n = 0, 1) 932
29.2.31 ADWINnLB : A/D Compare Function Window B Lower-Side/Upper-Side Level Setting Register n (n = L, U) 933

29.2.32 ADCMPSR0 : A/D Compare Function Window A Channel Status Register 0 935
29.2.33 ADCMPSR1 : A/D Compare Function Window A Channel Status Register 1 935
29.2.34 ADCMPSER : A/D Compare Function Window A Extended Input Channel Status Register 936
29.2.35 ADCMPBNSR : A/D Compare Function Window B Channel Select Register 937
29.2.36 ADCMPBSR : A/D Compare Function Window B Status Register 939
29.2.37 ADWINMON : A/D Compare Function Window A/B Status Monitor Register 940
29.2.38 ADHVREFCNT : A/D High-Potential/Low-Potential Reference Voltage Control Register 941
29.3 Operation 942
29.3.1 Scanning Operation 942
29.3.2 Single Scan Mode 943
29.3.3 Continuous Scan Mode 949
29.3.4 Group Scan Mode 953
29.3.5 Compare Function for Windows A and B 962
29.3.6 Analog Input Sampling and Scan Conversion Time 966
29.3.7 Usage Example of A/D Data Register Automatic Clearing Function 968
29.3.8 A/D-Converted Value Addition/Average Mode 969
29.3.9 Disconnection Detection Assist Function 969
29.3.10 Starting A/D Conversion with an Asynchronous Trigger 971
29.3.11 Starting A/D Conversion with a Synchronous Trigger from a Peripheral Module 972
29.4 Interrupt Sources and DTC Transfer Requests 972
29.4.1 Interrupt Requests 972
29.5 Event Link Function 973
29.5.1 Event Output to the ELC 973
29.5.2 ADC12 Operation through an Event from the ELC 973
29.6 Selecting Reference Voltage 974
29.7 A/D Conversion Procedure When Selecting Internal Reference Voltage as High-Potential Reference Voltage 974
29.8 Usage Notes 974
29.8.1 Constraints on Setting the Registers 974
29.8.2 Constraints on Reading the Data Registers 974
29.8.3 Constraints on Stopping A/D Conversion 975
29.8.4 A/D Conversion Restart and Termination Timing 977
29.8.5 Constraints on Scan End Interrupt Handling 977
29.8.6 Settings for the Module-Stop Function 977
29.8.7 Notes on Entering the Low-Power States 977
29.8.8 Error in Absolute Accuracy When Disconnection Detection Assistance Is in Use 977
29.8.9 ADHSC Bit Rewriting Procedure 977
29.8.10 Constraints on Operating Modes and Status Bits 978
29.8.11 Notes on Board Design 978
29.8.12 Constraints on Noise Prevention 978
29.8.13 Port Settings When Using the ADC12 Input 979
29.8.14 Notes on Canceling Software Standby Mode 979
29.8.15 S/H circuit hold time limitations 979
30. Temperature Sensor (TSN) 980
30.1 Overview 980
30.2 Register Descriptions 980

30.2.1 TSCDR : Temperature Sensor Calibration Data Register 980
30.3 Using the Temperature Sensor 981
30.3.1 Preparation for Using the Temperature Sensor 981
30.3.2 Procedures for Using the Temperature Sensor 982
31. Data Operation Circuit (DOC) 983
31.1 Overview 983
31.2 DOC Register Descriptions 983
31.2.1 DOCR : DOC Control Register 983
31.2.2 DODIR : DOC Data Input Register 984
31.2.3 DODSR : DOC Data Setting Register 985
31.3 Operation 985
31.3.1 Data Comparison Mode 985
31.3.2 Data Addition Mode 985
31.3.3 Data Subtraction Mode 986
31.4 Interrupt Source 986
31.5 Output of an Event Signal to the Event Link Controller (ELC) 987
31.6 Usage Notes 987
31.6.1 Settings for the Module-Stop State 987
32. SRAM 988
32.1 Overview 988
32.2 Register Descriptions 988
32.2.1 PARIOAD : SRAM Parity Error Operation After Detection Register 988
32.2.2 SRAMPRCR : SRAM Protection Register 988
32.2.3 Trace Control (for the MTB) 989
32.2.4 CoreSight (for MTB) 989
32.3 Operation 990
32.3.1 Parity Calculation Function 990
32.3.2 SRAM Error Sources 991
32.3.3 Access Cycle 991
32.3.4 Low-Power Function 992
32.4 Usage Notes 992
32.4.1 Instruction Fetch from the SRAM Area 992
32.4.2 SRAM Store Buffer 992
33. Flash Memory 993
33.1 Overview 993
33.2 Memory Structure 994
33.3 Register Descriptions 995
33.3.1 DFLCTL : Data Flash Control Register 995
33.3.2 PFBER : Prefetch Buffer Enable Register 995
33.3.3 FENTRYR : Flash P/E Mode Entry Register 996
33.3.4 FPR : Protection Unlock Register 997
33.3.5 FPSR : Protection Unlock Status Register 997
33.3.6 FPMCR : Flash P/E Mode Control Register 998
33.3.7 FISR : Flash Initial Setting Register 999
33.3.8 FRESETR : Flash Reset Register 1000
33.3.9 FASR : Flash Area Select Register 1000

33.3.10 FCR : Flash Control Register 1001
33.3.11 FEXCR : Flash Extra Area Control Register 1002
33.3.12 FSARH : Flash Processing Start Address Register H 1004
33.3.13 FSARL : Flash Processing Start Address Register L 1005
33.3.14 FEARH : Flash Processing End Address Register H 1005
33.3.15 FEARL : Flash Processing End Address Register L 1006
33.3.16 FWBL0 : Flash Write Buffer Register L0 1006
33.3.17 FWBH0 : Flash Write Buffer Register H0 1006
33.3.18 FRBL0 : Flash Read Buffer Register L0 1007
33.3.19 FRBH0 : Flash Read Buffer Register H0 1007
33.3.20 FSTATR1 : Flash Status Register 1 1007
33.3.21 FSTATR2 : Flash Status Register 2 1008
33.3.22 FEAMH : Flash Error Address Monitor Register H 1009
33.3.23 FEAML : Flash Error Address Monitor Register L 1009
33.3.24 FSCMR : Flash Start-Up Setting Monitor Register 1010
33.3.25 FAWSMR : Flash Access Window Start Address Monitor Register 1010
33.3.26 FAWEMR : Flash Access Window End Address Monitor Register 1011
33.3.27 UIDRn : Unique ID Registers n (n = 0 to 3) 1011
33.3.28 PNRn : Part Numbering Register n (n = 0 to 3) 1011
33.3.29 MCUVER : MCU Version Register 1012
33.4 Instruction Prefetch from Flash Memory 1012
33.5 Operating Modes Associated with the Flash Memory 1012
33.5.1 ID Code Protection 1013
33.6 Overview of Functions 1014
33.6.1 Configuration Area Bit Map 1016
33.6.2 Startup Area Select 1016
33.6.3 Protection by Access Window 1017
33.7 Programming Commands 1018
33.8 Suspend Operation 1018
33.9 Protection 1018
33.9.1 Startup Program Protection 1018
33.9.2 Area Protection 1019
33.10 Serial Programming Mode 1020
33.10.1 SCI Boot Mode 1020
33.11 Using a Serial Programmer 1021
33.11.1 Serial Programming 1021
33.12 Self-Programming 1021
33.12.1 Overview 1021
33.12.2 Background Operation 1022
33.13 Programming and Erasure 1022
33.13.1 Sequencer Modes 1022
33.13.2 Software Commands 1023
33.13.3 Software Command Usage 1024
33.14 Reading the Flash Memory 1041
33.14.1 Reading the Code Flash Memory 1041
33.14.2 Reading the Data Flash Memory 1041
33.15 Usage Notes 1041

33.15.1 Erase Suspended Area 1041
33.15.2 Suspension by Erase Suspend Commands 1042
33.15.3 Constraints on Additional Writes 1042
33.15.4 Reset during Programming and Erasure 1042
33.15.5 Non-Maskable Interrupt Disabled during Programming and Erasure 1042
33.15.6 Location of Interrupt Vectors during Programming and Erasure 1042
33.15.7 Programming and Erasure in Subosc-Speed Operating Mode 1042
33.15.8 Abnormal Termination during Programming and Erasure 1042
33.15.9 Actions Prohibited during Programming and Erasure 1042
33.15.10 Additional Programming Disabled 1042
33.15.11 Flash-IF clock (ICLK) during Program/Erase 1043
34. Internal Voltage Regulator 1044
34.1 Overview 1044
34.2 Operation 1044
35. Electrical Characteristics 1045
35.1 Absolute Maximum Ratings 1045
35.2 DC Characteristics 1046
35.2.1 Tj/Ta Definition 1046
35.2.2 I/O V_{IH}, V_{IL} 1047
35.2.3 I/O I_{OH}, I_{OL} 1047
35.2.4 I/O V_{OH}, V_{OL}, and Other Characteristics 1050
35.2.5 Operating and Standby Current 1052
35.2.6 VCC Rise, Fall Gradient, and Ripple Frequency 1054
35.3 AC Characteristics 1055
35.3.1 Frequency 1055
35.3.2 Clock Timing 1056
35.3.3 Reset Timing 1058
35.3.4 Wakeup Time 1059
35.3.5 NMI and IRQ Noise Filter 1062
35.3.6 I/O Ports, POEG, GPT, AGTW, KINT, and ADC12 Trigger Timing 1063
35.3.7 CAC Timing 1070
35.3.8 SCI Timing 1072
35.3.9 SPI Timing 1079
35.3.10 IIC Timing 1085
35.3.11 CLKOUT Timing 1086
35.4 ADC12 Characteristics 1087
35.5 TSN Characteristics 1096
35.6 OSC Stop Detect Characteristics 1096
35.7 POR and LVD Characteristics 1097
35.8 Comparator Characteristics 1102
35.9 Flash Memory Characteristics 1102
35.9.1 Code Flash Memory Characteristics 1103
35.9.2 Data Flash Memory Characteristics 1104
35.10 Serial Wire Debug (SWD) 1106
Appendix 1. Port States in each Processing Mode 1108
Appendix 2. Package Dimensions 1110

Appendix 3. I/O Registers 1115
3.1 Peripheral Base Addresses 1115
3.2 Access Cycles 1116

Ultra low power 64 MHz Arm® Cortex®-M23 core, up to 64-KB code flash memory, 8-KB SRAM, 12-bit A/D Converter, High Speed Analog Comparator, Safety features.

Features

■ Arm Cortex-M23 Core

- Armv8-M architecture
- Maximum operating frequency: 64 MHz
- Arm Memory Protection Unit (Arm MPU) with 8 regions
- Debug and Trace: DWT, FPB, CoreSight™ MTB-M23
- CoreSight Debug Port: SW-DP

■ Memory

- Up to 64-KB code flash memory
- 2-KB data flash memory (100,000 program/erase (P/E) cycles)
- 8-KB SRAM
- Memory protection units (MPU)
- 128-bit unique ID

■ Connectivity

- Serial Communications Interface (SCI) × 4
  - Asynchronous interfaces
  - 8-bit clock synchronous interface
  - Simple IIC
  - Simple SPI
  - Smart card interface
- Serial Peripheral Interface (SPI) × 1
- I^2C bus interface (IIC) × 1

■ Analog

- 12-Bit A/D Converter (ADC12) with 3 sample-and-hold circuits
- Temperature Sensor (TSN)
- High Speed Analog Comparator (ACMPHS) × 2

■ Timers

- General PWM Timer 16-bit (GPT16) × 4
- 32-bit Low Power Asynchronous General Purpose Timer (AGTW) × 2
- Watchdog Timer (WDT)

■ Safety

- SRAM parity error check
- Flash area protection
- ADC self-diagnosis function
- Clock Frequency Accuracy Measurement Circuit (CAC)
- Cyclic Redundancy Check (CRC) calculator
- Data Operation Circuit (DOC)
- Port Output Enable for GPT (POEG)
- Independent Watchdog Timer (IWDT)
- GPIO readback level detection
- Register write protection
- Main oscillator stop detection
- Illegal memory access detection

■ System and Power Management

- Low power modes
- Event Link Controller (ELC)
- Data Transfer Controller (DTC)
- Key Interrupt Function (KINT)
- Power-on reset
- Low Voltage Detection (LVD) with voltage settings

■ Multiple Clock Sources

- Main clock oscillator (MOSC) (1 to 20 MHz)
- High-speed on-chip oscillator (HOCO) (24/32/48/64 MHz)
- Middle-speed on-chip oscillator (MOCO) (8 MHz)
- Low-speed on-chip oscillator (LOCO) (32.768 kHz)
- Clock trim function for HOCO/MOCO/LOCO
- IWDT-dedicated on-chip oscillator (15 kHz)
- Clock out support

■ Up to 40 pins for general I/O ports

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

■ Operating Voltage

- VCC: 1.6 to 5.5 V

■ Operating Temperature and Packages

- Ta = -40°C to +105°C
  - 48-pin LQFP (7 mm × 7 mm, 0.5 mm pitch)
  - 48-pin HWQFN (7 mm × 7 mm, 0.5 mm pitch)
  - 32-pin LQFP (7 mm × 7 mm, 0.8 mm pitch)
  - 32-pin HWQFN (5 mm × 5 mm, 0.5 mm pitch)
  - 24-pin HWQFN (4 mm × 4 mm, 0.5 mm pitch)
- Ta = -40°C to +125°C
  - 48-pin LQFP (7 mm × 7 mm, 0.5 mm pitch)
  - 48-pin HWQFN (7 mm × 7 mm, 0.5 mm pitch)
  - 32-pin LQFP (7 mm × 7 mm, 0.8 mm pitch)
  - 32-pin HWQFN (5 mm × 5 mm, 0.5 mm pitch)
  - 24-pin HWQFN (4 mm × 4 mm, 0.5 mm pitch)

1. Overview

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

The MCU in this series incorporates an energy-efficient Arm Cortex®-M23 32-bit core, that is particularly well suited for cost-sensitive and low-power applications, with the following features:

- Up to 64-KB code flash memory
- 8-KB SRAM
- 12-bit A/D Converter (ADC12)

1.1 Function Outline

Table 1.1 Arm core

Feature	Functional description
Arm Cortex-M23 core	• Maximum operating frequency: up to 64 MHz • Arm Cortex-M23 core: – Revision: r1p0-00rel0 – Armv8-M architecture profile – Single-cycle integer multiplier – 19-cycle integer divider • Arm Memory Protection Unit (Arm MPU): – Armv8 Protected Memory System Architecture – 8 protect regions • SysTick timer: – Driven by SYSTICCLK (LOCO) or ICLK

Table 1.2 Memory

Feature	Functional description
Code flash memory	Maximum 64-KB of code flash memory. See section 33, Flash Memory.
Data flash memory	2-KB of data flash memory. See section 33, Flash Memory.
Option-setting memory	The option-setting memory determines the state of the MCU after a reset. See section 6, Option-Setting Memory.
SRAM	On-chip high-speed SRAM with parity bit. See section 32, SRAM.

Table 1.3 System (1 of 2)

Feature	Functional description
Operating modes	Two operating modes: • Single-chip mode • SCI boot mode See section 3, Operating Modes.
Resets	The MCU provides 12 resets (RES pin reset, power-on reset, independent watchdog timer reset, watchdog timer reset, voltage monitor 0/1/2 resets, SRAM parity error reset, bus master/slave MPU error resets, CPU stack pointer error reset, software reset). See section 5, Resets.
Low Voltage Detection (LVD)	The Low Voltage Detection (LVD) module monitors the voltage level input to the VCC pin. The detection level can be selected by register settings. The LVD module consists of three separate voltage level detectors (LVD0, LVD1, LVD2). LVD0, LVD1, and LVD2 measure the voltage level input to the VCC pin. LVD registers allow your application to configure detection of VCC changes at various voltage thresholds. See section 7, Low Voltage Detection (LVD).

Table 1.3 System (2 of 2)

Feature	Functional description
Clocks	• Main clock oscillator (MOSC) • High-speed on-chip oscillator (HOCO) • Middle-speed on-chip oscillator (MOCO) • Low-speed on-chip oscillator (LOCO) • IWDT-dedicated on-chip oscillator • Clock out support See section 8, Clock Generation Circuit.
Clock Frequency Accuracy Measurement Circuit (CAC)	The Clock Frequency Accuracy Measurement Circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock selected as the measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range. When measurement is complete or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated. See section 9, Clock Frequency Accuracy Measurement Circuit (CAC).
Interrupt Controller Unit (ICU)	The Interrupt Controller Unit (ICU) controls which event signals are linked to the Nested Vector Interrupt Controller (NVIC), and the Data Transfer Controller (DTC) modules. The ICU also controls non-maskable interrupts. See section 12, Interrupt Controller Unit (ICU).
Key Interrupt Function (KINT)	The key interrupt function (KINT) generates the key interrupt by detecting rising or falling edge on the key interrupt input pins. See section 18, Key Interrupt Function (KINT).
Low power modes	Power consumption can be reduced in multiple ways, including setting clock dividers, stopping modules, selecting power control mode in normal operation, and transitioning to low power modes. See section 10, Low Power Modes.
Register write protection	The register write protection function protects important registers from being overwritten due to software errors. The registers to be protected are set with the Protect Register (PRCR). See section 11, Register Write Protection.
Memory Protection Unit (MPU)	The MCU has four Memory Protection Units (MPUs) and a CPU stack pointer monitor function are provided. See section 14, Memory Protection Unit (MPU).
Watchdog Timer (WDT)	The Watchdog Timer (WDT) is a 14-bit down counter that can be used to reset the MCU when the counter underflows because the system has run out of control and is unable to refresh the WDT. In addition, the WDT can be used to generate a non-maskable interrupt or an underflow interrupt or watchdog timer reset. See section 22, Watchdog Timer (WDT).
Independent Watchdog Timer (IWDT)	The Independent Watchdog Timer (IWDT) consists of a 14-bit down counter that must be serviced periodically to prevent counter underflow. The IWDT provides functionality to reset the MCU or to generate a non-maskable interrupt or an underflow interrupt. Because the timer operates with an independent, dedicated clock source, it is particularly useful in returning the MCU to a known state as a fail-safe mechanism when the system runs out of control. The IWDT can be triggered automatically by a reset, underflow, refresh error, or a refresh of the count value in the registers. See section 23, Independent Watchdog Timer (IWDT).

Table 1.4 Event link

Feature	Functional description
Event Link Controller (ELC)	The Event Link Controller (ELC) uses the event requests generated by various peripheral modules as source signals to connect them to different modules, allowing direct link between the modules without CPU intervention. See section 16, Event Link Controller (ELC).

Table 1.5 Direct memory access

Feature	Functional description
Data Transfer Controller (DTC)	A Data Transfer Controller (DTC) module is provided for transferring data when activated by an interrupt request. See section 15, Data Transfer Controller (DTC).

Table 1.6 Timers

Feature	Functional description
General PWM Timer (GPT)	The General PWM Timer (GPT) is a 16-bit timer with GPT16 × 4 channels. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. In addition, PWM waveforms can be generated for controlling brushless DC motors. The GPT can also be used as a general-purpose timer. See section 20, General PWM Timer (GPT).
Port Output Enable for GPT (POEG)	The POEG issues requests to stop output from output pins of the general PWM timer (GPT). See section 19, Port Output Enable for GPT (POEG).
Low power Asynchronous General Purpose Timer (AGTW)	The Low Power Asynchronous General Purpose Timer (AGTW) is a 32-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events. This timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and can be accessed with the AGTW register. See section 21, Low Power Asynchronous General Purpose Timer (AGTW).

Table 1.7 Communication interfaces

Feature	Functional description
Serial Communications Interface (SCI)	The Serial Communications Interface (SCI) × 4 channels have asynchronous and synchronous serial interfaces: ● Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA)) ● 8-bit clock synchronous interface ● Simple IIC (master-only) ● Simple SPI ● Smart card interface The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. SCIn (n = 0) has FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using an on-chip baud rate generator. See section 24, Serial Communications Interface (SCI).
I^2C bus interface (IIC)	The I^2C bus interface (IIC) has 1 channel. The IIC module conforms with and provides a subset of the NXP I^2C (Inter-Integrated Circuit) bus interface functions. See section 25, I^2C Bus Interface (IIC).
Serial Peripheral Interface (SPI)	The Serial Peripheral Interface (SPI) has 1 channel. The SPI provides high-speed full-duplex synchronous serial communications with multiple processors and peripheral devices. See section 26, Serial Peripheral Interface (SPI).

Table 1.8 Analog

Feature	Functional description
12-bit A/D Converter (ADC12)	A 12-bit successive approximation A/D converter is provided. Up to 13 analog input channels are selectable. Temperature sensor output and internal reference voltage are selectable for conversion. See section 29, 12-Bit A/D Converter (ADC12).
Temperature Sensor (TSN)	The on-chip Temperature Sensor (TSN) determines and monitors the die temperature for reliable operation of the device. The sensor outputs a voltage directly proportional to the die temperature, and the relationship between the die temperature and the output voltage is fairly linear. The output voltage is provided to the ADC12 for conversion and can be further used by the end application. See section 30, Temperature Sensor (TSN).
High-Speed Analog Comparator (ACMPHS)	The High-Speed Analog Comparator (ACMPHS) compares a reference input voltage with an analog input voltage. Comparator channels ACMPHS0 and ACMPHS1 are independent of each other. The comparison result of the reference input voltage and analog input voltage can be read by software. The comparison result can also be output externally. The reference input voltage can be selected from an input to the CMPREFi (i = 0, 1) pin, input to the CMPIN0j (j = 0, 1, 2) pin or the internal reference voltage (V_{ref}) generated internally in the MCU. See section 28, High-Speed Analog Comparator (ACMPHS).

Table 1.9 Data processing

Feature	Functional description
Cyclic Redundancy Check (CRC) calculator	The Cyclic Redundancy Check (CRC) generates CRC codes to detect errors in the data. The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication. Additionally, various CRC-generation polynomials are available. The snoop function allows to monitor the access to specific addresses. This function is useful in applications that require CRC code to be generated automatically in certain events, such as monitoring writes to the serial transmit buffer and reads from the serial receive buffer. See section 27, Cyclic Redundancy Check (CRC).
Data Operation Circuit (DOC)	The Data Operation Circuit (DOC) compares, adds, and subtracts 16-bit data. When a selected condition applies, 16-bit data is compared and an interrupt can be generated. See section 31, Data Operation Circuit (DOC).

Table 1.10 I/O ports

Feature	Functional description
I/O ports	• I/O ports for the 48-pin LQFP/HWQFN – I/O pins: 39 – Input pins: 1 – Pull-up resistors: 39 – N-ch open-drain outputs: 28 – 5-V tolerance: 2 • I/O ports for the 32-pin LQFP/HWQFN – I/O pins: 25 – Input pins: 1 – Pull-up resistors: 25 – N-ch open-drain outputs: 17 • I/O ports for the 24-pin HWQFN – I/O pins: 17 – Input pins: 1 – Pull-up resistors: 17 – N-ch open-drain outputs: 10

1.2 Block Diagram

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

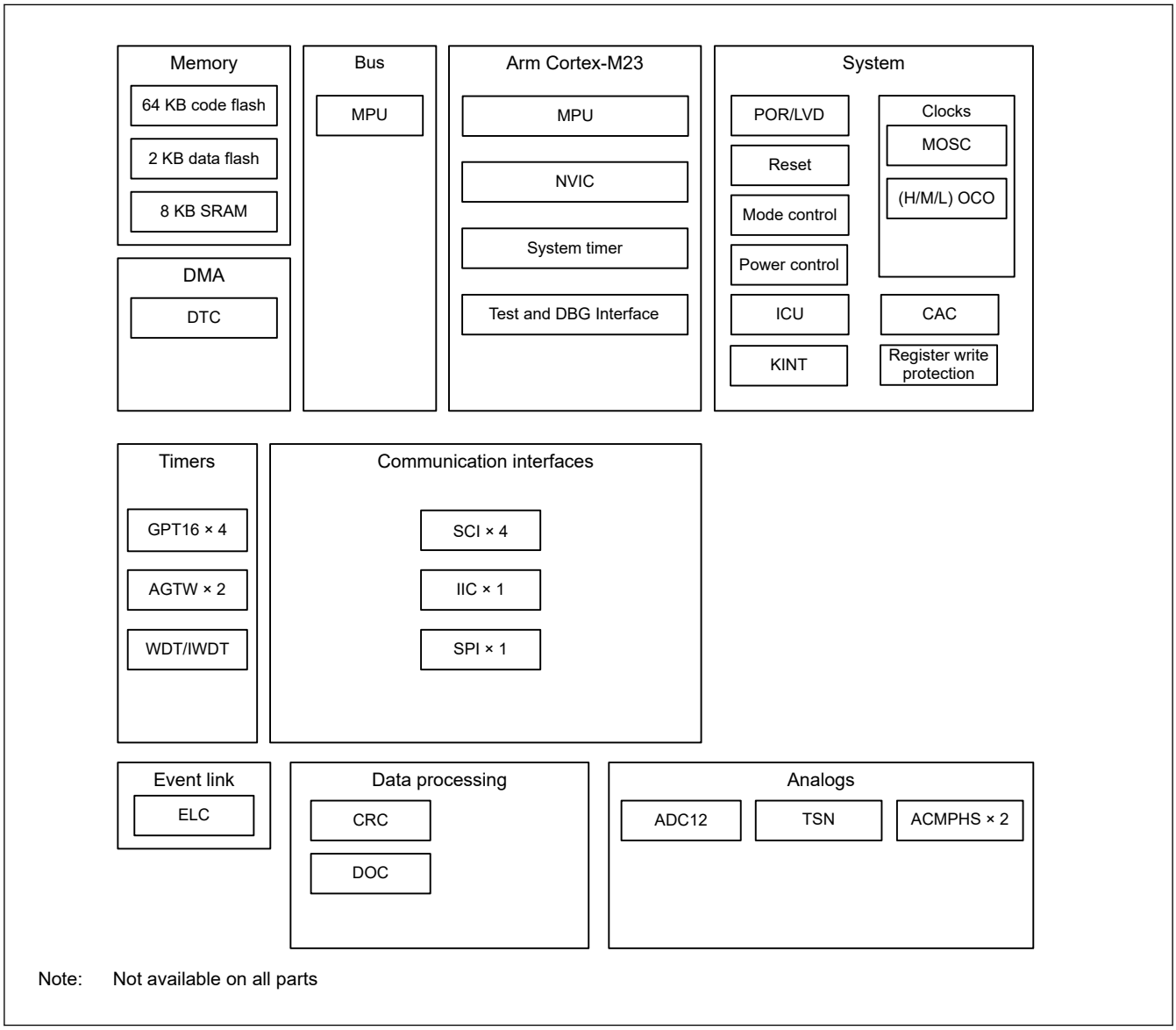


Figure 1.1 Block diagram

1.3 Part Numbering

Figure 1.2 shows the product part number information, including memory capacity and package type. Table 1.11 shows a list of products.

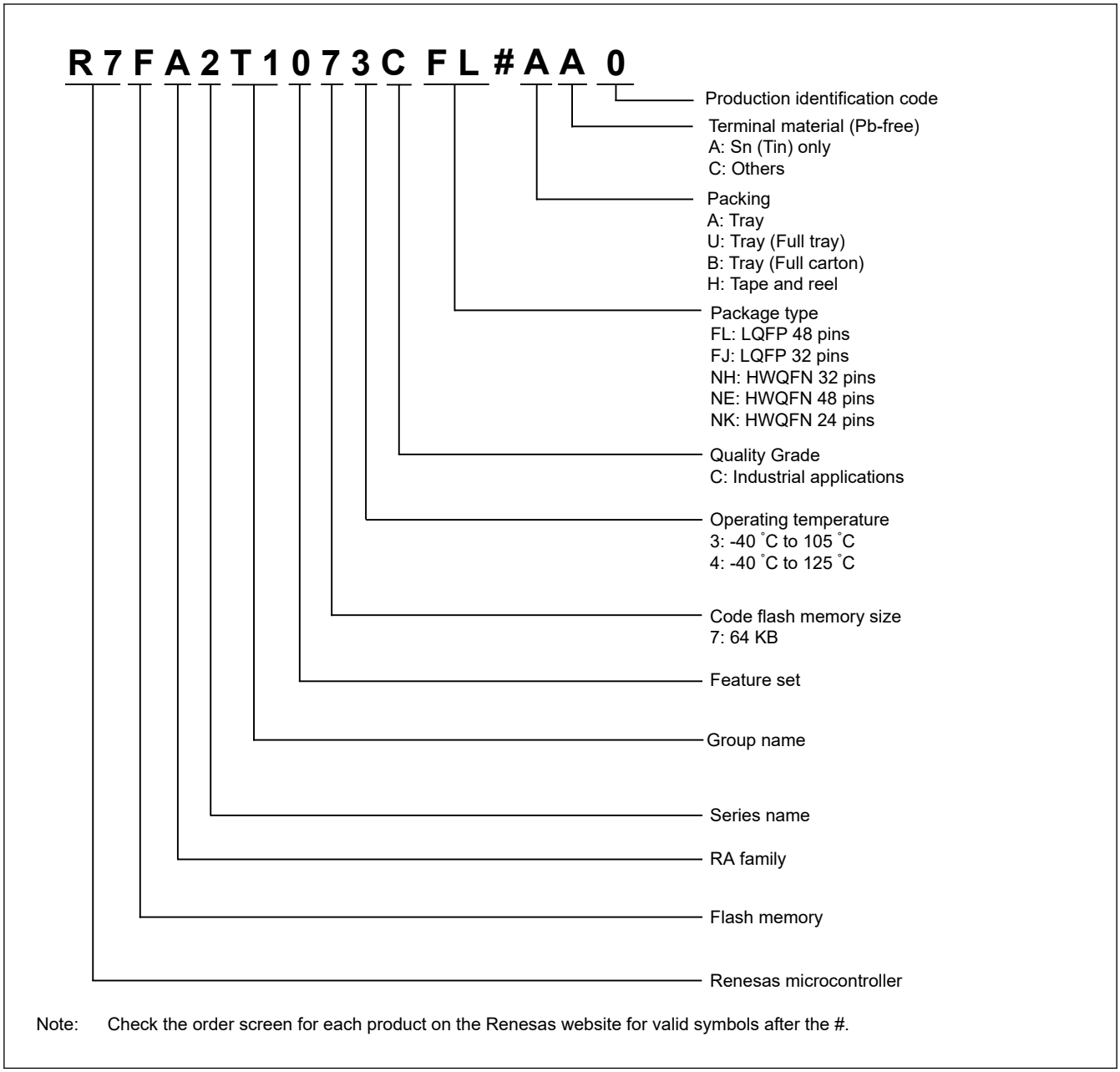


Figure 1.2 Part numbering scheme

Table 1.11 Product list

Product part number	Package code	Code flash	Data flash	SRAM	Operating temperature
R7FA2T1073CFL	PLQP0048KB-B	64	2	8	-40 to +105 °C
R7FA2T1073CFJ	PLQP0032GB-A				
R7FA2T1073CNH	PWQN0032KE-A				
R7FA2T1073CNE	PWQN0048KC-A				
R7FA2T1073CNK	PWQN0024KG-A				
R7FA2T1074CFL	PLQP0048KB-B				-40 to +125 °C
R7FA2T1074CFJ	PLQP0032GB-A				
R7FA2T1074CNH	PWQN0032KE-A				
R7FA2T1074CNE	PWQN0048KC-A				
R7FA2T1074CNK	PWQN0024KG-A				

1.4 Function Comparison

Table 1.12 Function comparison

Parts number		R7FA2T107xxFL	R7FA2T107xxNE	R7FA2T107xxFJ	R7FA2T107xxNH	R7FA2T107xCNK
Pin count		48		32		24
Package		LQFP	WQFN	LQFP	WQFN	WQFN
Code flash memory		64 KB				
Data flash memory		2 KB				
SRAM		8 KB				
System	CPU clock	64 MHz				
	ICU	Yes				
	KINT	5		4		
Event control	ELC	Yes				
DMA	DTC	Yes				
Timers	GPT (POEG)	2				
	GPT16	4				
	AGTW	2				
	WDT/IWDT	Yes				
Communication	SCI	4				
	IIC	1				
	SPI	1				
Analog	ADC12	13		10		9
	ACMPHS	2				
	TSN	Yes				
Data processing	CRC	Yes				
	DOC	Yes				
I/O ports	I/O pins	39		25		17
	Input pins	1				
	Pull-up resistors	39		25		17
	N-ch open-drain outputs	28		17		10
	5-V tolerance	2		0		0

1.5 Pin Functions

Table 1.13 Pin functions (1 of 3)

Function	Signal	I/O	Description
Power supply	VCC	Input	Power supply pin. Connect it to the system power supply. Connect this pin to VSS by a 0.1-μF capacitor. Place the capacitor close to the pin.
	VCL	I/O	Connect this pin to the VSS pin by the smoothing capacitor used to stabilize the internal power supply. Place the capacitor close to the pin.
	VSS	Input	Ground pin. Connect it to the system power supply (0 V).
Clock	XTAL	Output	Pins for a crystal resonator. An external clock signal can be input through the EXTAL pin.
	EXTAL	Input	
	CLKOUT	Output	Clock output pin
Operating mode control	MD	Input	Pin for setting the operating mode. The signal level on this pin must not be changed during operation mode transition on release from the reset state.
System control	RES	Input	Reset signal input pin. The MCU enters the reset state when this signal goes low.
CAC	CACREF	Input	Measurement reference clock input pin
On-chip debug	SWDIO	I/O	Serial wire debug data input/output pin
	SWCLK	Input	Serial wire clock pin
Interrupt	NMI	Input	Non-maskable interrupt request pin
	IRQ0 to IRQ7	Input	Maskable interrupt request pins
GPT16	GTETRGA, GTETRGB	Input	External trigger input pins
	GTADSM0	Output	A/D conversion start request monitor 0 output pin
	GTADSM1	Output	A/D conversion start request monitor 1 output pin
	GTIOCnA, GTIOCnB	I/O	Input capture, output compare, or PWM output pins
	GTCPPOk	Output	Toggle output synchronized with PWM period
	GTIU	Input	Hall sensor input pin U
	GTIV	Input	Hall sensor input pin V
	GTIW	Input	Hall sensor input pin W
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W phase)
AGTW	AGTEE0, AGTEE1	Input	External event input enable signals
	AGTIO0, AGTIO1	I/O	External event input and pulse output pins
	AGTO0	Output	Pulse output pins
	AGTOA0, AGTOA1	Output	Output compare match A output pins
	AGTOB0	Output	Output compare match B output pins

Table 1.13 Pin functions (2 of 3)

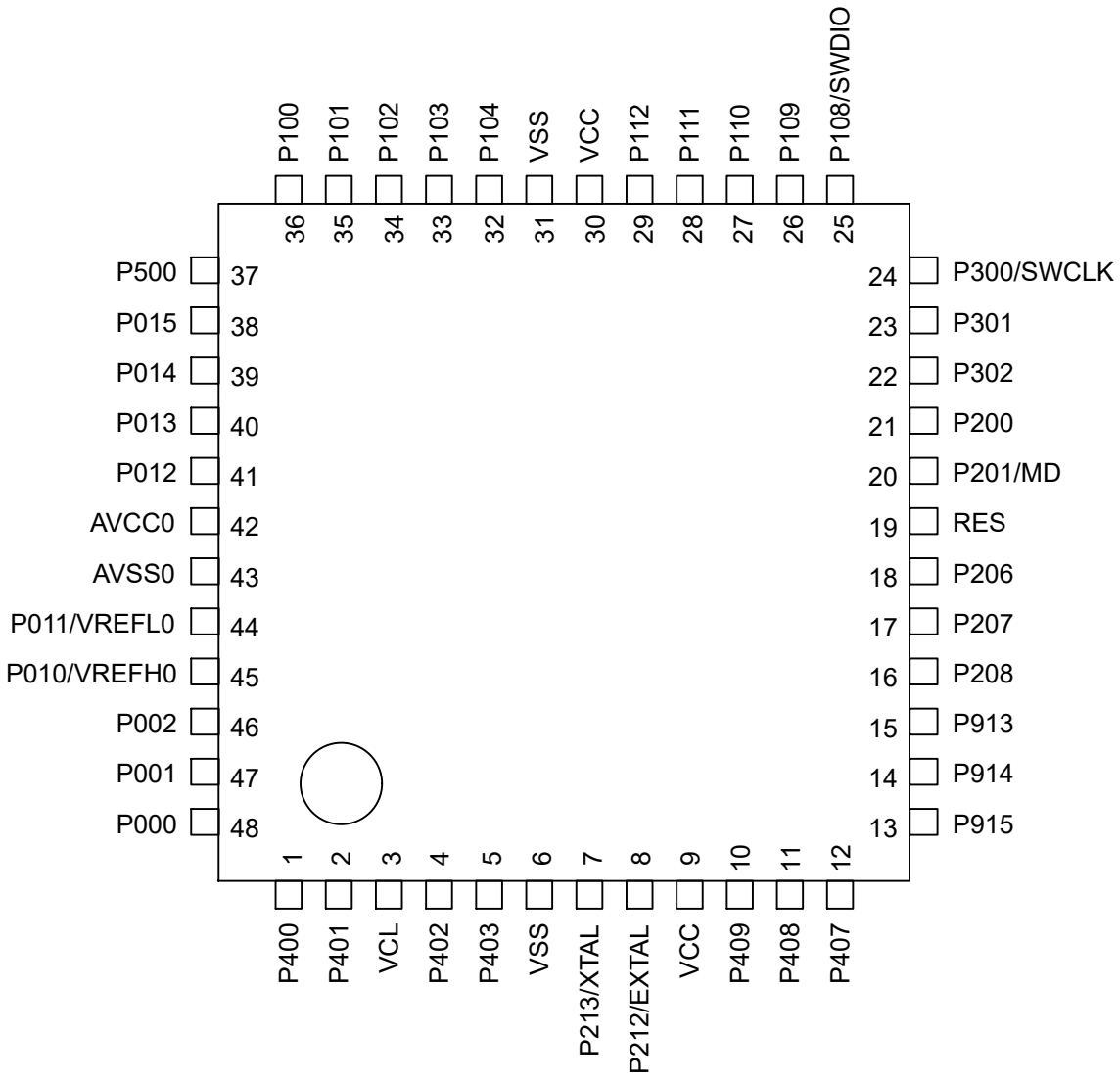
Function	Signal	I/O	Description
SCI	SCKn (n = 0 to 2, 9)	I/O	Input/output pins for the clock (clock synchronous mode)
	RXDn (n = 0 to 2, 9)	Input	Input pins for received data (asynchronous mode/clock synchronous mode)
	TXDn (n = 0 to 2, 9)	Output	Output pins for transmitted data (asynchronous mode/clock synchronous mode)
	CTSn_RTSn (n = 0 to 2, 9)	I/O	Input/output pins for controlling the start of transmission and reception (asynchronous mode/clock synchronous mode), active-low.
	SCLn (n = 0 to 2, 9)	I/O	Input/output pins for the IIC clock (simple IIC mode)
	SDAn (n = 0 to 2, 9)	I/O	Input/output pins for the IIC data (simple IIC mode)
	SCKn (n = 0 to 2, 9)	I/O	Input/output pins for the clock (simple SPI mode)
	MISOn (n = 0 to 2, 9)	I/O	Input/output pins for slave transmission of data (simple SPI mode)
	MOSIn (n = 0 to 2, 9)	I/O	Input/output pins for master transmission of data (simple SPI mode)
	SSn (n = 0 to 2, 9)	Input	Chip-select input pins (simple SPI mode), active-low
IIC	SCLn (n = 0)	I/O	Input/output pins for the clock
	SDAn (n = 0)	I/O	Input/output pins for data
SPI	RSPCKA	I/O	Clock input/output pin
	SSLA0	I/O	Input or output pin for slave selection
	SSLA1	Output	Output pins for slave selection
	MOSIA	I/O	Input or output pins for data output from the master
	MISOA	I/O	Input or output pins for data output from the slave
Analog power supply	AVCC0	Input	Analog power supply pin for the ADC12
	AVSS0	Input	Analog ground pin for the ADC12
	VREFH0	Input	Analog reference voltage supply pin for the ADC12. Connect this pin to AVCC0 when not using the ADC12.
	VREFL0	Input	Analog reference ground pin for the ADC12. Connect this pin to AVSS0 when not using the ADC12.
ADC12	AN000 to AN002 (3ch S/H), AN005 to AN010, AN019 to AN022	Input	Input pins for the analog signals to be processed by the A/D converter.
	ADTRG0_A, ADTRG0_B	Input	Input pin for the external trigger signals that start the A/D conversion, active-low.
KINT	KR00 to KR04	Input	Key interrupt input pins
I/O ports	P000 to P002, P010 to P015	I/O	General-purpose input/output pins
	P100 to P104, P108 to P112	I/O	General-purpose input/output pins
	P200	Input	General-purpose input pin
	P201, P206 to P208, P212, P213	I/O	General-purpose input/output pins
	P300 to P302	I/O	General-purpose input/output pins
	P400 to P403, P407 to P409	I/O	General-purpose input/output pins
	P500	I/O	General-purpose input/output pins
	P913 to P915	I/O	General-purpose input/output pins

Table 1.13 Pin functions (3 of 3)

Function	Signal	I/O	Description
ACMPHS	CMPIN0n (n = 1, 2, 3), CMPIN1	Input	Reference and Analog voltage input pin
	CMPREFn (n = 0, 1)	Input	Reference and Analog voltage input pin
	VCOUT	Output	ACMPHS0 and ACMPHS1 compare outputs are bundled on the VCOUT pin

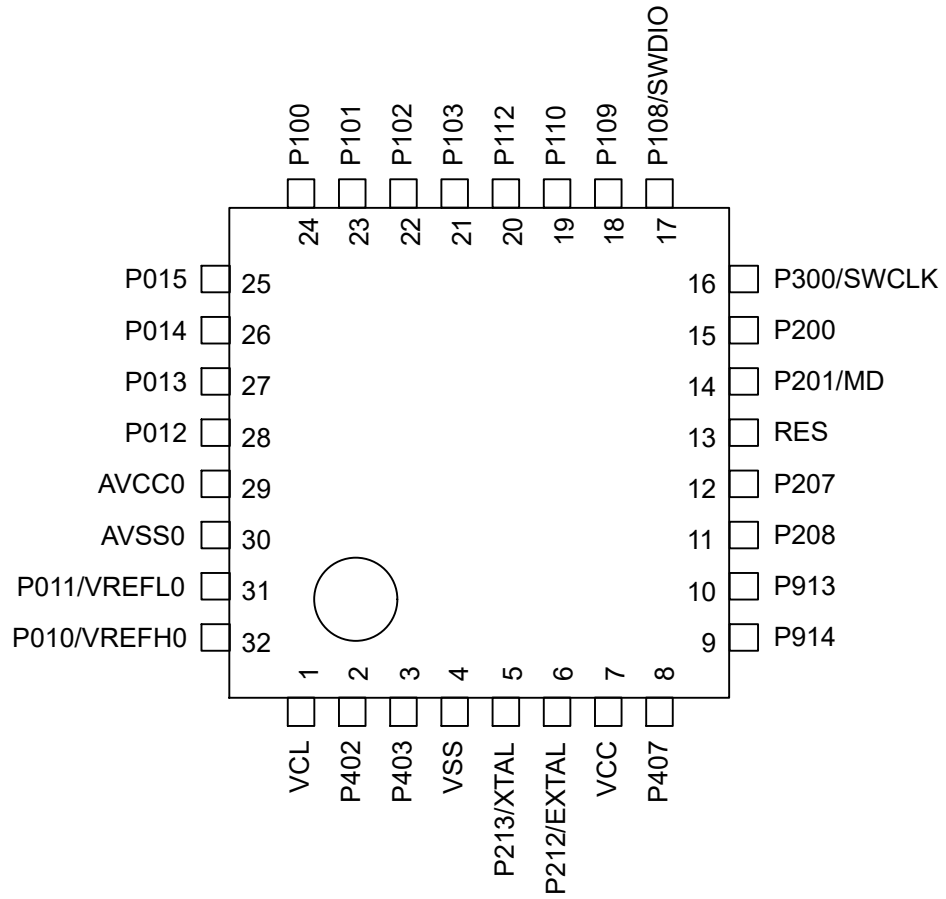
1.6 Pin Assignments

Figure 1.3 to Figure 1.5 show the pin assignments from the top view.



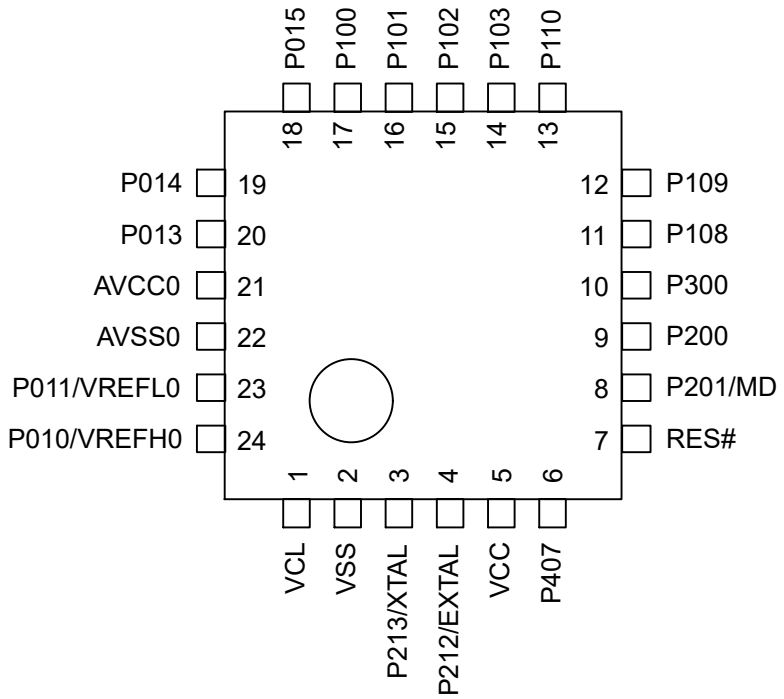
Note: For the product in a QFN package, solder the exposed die pad onto a plated area of the PCB that has no electrical connections.

Figure 1.3 Pin assignment for LQFP/QFN 48-pin (top view)



Note: For the product in a QFN package, solder the exposed die pad onto a plated area of the PCB that has no electrical connections.

Figure 1.4 Pin assignment for LQFP/QFN 32-pin (top view)



Note: For the product in a QFN package, solder the exposed die pad onto a plated area of the PCB that has no electrical connections.

Figure 1.5 Pin assignment for QFN 24-pin (top view)

1.7 Pin Lists

Table 1.14 Pin list (1 of 2)

Pin number					Timers			Communication interfaces			Analogs		HMI
LQFP/QFN 48-pin	LQFP/QFN 32-pin	LQFP/QFN 24-pin	Power, System, Clock, Debug, CAC	I/O ports	AGTW	GPT_OPS, POEG	GPT	SCI	IIC	SPI	ADC	ACMPHS	Interrupt
1	—	—	CACREF_C	P400	AGTIO1_C	—	GTIOC3A_B	SCK0_B/ SCK1_B	SCL0_A	—	—	—	IRQ0_A
2	—	—		P401	—	GTETRGA_B	GTIOC3B_B	CTS0_RTS0_B/ SS0_B	SDA0_A	—	—	—	IRQ5_C
3	1	1	VCL		—	—	—	—	—	—	—	—	—
4	2	—		P402	—	—	GTADSM0	—	—	—	—	—	—
5	3	—		P403	—	—	GTADSM1	—	—	—	—	—	—
6	4	2	VSS		—	—	—	—	—	—	—	—	—
7	5	3	XTAL	P213	—	GTOUUP_A	GTIOC0A_A	—	—	—	—	—	IRQ2_B
8	6	4	EXTAL	P212	AGTEE1_A	GTOULO_A	GTIOC0B_A	—	—	—	—	—	IRQ7_C
9	7	5	VCC		—	—	—	—	—	—	—	—	—
10	—	—		P409	—	GTIV_B	—	—	—	—	—	—	IRQ6_B
11	—	—		P408	—	GTIW_B	—	CTS1_RTS1_D/ SS1_D	SCL0_C	—	—	—	IRQ7_B
12	8	6		P407	AGTIO0_C	GTOULO_B/ GTCPPO0	GTIOC0B_B	CTS0_RTS0_D/ SS0_D	SDA0_B	—	ADTRG0_B	CMPREF1	—
13	—	—		P915	—	—	—	—	—	—	—	—	—
14	9	—		P914	AGTOA1_A	GTOVUP_A	GTIOC1A_A	—	—	—	—	—	—
15	10	—		P913	AGTIO1_F	GTOVLO_A	GTIOC1B_A	—	—	—	—	—	—
16	11	—		P208	AGTOB0_A	GTOUUP_B	GTIOC0A_B	—	—	—	—	—	—
17	12	—		P207	—	—	—	—	—	—	—	—	—
18	—	—		P206	—	GTIU_B	—	TXD0_C/ MOSI0_C/ SDA0_C	—	—	—	—	IRQ0_B
19	13	7	RES		—	—	—	—	—	—	—	—	—
20	14	8	MD	P201	—	—	—	—	—	—	—	—	—
21	15	9		P200	—	—	—	—	—	—	—	—	NMI
22	—	—		P302	—	GTETRGA_D	—	TXD2_A/ MOSI2_A/ SDA2_A	—	—	—	—	IRQ5_A
23	—	—		P301	AGTIO0_D	GTETRGB_A	—	RXD2_A/ MISO2_A/ SCL2_A/ CTS9_RTS9_D/ SS9_D	—	—	—	—	IRQ6_A
24	16	10	SWCLK	P300	—	GTETRGA_C/ GTCPPO1	GTIOC3A_A	TXD1/MOSI1/ SDA1	—	—	—	—	IRQ5_B
25	17	11	SWDIO	P108	AGTEE1_B	GTETRGB_C/ GTCPPO2	GTIOC3B_A	RXD1/MISO1/ SCL1/ CTS9_RTS9_B/ SS9_B	—	—	—	—	IRQ6_C
26	18	12	CLKOUT_B	P109	AGTOA1_B	GTOWUP_A	GTIOC2A_B	SCK1_E/ TXD9_B/ MOSI9_B/ SDA9_B	—	—	—	—	—
27	19	13	VCOUT	P110	AGTIO1_A	GTOWLO_A	GTIOC2B_B	CTS2_RTS2_B/ SS2_B/ RXD9_B/ MISO9_B/ SCL9_B	—	—	—	—	IRQ3_A
28	—	—		P111	AGTOA0	—	GTIOC1A_C	SCK2_B/ SCK9_B	—	—	—	—	IRQ4_A
29	20	—		P112	AGTOB0	—	GTIOC1B_C	SCK1_D	—	—	—	—	—
30	—	—	VCC		—	—	—	—	—	—	—	—	—
31	—	—	VSS		—	—	—	—	—	—	—	—	—
32	—	—		P104	—	GTETRGB_B	—	RXD0_C/ MISO0_C/ SCL0_C	—	SSLA1_A	—	—	KRM04/IRQ1_B
33	21	14		P103	—	GTOWUP_B/ GTCPPO3	GTIOC2A_A	CTS0_RTS0_A/ SS0_A/ RXD2_D/ MISO2_D/ SCL2_D	—	SSLA0_A	AN019	CMPIN1	KRM03/IRQ4_B

Table 1.14 Pin list (2 of 2)

Pin number			Power, System, Clock, Debug, CAC	I/O ports	Timers			Communication interfaces			Analogs		HMI
LQFP/QFN 48-pin	LQFP/QFN 32-pin	LQFP/QFN 24-pin			AGTW	GPT_OPS, POEG	GPT	SCI	IIC	SPI	ADC	ACMPHS	Interrupt
34	22	15		P102	AGTO0	GTOWLO_B	GTIOC2B_A	SCK0_A/ TXD2_D/ MOSI2_D/ SDA2_D	—	RSPCKA_A	ADTRG0_A/ AN020	CMPREF0	KRM02/IRQ1_A
35	23	16		P101	AGTEE0	GTOVUP_B	GTIOC1A_B	TXD0_A/ MOSI0_A/ SDA0_A/ CTS1_RTS1_A/ SS1_A	SDA0_C	MOSIA_A	AN021	—	KRM01/IRQ2_A
36	24	17		P100	AGTIO0_A	GTOVLO_B	GTIOC1B_B	RXD0_A/ MISO0_A/ SCL0_A/ SCK1_A	SCL0_D	MISOA_A	AN022	—	KRM00/IRQ0_C
37	—	—		P500	—	GTETRGA_A	GTIOC2A_C	—	—	—	—	—	—
38	25	18		P015	—	GTIU_A	—	—	—	—	AN002 (3ch S/H)	CMPIN01	IRQ2_C
39	26	19		P014	—	GTIV_A	—	—	—	—	AN001 (3ch S/H)	CMPIN02	IRQ3_B
40	27	20		P013	—	GTIW_A	—	—	—	—	AN000 (3ch S/H)	CMPIN03	IRQ1_C
41	28	—		P012	—	—	—	—	—	—	AN007	—	—
42	29	21	AVCC0	AVCC0	—	—	—	—	—	—	—	—	—
43	30	22	AVSS0	AVSS0	—	—	—	—	—	—	—	—	—
44	31	23	VREFL0	P011	—	—	—	—	—	—	AN006	—	—
45	32	24	VREFH0	P010	—	—	—	—	—	—	AN005	—	—
46	—	—		P002	—	—	—	—	—	—	AN010	—	—
47	—	—		P001	—	—	—	—	—	—	AN009	—	IRQ7_A
48	—	—		P000	—	—	—	—	—	—	AN008	—	—

Note: Several pin names have the added suffix of _A, _B, _C, _D, _E and _F. The suffix can be ignored when assigning functionality.

2. CPU

The MCU is based on the Arm® Cortex®-M23 core.

2.1 Overview

2.1.1 CPU

- Arm Cortex-M23
  - Revision: r1p0-00rel0
  - Armv8-M architecture profile
  - Main Extension is not implemented
  - Single-cycle integer multiplier
  - 19-cycle integer divider
- Memory Protection Unit (MPU)
  - Armv8 Protected Memory System Architecture
  - 8 protected regions
- SysTick timer
  - Driven by SYSTICCLK (LOCO) or ICLK

See reference 1. and reference 2. in section 2.8. References for details.

2.1.2 Debug

- Arm® CoreSight™ MTB-M23
  - Revision: r0p0-00rel0
  - Buffer size: 1 KB of 8-KB MTB SRAM
- Data Watchpoint Unit (DWT)
  - 2 comparators for watchpoints
- Flash Patch and Break point Unit (FPB)
  - 4 instruction comparators
- CoreSight Debug Access Port (DAP)
  - Serial Wire-Debug Port (SW-DP)
- Debug Register Module (DBGREG)
  - Reset control
  - Halt control

See reference 1. and reference 2. in section 2.8. References for details.

2.1.3 Operating Frequency

The operating frequencies for the MCU are as follows:

- CPU: maximum 64 MHz
- Serial Wire Debug (SWD) interface: maximum 12.5 MHz

2.1.4 Block Diagram

Figure 2.1 shows a block diagram of the Cortex-M23 core.

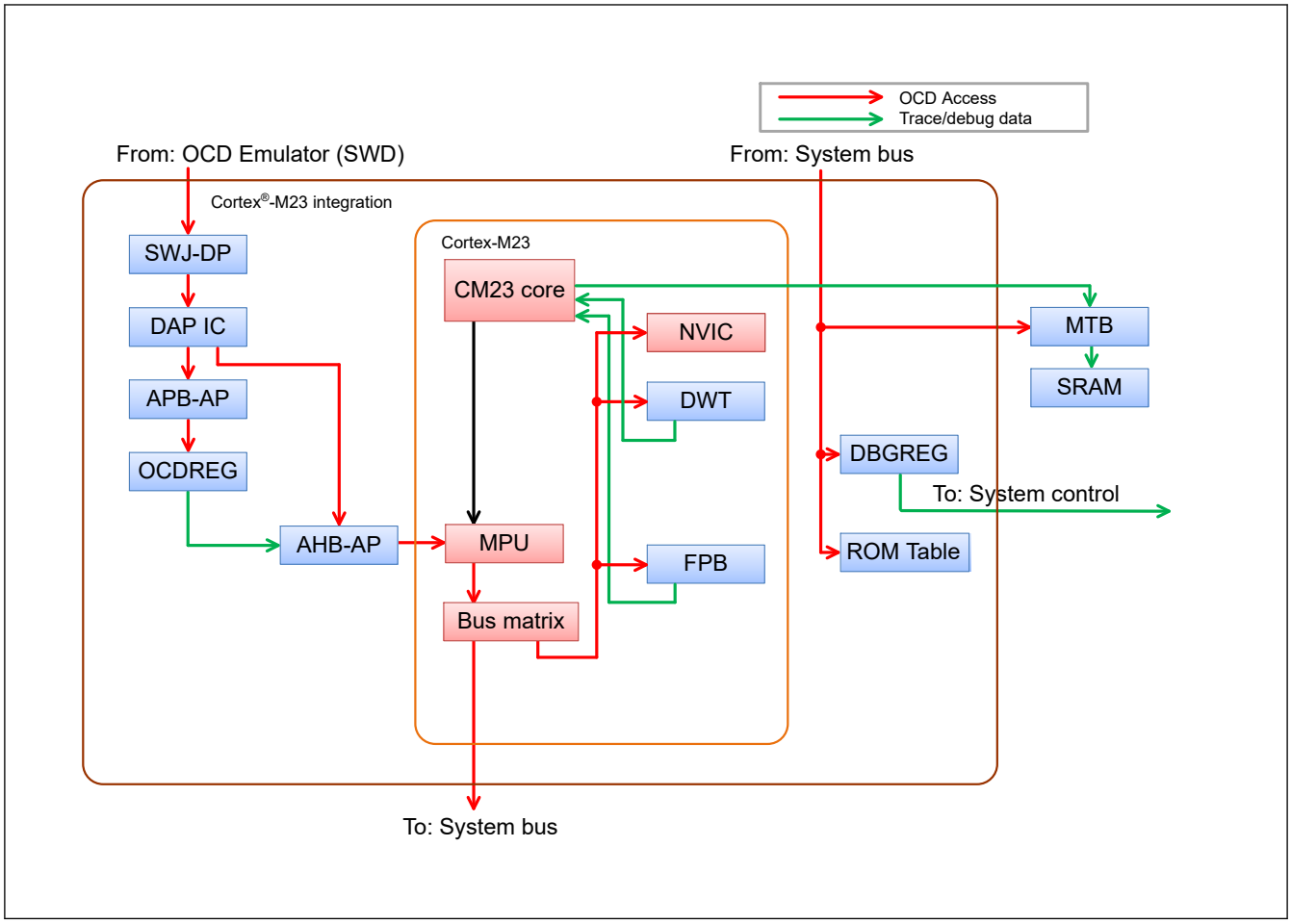


Figure 2.1 Cortex-M23 block diagram

2.2 Implementation Options

Table 2.1 shows the implementation options of the MCU.

Table 2.1 Implementation options (1 of 2)

Option	Implementation
Non-secure MPU	Included, 8 protect regions
Secure MPU	Not included
Security extension	Not included
Single-cycle multiplier	Included
Divider	Included, 19 cycles
Number of interrupts	32
Number of Wakeup Interrupt Controllers (WIC)	Not included
Cross Trigger Interface (CTI)	Not included
Micro Trace Buffer (MTB)	Included
Embedded Trace Macrocell (ETM)	Not included
Multi-drop support for serial wire	Not supported
Sleep mode power saving	Sleep mode and other low power modes are supported. For more details, see section 10, Low Power Modes. Note: SCB.SCR.SLEEPDEEP is ignored.
Endianness	Little-endian

Table 2.1 Implementation options (2 of 2)

Option	Implementation	
SysTick	Included	
SYST_CALIB register (0x4000_0147)	Bit [31] = 0 Bit [30] = 1 Bits [29:24] = 0x00 Bits [23:0] = 0x000147	Reference clock provided TENMS value is inexact Reserved TENMS: (32768 × 10 ms) - 1/32.768 kHz = 326.66 decimal = 327 with skew = 0x000147
Event input/output	Not implemented	
System reset request output	The SYSRESETREQ bit in Application Interrupt and Reset Control Register causes a CPU reset	
Auxiliary fault inputs (AUXFAULT)	Not implemented	

2.3 SWD Interface

Table 2.2 shows the SWD pins.

Table 2.2 SWD pins

Name	I/O	Function	When not in use
SWCLK	Input	Serial wire clock pin	Pull-up
SWDIO	I/O	Serial wire data I/O pin	Pull-up

2.4 Debug Function

2.4.1 Debug Mode Definition

Table 2.3 shows the CPU debug modes and usage conditions.

Table 2.3 CPU debug mode and conditions

Conditions		Mode	
OCD connect*1	SWD authentication	Debug mode	Debug authentication*2
Not connected	—	User mode	Disabled
Connected	Failed	User mode	Disabled
Connected	Passed	OCD mode	Enabled

Note 1. OCD connect is determined by the CDBGPWRUPREQ bit output in the SWJ-DP register. The bit can only be written by the OCD. However, the level of the bit can be confirmed by reading the DBGSTR.CDBGPWRUPREQ bit.
Note 2. Debug authentication is defined by the Armv8-M Architecture. Enabled means that both invasive and non-invasive CPU debugging are permitted. Disabled means that both are not permitted.

2.4.2 Debug Mode Effects

This section describes the effects of debug mode, which occur both internally and externally to the CPU.

2.4.2.1 Low power mode

All CoreSight debug components can store the register settings even when the CPU enters Software Standby, or Snooze mode. However, AHB-AP cannot respond to On-Chip Debug (OCD) access in these low power modes. The OCD must wait for cancellation of the low power mode to access the CoreSight debug components. To request low power mode cancellation, the OCD can set the DBIRQ bit in the MCUCTRL register. For details, see section 2.5.6.3. MCUCTRL : MCU Control Register.

2.4.2.2 Reset

In OCD mode, some resets depend on the CPU status and the DBGSTOPCR register setting.

Table 2.4 Reset or interrupt and mode setting

Reset or interrupt name	Control in On-Chip Debug (OCD) mode	
	OCD break mode	OCD run mode
RES pin reset	Same as user mode	
Power-on reset	Same as user mode	
Independent watchdog timer reset/interrupt	Does not occur*1	Depends on DBGSTOPCR setting
Watchdog timer reset/interrupt	Does not occur*1	Depends on DBGSTOPCR setting
Voltage monitor 0 reset	Depends on DBGSTOPCR setting	
Voltage monitor 1 reset/interrupt	Depends on DBGSTOPCR setting	
Voltage monitor 2 reset/interrupt	Depends on DBGSTOPCR setting	
SRAM parity error reset/interrupt	Depends on DBGSTOPCR setting	
Bus master MPU error reset/interrupt	Same as user mode	
Bus slave MPU error reset/interrupt	Same as user mode	
CPU stack pointer error reset/interrupt	Same as user mode	
Software reset	Same as user mode	

Note: In OCD break mode, the CPU is halted. In OCD run mode, the CPU is in OCD mode and the CPU is not halted.
Note 1. The IWDT and WDT always stop in this mode.

2.5 Programmers Model

2.5.1 Address Spaces

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

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

Figure 2.2 shows a block diagram of the AP connection and address spaces.

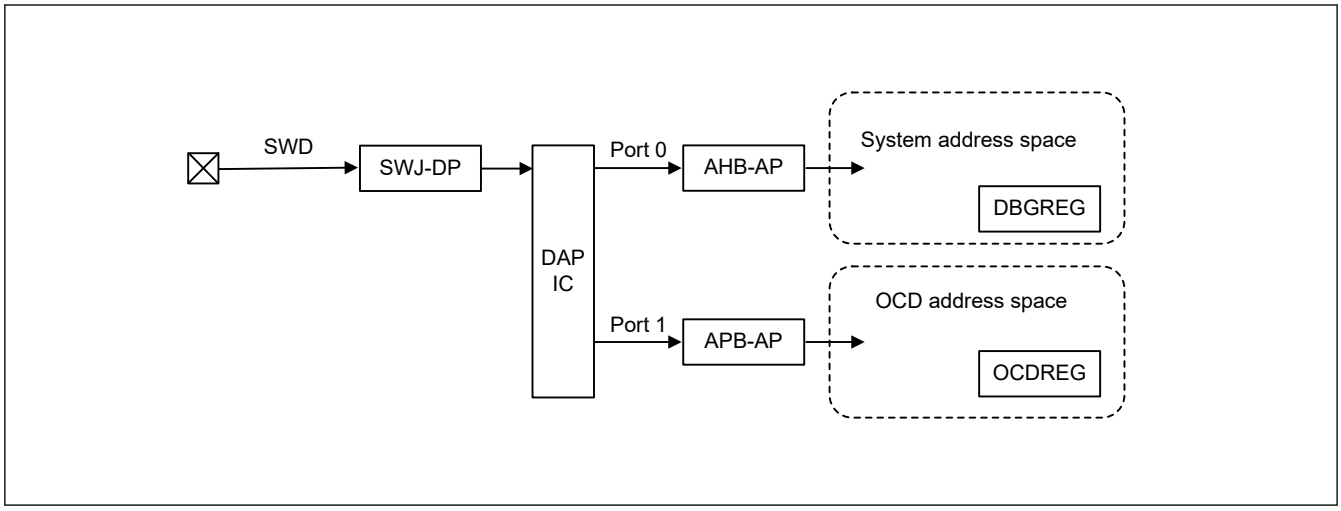


Figure 2.2 SWD authentication block diagram

For debugging purposes, there are two register modules, DBGREG and OCDREG. DBGREG is located in the system address space and can be accessed from the OCD emulator, the CPU, and other bus masters in the MCU. OCDREG is located in the OCD address space and can only be accessed from the OCD tool. The CPU and other bus masters cannot access OCDREG.

2.5.2 Cortex-M23 Peripheral Address Map

In the system address space, the Cortex-M23 core has a Private Peripheral Bus (PPB) that can only be accessed from the CPU and OCD emulator. Table 2.5 shows the address map of the MCU.

Table 2.5 Cortex-M23 peripheral address map

Component name	Start address	End address	Note
DWT	0xE000_1000	0xE000_1FFF	See reference 2. in section 2.8. References.
FPB	0xE000_2000	0xE000_2FFF	See reference 2. in section 2.8. References.
SCS	0xE000_E000	0xE000_EFFF	See reference 2. in section 2.8. References.

2.5.3 External Debug Address Map

In the system address space, the Cortex-M23 core has external debug components. These components can be accessed from the CPU and other bus masters through the system bus. Table 2.6 shows the address map of the Cortex-M23 external debug components.

Table 2.6 External debug address map

Component name	Start address	End address	Note
MTB (SRAM area)	0x2000_4000	0x2000_5FFF	MTB uses up to 1 KB as trace buffer See reference 6. in section 2.8. References.
MTB (SFR area)	0x4001_9000	0x4001_9FFF	See reference 6. in section 2.8. References.
ROM Table	0x4001_A000	0x4001_AFFF	See reference 6. in section 2.8. References.

2.5.4 CoreSight ROM Table

The MCU contains one CoreSight ROM Table, which lists all components implemented in the user area.

2.5.4.1 ROM entries

Table 2.7 shows the ROM entries. The OCD emulator can use the ROM entries to determine which components are implemented in a system. See reference 4. for details.

Table 2.7 ROM entries

#	Address	Access size	R/W	Value	Target module pointer
0	0x4001_A000	32 bits	R	0x9FFF4003	SCS
1	0x4001_A004	32 bits	R	0x9FFE7003	DWT
2	0x4001_A008	32 bits	R	0x9FFE8003	FPB
3	0x4001_A00C	32 bits	R	0xFFFFF003	MTB
4	0x4001_A010	32 bits	R	0x00000000	End of entries

2.5.4.2 CoreSight component registers

The CoreSight ROM Table lists the CoreSight component registers defined in the Arm CoreSight architecture.

Table 2.8 shows the registers. See reference 5. in section 2.8. References for details of each register.

Table 2.8 CoreSight component registers in the CoreSight ROM Table (1 of 2)

Name	Address	Access size	R/W	Initial value
MEMTYPE	0x4001_AFCC	32 bits	R	0x00000001
PIDR4	0x4001_AFD0	32 bits	R	0x00000004
PIDR5	0x4001_AFD4	32 bits	R	0x00000000
PIDR6	0x4001_AFD8	32 bits	R	0x00000000

Table 2.8 CoreSight component registers in the CoreSight ROM Table (2 of 2)

Name	Address	Access size	R/W	Initial value
PIDR7	0x4001_AFDC	32 bits	R	0x00000000
PIDR0	0x4001_AFE0	32 bits	R	0x00000069
PIDR1	0x4001_AFE4	32 bits	R	0x00000030
PIDR2	0x4001_AFE8	32 bits	R	0x0000000A
PIDR3	0x4001_AFEC	32 bits	R	0x00000000
CIDR0	0x4001_AFF0	32 bits	R	0x0000000D
CIDR1	0x4001_AFF4	32 bits	R	0x00000010
CIDR2	0x4001_AFF8	32 bits	R	0x00000005
CIDR3	0x4001_AFFC	32 bits	R	0x000000B1

2.5.5 DBGREG Module

The DBGREG module controls the debug functionalities and is implemented as a CoreSight-compliant component.

Table 2.9 shows the DBGREG registers other than the CoreSight component registers.

Table 2.9 Non-CoreSight DBGREG registers

Name		DAP port	Address	Access size	R/W
Debug Status Register	DBGSTR	Port 0	0x4001_B000	32 bits	R
Debug Stop Control Register	DBGSTOPCR	Port 0	0x4001_B010	32 bits	R/W

2.5.5.1 DBGSTR : Debug Status Register

Base address: DBG = 0x4001_B000

Offset address: 0x00

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

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

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

The DBGSTR register is a status register which indicates the state of the debug power-up request to the MCU from the emulator.

2.5.5.2 DBGSTOPCR : Debug Stop Control Register

Base address: DBG = 0x4001_B000

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	DBGSTOP_RPER	—	—	—	—	—	DBGSTOP_LVD2	DBGSTOP_LVD1	DBGSTOP_LVD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	DBGSTOP_IWDT	Mask bit for IWDT reset/interrupt in the OCD run mode In the OCD break mode, the reset/interrupt is masked and IWDT counter is stopped, regardless of this bit value. 0: Enable IWDT reset/interrupt 1: Mask IWDT reset/interrupt and stop IWDT counter	R/W
1	DBGSTOP_WDT	Mask bit for WDT reset/interrupt in the OCD run mode In the OCD break mode, the reset/interrupt is masked and WDT counter is stopped, regardless of this bit value. 0: Enable WDT reset/interrupt 1: Mask WDT reset/interrupt and stop WDT counter	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
16	DBGSTOP_LVD0	Mask bit for LVD0 reset 0: Enable LVD0 reset 1: Mask LVD0 reset	R/W
17	DBGSTOP_LVD1	Mask bit for LVD1 reset/interrupt 0: Enable LVD1 reset/interrupt 1: Mask LVD1 reset/interrupt	R/W
18	DBGSTOP_LVD2	Mask bit for LVD2 reset/interrupt 0: Enable LVD2 reset/interrupt 1: Mask LVD2 reset/interrupt	R/W
23:19	—	These bits are read as 0. The write value should be 0.	R/W
24	DBGSTOP_RPER	Mask bit for SRAM parity error reset/interrupt 0: Enable SRAM parity error reset/interrupt 1: Mask SRAM parity error reset/interrupt	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The Debug Stop Control Register (DBGSTOPCR) controls the functional stop in OCD mode. All bits in the register are regarded as 0 when the MCU is not in OCD mode.

2.5.5.3 DBGREG CoreSight component registers

The DBGREG module provides the CoreSight component registers defined in the Arm CoreSight architecture.

Table 2.10 shows the registers. See reference 4. in section 2.8. References for details of each register.

Table 2.10 DBGREG CoreSight component registers (1 of 2)

Name	Address	Access size	R/W	Initial value
PIDR4	0x4001_BFD0	32 bits	R	0x00000004
PIDR5	0x4001_BFD4	32 bits	R	0x00000000

Table 2.10 DBGREG CoreSight component registers (2 of 2)

Name	Address	Access size	R/W	Initial value
PIDR6	0x4001_BFD8	32 bits	R	0x00000000
PIDR7	0x4001_BFDC	32 bits	R	0x00000000
PIDR0	0x4001_BFE0	32 bits	R	0x00000005
PIDR1	0x4001_BFE4	32 bits	R	0x00000030
PIDR2	0x4001_BFE8	32 bits	R	0x0000001A
PIDR3	0x4001_BFEC	32 bits	R	0x00000000
CIDR0	0x4001_BFF0	32 bits	R	0x0000000D
CIDR1	0x4001_BFF4	32 bits	R	0x000000F0
CIDR2	0x4001_BFF8	32 bits	R	0x00000005
CIDR3	0x4001_BFFC	32 bits	R	0x000000B1

2.5.6 OCDREG Module

The OCDREG module is only accessible by the On-Chip Debug (OCD) emulator. OCDREG is implemented as a CoreSight-compliant component.

Table 2.11 lists the OCDREG registers.

Table 2.11 OCDREG registers

Name		DAP port	Address	Access size	R/W
ID Authentication Code Register 0	IAUTH0	Port 1	0x8000_0000	32 bits	W
ID Authentication Code Register 1	IAUTH1	Port 1	0x8000_0100	32 bits	W
ID Authentication Code Register 2	IAUTH2	Port 1	0x8000_0200	32 bits	W
ID Authentication Code Register 3	IAUTH3	Port 1	0x8000_0300	32 bits	W
MCU Status Register	MCUSTAT	Port 1	0x8000_0400	32 bits	R
MCU Control Register	MCUCTRL	Port 1	0x8000_0410	32 bits	R/W

Note: OCDREG is located in the dedicated OCD address space. This address map is independent from the system address map. See section 2.5.2. Cortex-M23 Peripheral Address Map.

2.5.6.1 IAUTHn : ID Authentication Code Register n (n = 0 to 3)

Four authentication registers are provided for a 128-bit key. These registers must be written in sequential order from IAUTHn (n = 0 to 3).

The initial value of the registers is all 0xFFFFFFFF. This means that SWD access is initially permitted when the ID code in the OSIS register has the initial value. See section 2.7.1. Unlock ID Code.

Base address: CPU_OCD = 0x8000_0000

Offset address: 0x000 + 0x100 × n

Bit position:	31																												0
Bit field:	IAUTHn: AID (32+32×n -1) to (32×n) bits																												
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

2.5.6.2 MCUSTAT : MCU Status Register

Base address: CPU_OCD = 0x8000_0000

Offset address: 0x400

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

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

Bit	Symbol	Function	R/W
0	AUTH	Authentication status 0: Authentication failed 1: Authentication succeeded	R
1	CPUSLEEP	Sleep mode status 0: CPU is not in Sleep mode 1: CPU in Sleep mode	R
2	CPUSTOPCLK	CPU clock status 0: CPU clock is not stopped. 1: CPU clock is stopped.	R
31:3	—	These bits are read as 0.	R

Note 1. Depends on the MCU status.

2.5.6.3 MCUCTRL : MCU Control Register

Base address: CPU_OCD = 0x8000_0000

Offset address: 0x410

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

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

Bit	Symbol	Function	R/W
0	EDBGRQ	External Debug Request Writing 1 to the bit causes a CPU halt. When the EDBGRQ bit is set to 0 or the CPU is halted, the EDBGRQ bit is cleared. 0: Debug event not requested 1: Debug event requested	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	DBIRQ	Debug Interrupt Request Writing 1 to the bit wakes up the MCU from low power mode. The condition can be cleared by writing 0 to the DBIRQ bit. 0: Debug interrupt not requested 1: Debug interrupt requested	R/W

Bit	Symbol	Function	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set DBIRQ and EDBGRQ to the same value.

2.5.6.4 OCDREG CoreSight component registers

The OCDREG module provides the CoreSight component registers defined in the Arm CoreSight architecture.

Table 2.12 shows the registers. See reference 4. in section 2.8. References for details of each register.

Table 2.12 OCDREG CoreSight component registers

Name	Address	Access size	R/W	Initial value
PIDR4	0x8000_0FD0	32 bits	R	0x00000004
PIDR5	0x8000_0FD4	32 bits	R	0x00000000
PIDR6	0x8000_0FD8	32 bits	R	0x00000000
PIDR7	0x8000_0FDC	32 bits	R	0x00000000
PIDR0	0x8000_0FE0	32 bits	R	0x00000004
PIDR1	0x8000_0FE4	32 bits	R	0x00000030
PIDR2	0x8000_0FE8	32 bits	R	0x0000000A
PIDR3	0x8000_0FEC	32 bits	R	0x00000000
CIDR0	0x8000_0FF0	32 bits	R	0x0000000D
CIDR1	0x8000_0FF4	32 bits	R	0x000000F0
CIDR2	0x8000_0FF8	32 bits	R	0x00000005
CIDR3	0x8000_0FFC	32 bits	R	0x000000B1

2.6 SysTick Timer

The SysTick timer provides a simple 24-bit down counter. The reference clock for the timer can be selected as the CPU clock (ICLK) or SysTick timer clock (SYSTICCLK). See section 8, Clock Generation Circuit and reference 1. in section 2.8. References for details.

2.7 OCD Emulator Connection

A SWD authentication mechanism checks access permission for debug and MCU resources. To obtain full debug functionality, a pass result of the authentication mechanism is required.

Figure 2.3 shows a block diagram of the authentication mechanism.

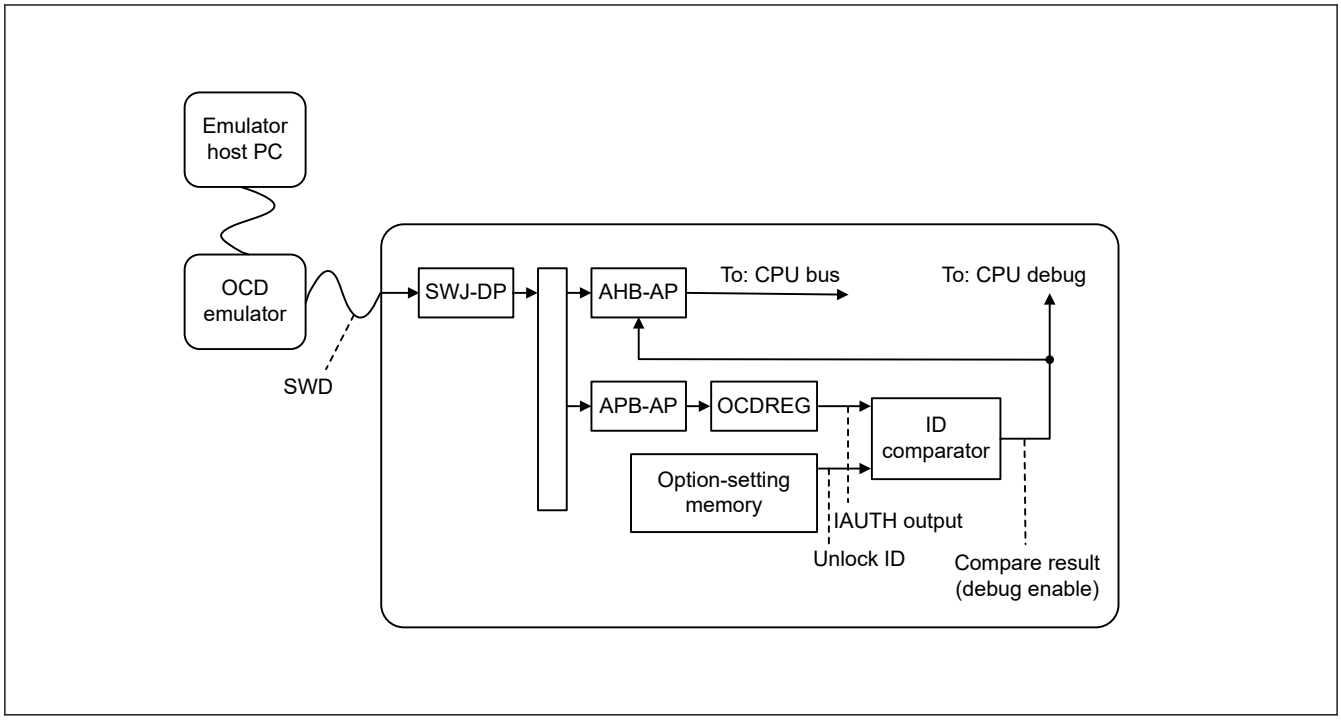


Figure 2.3 SWD Authentication mechanism block diagram

An ID comparator is available in the MCU for authentication. The comparator compares the 128-bit IAUTH output from the OCDREG and the 128-bit unlock ID code from the option-setting memory. When the two outputs are identical, the CPU debug functions and system bus access from the OCD emulator are permitted.

After the OCD emulator gets access permission, the OCD emulator must set the DBGEN bit in the System Control OCD Control Register (SYOCDCR). In addition, the OCD emulator must clear the DBGEN bit before disconnecting. See section 10.2.11. SYOCDCR : System Control OCD Control Register.

2.7.1 Unlock ID Code

The unlock ID code is used for checking permission for debug and access to on-chip resources. If the unlock ID code matches the 128-bit data written in the ID Authentication Code Registers 0 to 3, the SWD debugger obtains access permission. Unlock ID code is written in the OCD/Serial Programmer ID Setting Register (OSIS) in the option-setting memory. The initial value of the unlock ID code is all 1s (0xFFFFFFFF_FFFFFFFF_FFFFFFFF_FFFFFFFF). See section 6, Option-Setting Memory for details.

2.7.2 DBGEN

After the OCD emulator gets access permission, the OCD emulator must set the DBGEN bit in the System Control OCD Control Register (SYOCDCR). In addition, the OCD emulator must clear the DBGEN bit before disconnecting it. See section 10, Low Power Modes for details.

2.7.3 Restrictions on Connecting an OCD emulator

This section describes the restrictions on emulator access.

2.7.3.1 Starting connection while in low power mode

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

2.7.3.2 Changing low power mode while in OCD mode

When the MCU is in OCD mode, the low power mode can be changed. However, system bus access from AHB-AP is prohibited in Software Standby or Snooze mode. Only SWJ-DP, APB-AP, and OCDREG can be accessed from the OCD emulator in these modes. Table 2.13 shows the restrictions.

Table 2.13 Restrictions by mode

Active mode	Start OCD emulator connection	Change low power mode	Access AHB-AP and system bus	Access APB-AP and OCDREG
Normal	Yes	Yes	Yes	Yes
Sleep	Yes	Yes	Yes	Yes
Software Standby	No	Yes	No	Yes
Snooze	No	Yes	No	Yes

If system bus access is required in Software Standby or Snooze mode, set the MCUCTRL.DBIRQ bit in OCDREG to wake up the MCU from the low power modes. Simultaneously, by asserting the MCUCTRL.DBIRQ bit in OCDREG, the OCD emulator can wake up the MCU without starting CPU execution by using a CPU break.

2.7.3.3 Modify the unlock ID code in OSIS

After modifying the unlock ID code in the OSIS, the OCD emulator must reset the MCU by asserting the RES pin or setting the SYSRESETREQ bit of the Application Interrupt and Reset Control Register in the system control block to 1. The modified unlock ID code is reflected after reset. For the system control block, see reference 2. listed in section 2.8. References.

The emulator must set the modified unlock ID code in the IAUTH0 to IAUTH3 registers immediately before the MCU is placed in the reset state. When the IAUTH0 to IAUTH3 registers have been overwritten, writing to the SYSRESETREQ bit is not possible. Place the MCU in the reset state by asserting the signal on the RES pin.

2.7.3.4 Connecting sequence and SWD authentication

Because the OCD emulator is protected by the SWD authentication mechanism, the OCD might be required to input the ID code to the SWD authentication registers. The OSIS value in the option-setting memory determines whether the code is required. After negation of the RES pin, a wait time is required before comparing the OSIS value at cold start. See section 35.3.3. Reset Timing. The SWD authentication process is described in detail below.

(1) When MSB of the OSIS register is 0 (bit [127] = 0)

The ID code is always mismatched and connection to the OCD is prohibited.

(2) When bits in the OSIS register is all 1s (initial value)

ID authentication is not required and the OCD can use the AHB-AP without authentication. For details of the settings for using the AHB-AP, see reference 4. in section 2.8. References.

1. Connect the OCD emulator to the MCU through the SWD interface.
2. Set up SWJ-DP to access the DAP bus. In the setup, the OCD emulator must assert CDBGPWRUPREQ in the SWJ-DP Control Status Register, then wait until CDBGPWRUPACK in the same register is asserted.
3. Set up the AHB-AP to access the system address space. The AHB-AP is connected to the DAP bus port 0.
4. Start accessing the CPU debug resources using the AHB-AP.

(3) When OSIS[127:126] = 10b

ID authentication is required and the OCD must write the unlock ID code to the IAUTH registers 0 to 3 in OCDREG before using the AHB-AP.

1. Connect the OCD debugger to the MCU through the SWD interface.
2. Set up SWJ-DP to access the DAP bus. In the setup, the OCD emulator must assert CDBGPWRUPREQ in SWJ-DP Control Status Register, then wait until CDBGPWRUPACK in the same register is asserted.
3. Set up the APB-AP to access OCDREG. The APB-AP is connected to the DAP bus port 1.
4. Write the 128-bit ID code to IAUTH registers 0 to 3 in OCDREG using the APB-AP.
5. If the 128-bit ID code matches the OSIS value, the AHB-AP is authorized to issue an AHB transaction. The authorization result can be confirmed by the AUTH bit in the MCUSTAT Register or the DbgStatus bit in the AHB-AP Control Status Word Register.

- When the DbgStatus bit is 1, the 128-bit ID code is matched with the OSIS value. AHB transfers are permitted.
- When the DbgStatus bit is 0, the 128-bit ID code is not matched with the OSIS value. AHB transfers are not permitted.

6. Set up the AHB-AP to access the system address space. The AHB-AP is connected to the DAP bus port 0.
7. Start accessing the CPU debug resources using the AHB-AP.

(4) When OSIS[127:126] = 11b

OCD authentication is required and the OCD must write the unlock ID code to the IAUTH registers 0 to 3 in OCDREG. The connection sequence is the same when OSIS[127:126] is 10b except for “ALeRASE” capability.

When IATUH registers 0 to 3 are “ALeRASE” in ASCII code, the contents of the code flash, data flash, and configuration area are erased at once. See section 33, Flash Memory for details.

The ALeRASE sequence is as follows:

1. Connect the OCD debugger to the MCU through the SWD interface.
2. Set up SWJ-DP to access the DAP bus. In the setup, the OCD emulator must assert CDBGPWRUPREQ in the SWJ-DP Control Status Register, then wait until CDBGPWRUPACK in the same register is asserted.
3. Set up the APB-AP to access OCDREG. The APB-AP is connected to the DAP bus port 1.
4. Write the 128-bit ID code to IAUTH registers 0 to 3 in OCDREG using the APB-AP.
5. If the 128-bit ID code is “ALeRASE” in ASCII code (0x414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFF), the contents of the code flash, data flash, and configuration area are erased. Thereafter, the MCU transitions to Sleep mode.

2.8 References

1. *ARM®v8-M Architecture Reference Manual* (ARM DDI 0553B.a)
2. *ARM® Cortex®-M23 Processor Technical Reference Manual* (ARM DDI 0550C)
3. *ARM® Cortex®-M23 Device Generic User Guide* (ARM DUI 1095A)
4. *ARM® CoreSight™ SoC-400 Technical Reference Manual* (ARM DDI 0480G)
5. *ARM® CoreSight™ Architecture Specification* (ARM IHI 0029E)
6. *ARM® CoreSight™ MTB-M23 Technical Reference Manual* (ARM DDI 0564C)

2.9 Usage Notes

The memory cannot be debugged if the security MPU is enabled. Disable the security MPU when debugging a program. OCD emulation connection only valid when SECMPUAC register is 0xFFFF_FFFF.

3. Operating Modes

3.1 Operating Mode Types and Selection

Table 3.1 shows the selection of operating modes by the mode-setting pin. For details on each of the operating modes, see section 3.2. Details of Operating Modes. Operation starts when the on-chip flash memory is enabled, regardless of the mode in which operation started.

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

Mode-setting pin (MD)	Operating mode
1	Single-chip mode
0	SCI boot mode

3.2 Details of Operating Modes

3.2.1 Single-Chip Mode

In single-chip mode, all I/O pins are available for use as input or output port, inputs or outputs for peripheral functions, or as interrupt inputs.

When a reset is released while the MD pin is high, the MCU starts in single-chip mode and the on-chip flash is enabled.

3.2.2 SCI Boot Mode

In this mode, the on-chip flash memory programming routine (SCI boot program), stored in the boot area within the MCU, is used. The on-chip flash, including code flash memory and data flash memory, can be modified from outside the MCU by using a universal asynchronous receiver/transmitter (UART) SCI. For details, see section 33, Flash Memory. The MCU starts in SCI boot mode if the MD pin is held low on release from the reset state.

3.3 Operating Modes Transitions

3.3.1 Operating Mode Transitions as Determined by the Mode-Setting Pin

Figure 3.1 shows operating mode transitions determined by the MD pin settings.

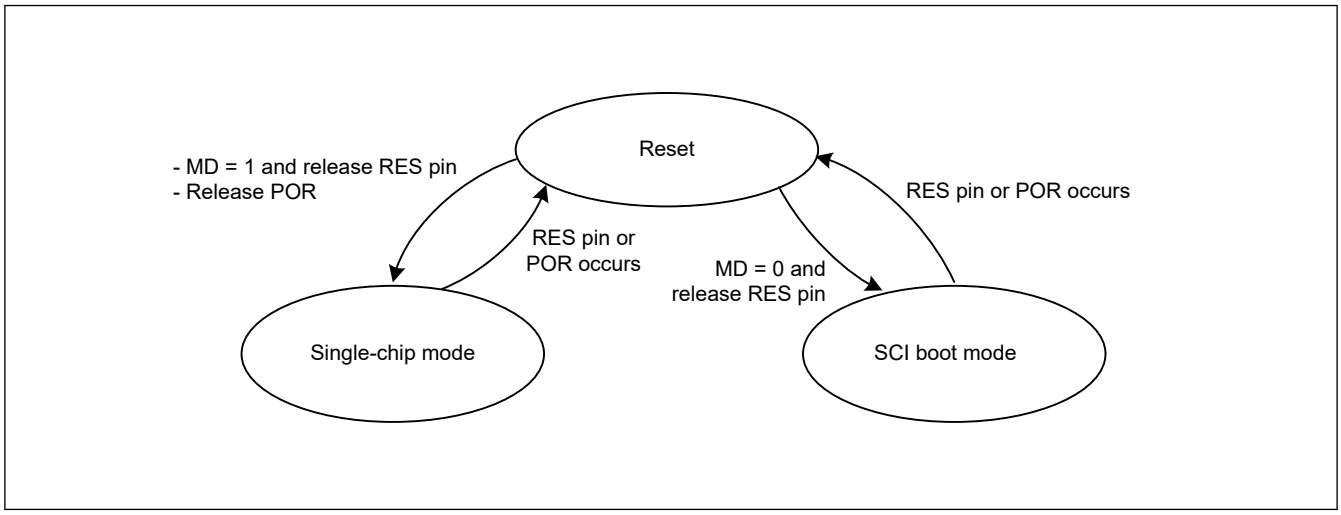
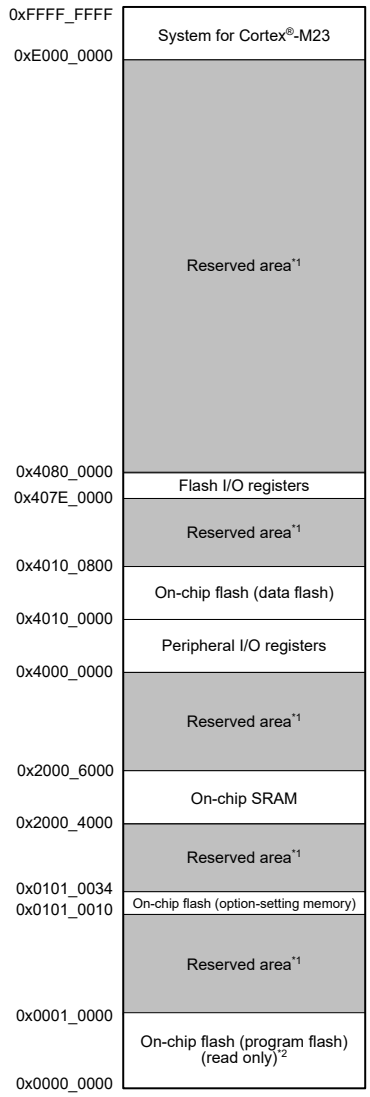


Figure 3.1 Mode-setting pin level and operating mode

4. Address Space

4.1 Address Space

The MCU supports a 4-GB linear address space ranging from 0x0000_0000 to 0xFFFF_FFFF that can contain both program and data. Figure 4.1 shows the memory map of a 64-KB flash product.



Note 1. Do not access reserved areas.
Note 2. Some regions are reserved for the option-setting memory. For details on the regions, see section 6, Option-Setting Memory.

Figure 4.1 Memory map

5. Resets

5.1 Overview

The MCU provides 12 resets. Table 5.1 lists the reset names and sources.

Table 5.1 Reset names and sources

Reset name	Source
RES pin reset	Voltage input to the RES pin is driven low
Power-on reset	VCC rise (voltage detection V_{POR})*1
Independent watchdog timer reset	IWDT underflow or refresh error
Watchdog timer reset	WDT underflow or refresh error
Voltage monitor 0 reset	VCC fall (voltage detection V_{det0})*1
Voltage monitor 1 reset	VCC fall (voltage detection V_{det1})*1
Voltage monitor 2 reset	VCC fall (voltage detection V_{det2})*1
SRAM parity error reset	SRAM parity error detection
Bus master MPU error reset	Bus master MPU error detection
Bus slave MPU error reset	Bus slave MPU error detection
CPU stack pointer error reset	CPU stack pointer error detection
Software reset	Register setting (use the Arm® software reset bit AIRCR.SYSRESETREQ)

Note 1. For details on the voltages to be monitored (V_{POR}, V_{det0}, V_{det1}, and V_{det2}), see section 7, Low Voltage Detection (LVD) and section 35, Electrical Characteristics.

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

Table 5.2 Reset detect flags initialized by each reset source (1 of 5)

Flags to be initialized	Reset source				
	RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset
Power-On Reset Detect Flag (RSTSR0.PORF)	✓	—	—	—	—
Voltage Monitor 0 Reset Detect Flag (RSTSR0.LVD0RF)	✓	✓	—	—	—
Independent Watchdog Timer Reset Detect Flag (RSTSR1.IWDTRF)	✓	✓	✓	—	—
Watchdog Timer Reset Detect Flag (RSTSR1.WDTRF)	✓	✓	✓	—	—
Voltage Monitor 1 Reset Detect Flag (RSTSR0.LVD1RF)	✓	✓	✓	—	—
Voltage Monitor 2 Reset Detect Flag (RSTSR0.LVD2RF)	✓	✓	✓	—	—
Software Reset Detect Flag (RSTSR1.SWRF)	✓	✓	✓	—	—
SRAM Parity Error Reset Detect Flag (RSTSR1.RPERF)	✓	✓	✓	—	—
Bus Slave MPU Error Reset Detect Flag (RSTSR1.BUSSRF)	✓	✓	✓	—	—
Bus Master MPU Error Reset Detect Flag (RSTSR1.BUSMRF)	✓	✓	✓	—	—
CPU Stack Pointer Error Reset Detect Flag (RSTSR1.SPERF)	✓	✓	✓	—	—

Table 5.2 Reset detect flags initialized by each reset source (2 of 5)

Flags to be initialized	Reset source				
	RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	—	✓	—	—	—

Table 5.2 Reset detect flags initialized by each reset source (3 of 5)

Flags to be initialized	Reset source			
	Voltage monitor 1 reset	Voltage monitor 2 reset	Software reset	SRAM parity error reset
Power-On Reset Detect Flag (RSTSR0.PORF)	—	—	—	—
Voltage Monitor 0 Reset Detect Flag (RSTSR0.LVD0RF)	—	—	—	—
Independent Watchdog Timer Reset Detect Flag (RSTSR1.IWDTRF)	—	—	—	—
Watchdog Timer Reset Detect Flag (RSTSR1.WDTRF)	—	—	—	—
Voltage Monitor 1 Reset Detect Flag (RSTSR0.LVD1RF)	—	—	—	—
Voltage Monitor 2 Reset Detect Flag (RSTSR0.LVD2RF)	—	—	—	—
Software Reset Detect Flag (RSTSR1.SWRF)	—	—	—	—
SRAM Parity Error Reset Detect Flag (RSTSR1.RPERF)	—	—	—	—
Bus Slave MPU Error Reset Detect Flag (RSTSR1.BUSSRF)	—	—	—	—
Bus Master MPU Error Reset Detect Flag (RSTSR1.BUSMRF)	—	—	—	—
CPU Stack Pointer Error Reset Detect Flag (RSTSR1.SPERF)	—	—	—	—
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	—	—	—	—

Table 5.2 Reset detect flags initialized by each reset source (4 of 5)

Flags to be initialized	Reset source		
	Bus master MPU error reset	Bus slave MPU error reset	CPU stack pointer error reset
Power-On Reset Detect Flag (RSTSR0.PORF)	—	—	—
Voltage Monitor 0 Reset Detect Flag (RSTSR0.LVD0RF)	—	—	—
Independent Watchdog Timer Reset Detect Flag (RSTSR1.IWDTRF)	—	—	—
Watchdog Timer Reset Detect Flag (RSTSR1.WDTRF)	—	—	—
Voltage Monitor 1 Reset Detect Flag (RSTSR0.LVD1RF)	—	—	—
Voltage Monitor 2 Reset Detect Flag (RSTSR0.LVD2RF)	—	—	—
Software Reset Detect Flag (RSTSR1.SWRF)	—	—	—
SRAM Parity Error Reset Detect Flag (RSTSR1.RPERF)	—	—	—
Bus Slave MPU Error Reset Detect Flag (RSTSR1.BUSSRF)	—	—	—
Bus Master MPU Error Reset Detect Flag (RSTSR1.BUSMRF)	—	—	—

Table 5.2 Reset detect flags initialized by each reset source (5 of 5)

Flags to be initialized	Reset source		
	Bus master MPU error reset	Bus slave MPU error reset	CPU stack pointer error reset
CPU Stack Pointer Error Reset Detect Flag (RSTSR1.SPERF)	—	—	—
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	—	—	—

Note: ✓ : Initialized to 0
— : Not initialized

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

Registers to be initialized		Reset source				
		RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset
Registers related to the IWDT	IWDTRR, IWDTSR	✓	✓	✓	✓	✓
Registers related to the WDT	WDTRR, WDTCR, WDTSR, WDTRCR, WDTCSTPR	✓	✓	✓	✓	✓
Registers related to the voltage monitor function 1	LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL	✓	✓	✓	✓	✓
	LVD1CR1/LVD1SR	✓	✓	✓	✓	✓
Registers related to the voltage monitor function 2	LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL	✓	✓	✓	✓	✓
	LVD2CR1/LVD2SR	✓	✓	✓	✓	✓
Register related to the LOCO	LOCOCR	✓	✓	✓	✓	✓
	LOCOUTCR	—	✓	✓	—	—
Register related to the MOSC	MOMCR	✓	✓	✓	✓	✓
Register related to the AGT		—	✓	✓	—	—
Register related to the MPU		✓	✓	✓	✓	✓
Pin state		✓	✓	✓	✓	✓
Registers other than those shown, CPU, and internal state		✓	✓	✓	✓	✓

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

Registers to be initialized		Reset source			
		Voltage monitor 1 reset	Voltage monitor 2 reset	Software reset	SRAM parity error reset
Registers related to the IWDT	IWDTRR, IWDTSR	✓	✓	✓	✓
Registers related to the WDT	WDTRR, WDTCR, WDTSR, WDTRCR, WDTCSTPR	✓	✓	✓	✓
Registers related to the voltage monitor function 1	LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL	—	—	—	—
	LVD1CR1/LVD1SR	—	—	—	—

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

Registers to be initialized		Reset source			
		Voltage monitor 1 reset	Voltage monitor 2 reset	Software reset	SRAM parity error reset
Registers related to the voltage monitor function 2	LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL	—	—	—	—
	LVD2CR1/LVD2SR	—	—	—	—
Register related to the LOCO	LOCOCR	✓	✓	✓	✓
	LOCOUTCR	✓	✓	—	—
Register related to the MOSC	MOMCR	✓	✓	✓	✓
Register related to the AGT		✓	✓	—	—
Register related to the MPU		✓	✓	✓	✓
Pin state		✓	✓	✓	✓
Registers other than those shown, CPU, and internal state		✓	✓	✓	✓

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

Registers to be initialized		Reset source		
		Bus master MPU error reset	Bus slave MPU error reset	CPU stack pointer error reset
Registers related to the IWDT	IWDTRR, IWDTSR	✓	✓	✓
Registers related to the WDT	WDTRR, WDTCR, WDTSR, WDTRCR, WDTCSTPR	✓	✓	✓
Registers related to the voltage monitor function 1	LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL	—	—	—
	LVD1CR1/LVD1SR	—	—	—
Registers related to the voltage monitor function 2	LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL	—	—	—
	LVD2CR1/LVD2SR	—	—	—
Register related to the LOCO	LOCOCR	✓	✓	✓
	LOCOUTCR	—	—	—
Register related to the MOSC	MOMCR	✓	✓	✓
Register related to the AGT		—	—	—
Register related to the MPU		—	—	—
Pin state		✓	✓	✓
Registers other than those shown, CPU, and internal state		✓	✓	✓

Note: ✓ : Initialized
— : Not initialized

Table 5.4 shows the states of LOCO when a reset occurs.

Table 5.4 States of LOCO when a reset occurs

		Reset source	
		POR/LVD0/LVD1/LVD2	Other
LOCO	Enable or disable	Initialized to enable	

When a reset is canceled, reset exception handling starts.

Table 5.5 lists the pin related to the reset function.

Table 5.5 Pin related to reset

Pin name	I/O	Function
RES	Input	Reset pin

5.2 Register Descriptions

5.2.1 RSTSR0 : Reset Status Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x410

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	LVD2RF	LVD1RF	LVD0RF	PORF
Value after reset:	0	0	0	0	x*1	x*1	x*1	x*1

Bit	Symbol	Function	R/W
0	PORF	Power-On Reset Detect Flag 0: Power-on reset not detected 1: Power-on reset detected	R/W*2
1	LVD0RF	Voltage Monitor 0 Reset Detect Flag 0: Voltage monitor 0 reset not detected 1: Voltage monitor 0 reset detected	R/W*2
2	LVD1RF	Voltage Monitor 1 Reset Detect Flag 0: Voltage monitor 1 reset not detected 1: Voltage monitor 1 reset detected	R/W*2
3	LVD2RF	Voltage Monitor 2 Reset Detect Flag 0: Voltage monitor 2 reset not detected 1: Voltage monitor 2 reset detected	R/W*2
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the reset source.
Note 2. The register is cleared when a reset source listed in section 5.1. Overview occurs or when 0 is written to clear a flag. Bits other than the flag that is cleared should be set to 1.

PORF flag (Power-On Reset Detect Flag)

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

[Setting condition]

- When a power-on reset occurs.

[Clearing conditions]

- When a reset listed in section 5.1. Overview occurs
- When PORF is read as 1 and then 0 is written to PORF

LVD0RF flag (Voltage Monitor 0 Reset Detect Flag)

The LVD0RF flag indicates that the VCC voltage fell below V_{det0}.

[Setting condition]

- When a voltage monitor 0 reset occurs.

[Clearing conditions]

- When a reset listed in section 5.1. Overview occurs

- When LVD0RF is read as 1 and then 0 is written to LVD0RF.

LVD1RF flag (Voltage Monitor 1 Reset Detect Flag)

The LVD1RF flag indicates that the VCC voltage fell below V_{det1}.

[Setting condition]

- When a voltage monitor 1 reset occurs.

[Clearing conditions]

- When a reset listed in section 5.1. Overview occurs
- When LVD1RF is read as 1 and then 0 is written to LVD1RF

LVD2RF flag (Voltage Monitor 2 Reset Detect Flag)

The LVD2RF flag indicates that the VCC voltage fell below V_{det2}.

[Setting condition]

- When a voltage monitor 2 reset occurs.

[Clearing conditions]

- When a reset listed in section 5.1. Overview occurs
- When LVD2RF is read as 1 and then 0 is written to LVD2RF

5.2.2 RSTSR1 : Reset Status Register 1

Base address: SYSC = 0x4001_E000

Offset address: 0x0C0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SPERF	BUSMRF	BUSSRF	—	RPERF	—	—	—	—	—	SWRF	WDTRF	IWDTRF
Value after reset:	0	0	0	x*1	x*1	x*1	0	x*1	0	0	0	0	0	x*1	x*1	x*1

Bit	Symbol	Function	R/W
0	IWDTRF	Independent Watchdog Timer Reset Detect Flag 0: Independent watchdog timer reset not detected 1: Independent watchdog timer reset detected	R/W*2
1	WDTRF	Watchdog Timer Reset Detect Flag 0: Watchdog timer reset not detected 1: Watchdog timer reset detected	R/W*2
2	SWRF	Software Reset Detect Flag 0: Software reset not detected 1: Software reset detected	R/W*2
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	RPERF	SRAM Parity Error Reset Detect Flag 0: SRAM parity error reset not detected 1: SRAM parity error reset detected	R/W*2
9	—	This bit is read as 0. The write value should be 0.	R/W
10	BUSSRF	Bus Slave MPU Error Reset Detect Flag 0: Bus slave MPU error reset not detected 1: Bus slave MPU error reset detected	R/W*2
11	BUSMRF	Bus Master MPU Error Reset Detect Flag 0: Bus master MPU error reset not detected 1: Bus master MPU error reset detected	R/W*2

Bit	Symbol	Function	R/W
12	SPERF	CPU Stack Pointer Error Reset Detect Flag 0: CPU stack pointer error reset not detected 1: CPU stack pointer error reset detected	R/W*2
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the reset source.
Note 2. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 after 1 is read.

IWDTRF flag (Independent Watchdog Timer Reset Detect Flag)

The IWDTRF flag indicates that an independent watchdog timer reset occurs.

[Setting condition]

- When an independent watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in section 5.1. Overview occurs
- When 1 is read and then 0 is written to IWDTRF.

WDTRF flag (Watchdog Timer Reset Detect Flag)

The WDTRF flag indicates that a watchdog timer reset occurs.

[Setting condition]

- When a watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in section 5.1. Overview occurs
- When 1 is read and then 0 is written WDTRF.

SWRF flag (Software Reset Detect Flag)

The SWRF flag indicates that a software reset occurs.

[Setting condition]

- When a software reset occurs.

[Clearing conditions]

- When a reset listed in section 5.1. Overview occurs
- When 1 is read and then 0 is written to SWRF.

RPERF flag (SRAM Parity Error Reset Detect Flag)

The RPERF flag indicates that an SRAM parity error reset occurs.

[Setting condition]

- When an SRAM parity error reset occurs.

[Clearing conditions]

- When a reset listed in section 5.1. Overview occurs
- When 1 is read as 1 and then 0 is written to RPERF.

BUSSRF flag (Bus Slave MPU Error Reset Detect Flag)

The BUSSRF flag indicates that a bus slave MPU error reset occurs.

[Setting condition]

- When a bus slave MPU error reset occurs.

[Clearing conditions]

- When a reset listed in section 5.1. Overview occurs
- When 1 is read and then 0 is written to BUSSRF.

BUSMRF flag (Bus Master MPU Error Reset Detect Flag)

The BUSMRF flag indicates that a bus master MPU error reset occurs.

[Setting condition]

- When a bus master MPU error reset occurs.

[Clearing conditions]

- When a reset listed in section 5.1. Overview occurs
- When 1 is read and then 0 is written to BUSMRF.

SPERF flag (CPU Stack Pointer Error Reset Detect Flag)

The SPERF flag indicates that a stack pointer error reset occurs.

[Setting condition]

- When a stack pointer error reset occurs.

[Clearing conditions]

- When a reset listed in section 5.1. Overview occurs
- When 1 is read and then 0 is written to SPERF.

5.2.3 RSTSR2 : Reset Status Register 2

Base address: SYSC = 0x4001_E000

Offset address: 0x411

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

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

Note 1. The value after reset depends on the reset source.
Note 2. Only 1 can be written to set the flag.

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

CWSF flag (Cold/Warm Start Determination Flag)

The CWSF flag indicates the type of reset processing, either cold start or warm start. The determines whether a power-on reset caused the reset processing (cold start) or a reset signal input during operation caused the reset processing (warm start). CWSF flag is initialized by a power-on reset. It is not initialized by a reset signal generated by the RES pin.

[Setting condition]

- When 1 is written by software.

[Clearing condition]

- When a reset listed in section 5.1. Overview occurs.

5.3 Operation

5.3.1 RES Pin Reset

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

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

For details, see section 35, Electrical Characteristics.

5.3.2 Power-On Reset

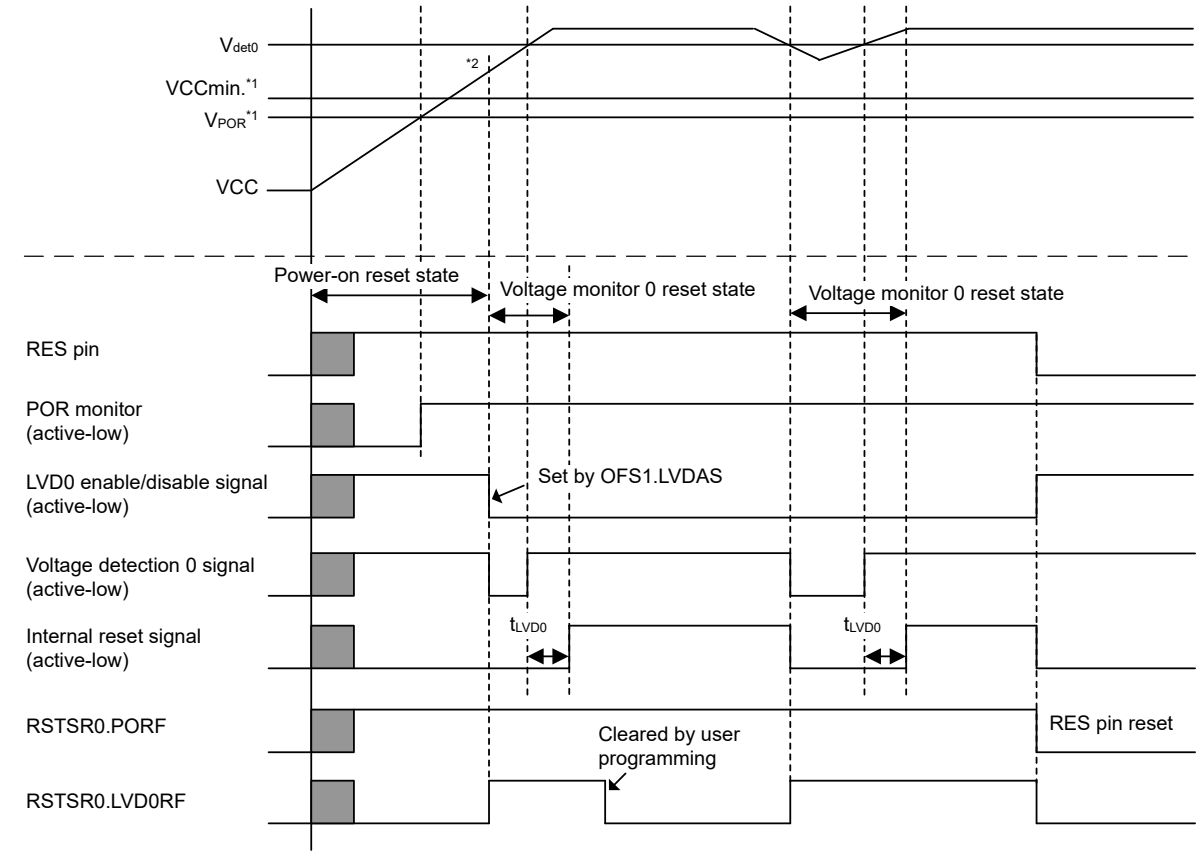
The power-on reset (POR) is an internal reset generated by the power-on reset circuit. A power-on reset is generated under the following conditions.

1. If the RES pin is in a high level state when power is supplied
2. If the RES pin is in a high level state when VCC is below V_{POR}

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

After a power-on reset is generated, the PORF flag in the RSTSR0 is set to 1. The PORF flag is initialized by the RES pin reset. When VCC falls below V_{POR}, a power-on reset state is occurred.

Figure 5.1 shows example of operations during a power-on reset.



Note: For details on the electrical characteristics, see section 35, Electrical Characteristics.
Note 1. V_{POR} shows a power-on reset detection level, and VCCmin shows the minimum guaranteed voltage of MCU.
Note 2. At power-on, VCC should rise to the minimum guaranteed voltage before the power-on reset is released.

Figure 5.1 Example of operations during a power-on reset

5.3.3 Voltage Monitor Reset

The voltage monitor i (i = 0, 1, 2) reset is an internal reset generated by the voltage monitor i circuit. If the Voltage Detection 0 Circuit Start (LVDAS) bit in the Option Function Select Register 1 (OFS1) is 0 (voltage monitor 0 reset is enabled after a reset) and VCC falls below V_{det0}, the RSTSR0.LVD0RF flag becomes 1 and the voltage detection circuit generates voltage monitor 0 reset. Clear the OFS1.LVDAS bit to 0 if the voltage monitor 0 reset is to be used. After VCC exceeds V_{det0} and the voltage monitor 0 reset time (t_{LVD0}) elapses, the internal reset is canceled and the CPU starts the reset exception handling.

When the Voltage Monitor 1 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 1 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 1 Circuit Control Register 0 (LVD1CR0), the RSTSR0.LVD1RF flag is set to 1 and the voltage detection circuit generates a voltage monitor 1 reset if VCC falls to or below V_{det1}.

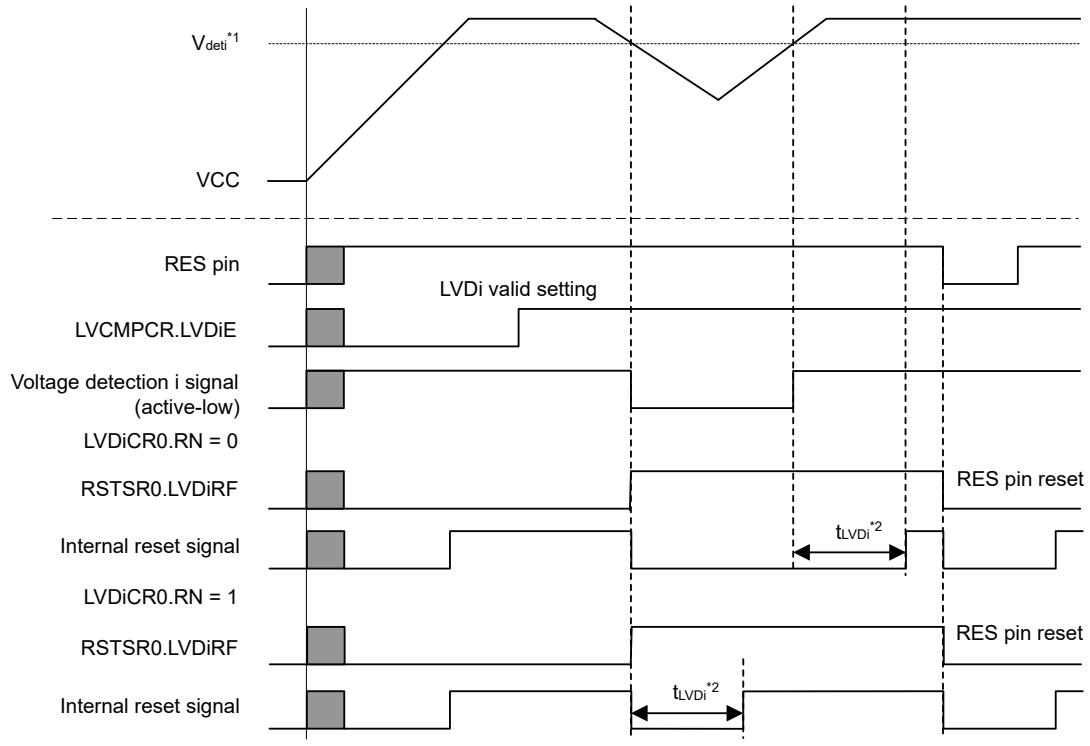
Likewise, when the Voltage Monitor 2 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 2 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 2 Circuit Control Register 0 (LVD2CR0), the RSTSR0.LVD2RF flag is set to 1 and the voltage detection circuit generates a voltage monitor 2 reset if VCC falls to or below V_{det2}.

Similarly, timing for release from the voltage monitor 1 reset state is selectable with the Voltage Monitor 1 Reset Negate Select bit (RN) in the LVD1CR0. When the LVD1CR0.RN bit is 0 and VCC falls to or below V_{det1}, the CPU is released from the internal reset state and starts reset exception handling when the LVD1 reset time (t_{LVD1}) elapses after VCC rises above V_{det1}. When the LVD1CR0.RN bit is 1 and VCC falls to or below V_{det1}, the CPU is released from the internal reset state and starts reset exception handling when the LVD1 reset time (t_{LVD1}) elapses.

Likewise, timing for release from the voltage monitor 2 reset state is selectable by setting the Voltage Monitor 2 Reset Negate Select bit (RN) in the LDV2CR0 register.

Detection levels V_{det1} and V_{det2} can be changed in the Voltage Detection Level Select Register (LVDLVLR).

Figure 5.2 shows example of operations during voltage monitor 1 and 2 resets. For details on the voltage monitor 1 reset and voltage monitor 2 reset, see section 7, Low Voltage Detection (LVD).



Note: For details on the electrical characteristics, see section 35, Electrical Characteristics.
Note 1. V_{deti} indicates the detection level of a voltage monitor 1 reset and a voltage monitor 2 reset (i = 1, 2).
Note 2. t_{LVDi} indicates the time for a voltage monitor 1 reset and a voltage monitor 2 reset (i = 1, 2).

Figure 5.2 Example of operations during voltage monitor 1 and voltage monitor 2 resets

5.3.4 Independent Watchdog Timer Reset

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

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

For details on the independent watchdog timer reset, see section 23, Independent Watchdog Timer (IWDT).

5.3.5 Watchdog Timer Reset

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

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

For details on the watchdog timer reset, see section 22, Watchdog Timer (WDT).

5.3.6 Software Reset

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

For details on the SYSRESETREQ bit, see the *ARM® Cortex®-M23 Technical Reference Manual*.

5.3.7 Determination of Cold/Warm Start

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

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

Figure 5.3 shows an example of cold/warm start determination operation.

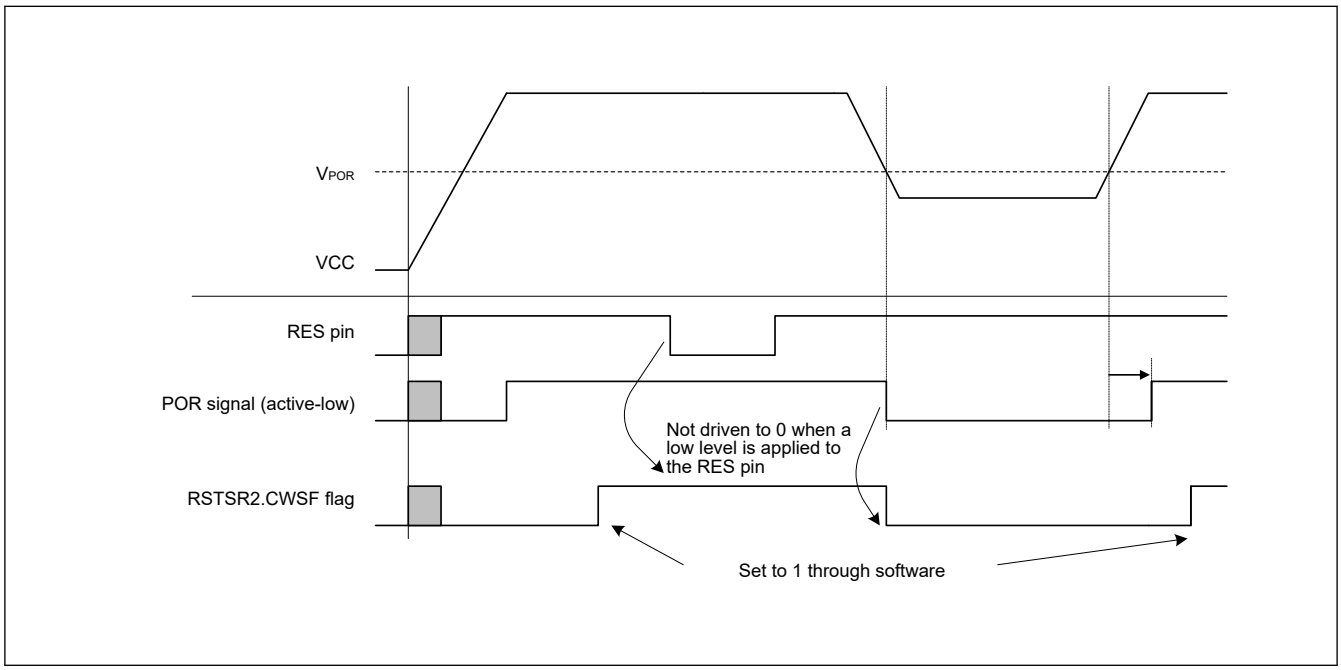
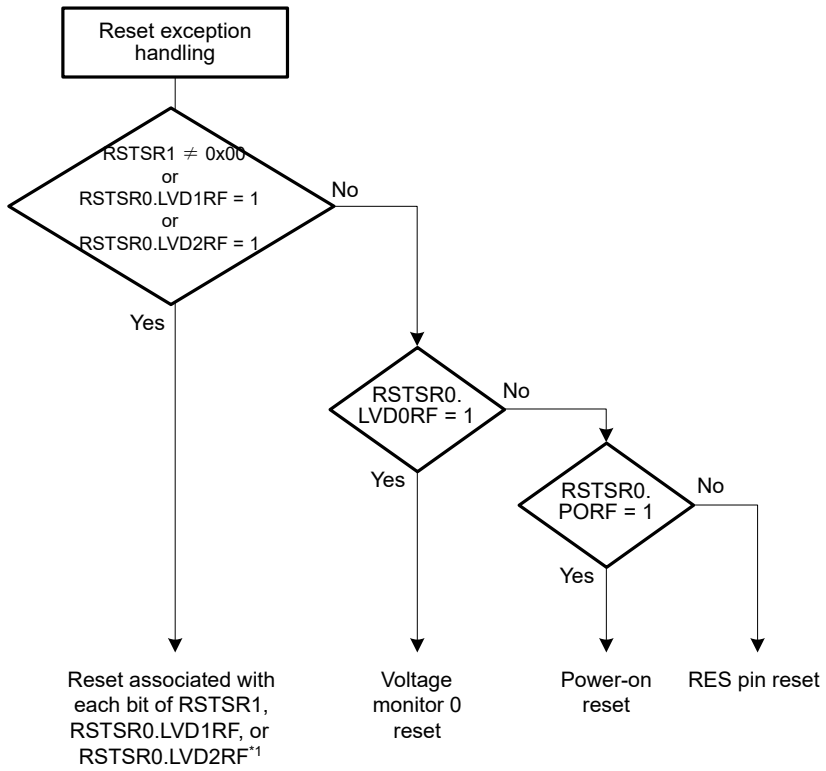


Figure 5.3 Example of cold/warm start determination operation

5.3.8 Determination of Reset Generation Source

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

Figure 5.4 shows an example of the flow to identify a reset generation source. The reset flag must be written with 0 after it is read as 1.



Note 1. When resets are generated simultaneously by two or more sources, the two or more corresponding reset flags will be set to 1.

Figure 5.4 Example of reset generation source determination flow

5.4 Usage Notes

5.4.1 Note on RES pin reset

A power-on reset may occur if RES pin reset is used with the following condition.

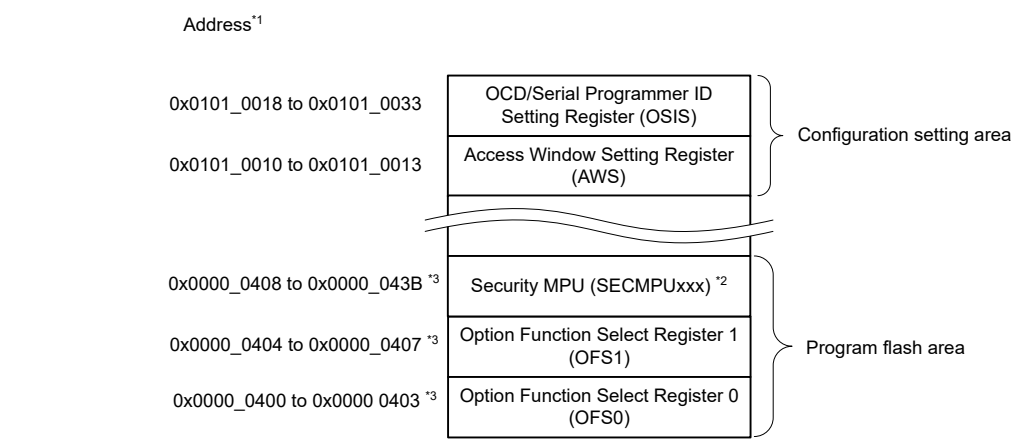
- When VCC ≤ 1.7V
- Voltage detection 0 circuit is enabled.

6. Option-Setting Memory

6.1 Overview

The option-setting memory determines the state of the MCU after a reset. The Option-setting memory is allocated to the configuration setting area and the program flash area of the flash memory. The available methods of setting are different for the two areas.

Figure 6.1 shows the option-setting memory area.



Note 1. The option-setting memory must be allocated to the user area of the flash memory.
Note 2. See Table 6.2 for details.
Note 3. The address of these registers will be changed When the boot swap is set. See section 6.2.1. OFS0 : Option Function Select Register 0, section 6.2.2. OFS1 : Option Function Select Register 1 and section 6.2.3. MPU Registers for details.

Figure 6.1 Option-setting memory area

6.2 Register Descriptions

6.2.1 OFS0 : Option Function Select Register 0

Address: 0x0000_0400 and 0x0000_2400*1

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	WDTSTPCTL	—	WDTRSTIRQS	WDTRPSS[1:0]		WDTRPES[1:0]		WDTCKS[3:0]				WDTTOPS[1:0]		WDTSTRT	—
Value after reset:	User setting*2															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	IWDTSTPCTL	—	IWDTRSTIRQS	IWDTRPSS[1:0]		IWDTRPES[1:0]		IWDTCKS[3:0]				IWDTTOPS[1:0]		IWDTSTRT	—
Value after reset:	User setting*2															

Bit	Symbol	Function	R/W
0	—	When read, this bit returns the written value.	R
1	IWDTSTRT	IWDT Start Mode Select 0: Automatically activate IWDT after a reset (auto start mode) 1: Disable IWDT after a reset	R

Bit	Symbol	Function	R/W
3:2	IWDTTOPS[1:0]	IWDT Timeout Period Select 0 0: 128 cycles (0x007F) 0 1: 512 cycles (0x01FF) 1 0: 1024 cycles (0x03FF) 1 1: 2048 cycles (0x07FF)	R
7:4	IWDTCKS[3:0]	IWDT-Dedicated Clock Frequency Division Ratio Select 0x0: × 1 0x2: × 1/16 0x3: × 1/32 0x4: × 1/64 0xF: × 1/128 0x5: × 1/256 Others: Setting prohibited	R
9:8	IWDTRPES[1:0]	IWDT Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (no window end position setting)	R
11:10	IWDTRPSS[1:0]	IWDT Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (no window start position setting)	R
12	IWDTRSTIRQS	IWDT Reset Interrupt Request Select 0: Interrupt 1: Reset	R
13	—	When read, this bit returns the written value.	R
14	IWDTSTPCTL	IWDT Stop Control 0: Continue counting 1: Stop counting when in Sleep, Snooze, or Software Standby mode	R
16:15	—	When read, these bits return the written value.	R
17	WDTSTRT	WDT Start Mode Select 0: Automatically activate WDT after a reset (auto start mode) 1: Stop WDT after a reset (register start mode)	R
19:18	WDTTOPS[1:0]	WDT Timeout Period Select 0 0: 1024 cycles (0x03FF) 0 1: 4096 cycles (0x0FFF) 1 0: 8192 cycles (0x1FFF) 1 1: 16384 cycles (0x3FFF)	R
23:20	WDTCKS[3:0]	WDT Clock Frequency Division Ratio Select 0x1: WDTCLK divided by 4 0x4: WDTCLK divided by 64 0xF: WDTCLK divided by 128 0x6: WDTCLK divided by 512 0x7: WDTCLK divided by 2048 0x8: WDTCLK divided by 8192 Others: Setting prohibited	R
25:24	WDTRPES[1:0]	WDT Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (no window end position setting)	R
27:26	WDTRPSS[1:0]	WDT Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (no window start position setting)	R

Bit	Symbol	Function	R/W
28	WDTRSTIRQS	WDT Reset Interrupt Request Select 0: Interrupt 1: Reset	R
29	—	When read, these bits return the written value.	R
30	WDTSTPCTL	WDT Stop Control 0: Continue counting 1: Stop counting when entering Sleep mode	R
31	—	When read, these bits return the written value.	R

Note 1. When the boot swap is set, the address of this register changes. Therefore, set 0x0000_2400 and 0x0000_0400 to the same value if boot swap is used.
Note 2. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

IWDTSTRT bit (IWDT Start Mode Select)

The IWDTSTRT bit selects the mode in which the IWDT is activated after a reset (stopped state or activated state).

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

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

For details, see section 23, Independent Watchdog Timer (IWDT).

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

The IWDTCKS[3:0] bits specify the division ratio of the prescaler for dividing the frequency of the clock for the IWDT as 1/1, 1/16, 1/32, 1/64, 1/128, and 1/256. Using this setting combined with the IWDTTOPS[1:0] bits setting, the IWDT counting period can be set from 128 to 524288 IWDT clock cycles.

For details, see section 23, Independent Watchdog Timer (IWDT).

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

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

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

For details, see section 23, Independent Watchdog Timer (IWDT).

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

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

For details, see section 23, Independent Watchdog Timer (IWDT).

IWDTRSTIRQS bit (IWDT Reset Interrupt Request Select)

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

For details, see section 23, Independent Watchdog Timer (IWDT).

IWDTSTPCTL bit (IWDT Stop Control)

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

Table 6.1 shows the count stop control by the IWDTSTPCTL bit.

Table 6.1 Count Stop Control by the IWDTSTPCTL Bit

IWDTSTPCTL	Mode	Counting of IWDT
0	Sleep / snooze/ software standby mode	Continue counting
1	Sleep / snooze / software standby mode	Stop counting

For details, see section 23, Independent Watchdog Timer (IWDT).

WDTSTRT bit (WDT Start Mode Select)

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

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

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

For details, see section 22, Watchdog Timer (WDT).

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

The WDTCKS[3:0] bits specify the division ratio of the prescaler for dividing the frequency of WDTCLK as 1/4, 1/64, 1/128, 1/512, 1/2048, and 1/8192. Using this setting combined with the WDTTOPS[1:0] bits setting, the WDT counting period can be set from 4096 to 134217728 WDTCLK cycles.

For details, see section 22, Watchdog Timer (WDT).

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

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

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

For details, see section 22, Watchdog Timer (WDT).

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

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

Refresh is not possible outside this period.

For details, see section 22, Watchdog Timer (WDT).

WDTRSTIRQS bit (WDT Reset Interrupt Request Select)

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

For details, see section 22, Watchdog Timer (WDT).

WDTSTPCTL bit (WDT Stop Control)

The WDTSTPCTL bit specifies whether to stop counting when entering Sleep mode.

For details, see section 22, Watchdog Timer (WDT).

6.2.2 OFS1 : Option Function Select Register 1

Address: 0x0000_0404 and 0x0000_2404*1

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ICSATS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*2															

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	HOCOFRQ1[2:0]			—	—	—	HOCOEN	—	—	VDSEL0[2:0]			LVDAS	—	—
Value after reset:	The value set by the user*2															

Bit	Symbol	Function	R/W
1:0	—	When read, these bits return the written value.	R
2	LVDAS	Voltage Detection 0 Circuit Start 0: Enable voltage monitor 0 reset after a reset 1: Disable voltage monitor 0 reset after a reset	R
5:3	VDSEL0[2:0]	Voltage Detection 0 Level Select*3 0 0 0: V_{det0_0} 0 0 1: V_{det0_1} 0 1 0: V_{det0_2} 0 1 1: V_{det0_3} 1 0 0: V_{det0_4} Others: Setting prohibited	R
7:6	—	When read, these bits return the written value.	R
8	HOCOEN	HOCO Oscillation Enable 0: Enable HOCO oscillation after a reset 1: Disable HOCO oscillation after a reset	R
11:9	—	When read, these bits return the written value.	R
14:12	HOCOFRQ1[2:0]	HOCO Frequency Setting 1 0 0 0: 24 MHz 0 1 0: 32 MHz 1 0 0: 48 MHz 1 0 1: 64 MHz Others: Setting prohibited	R
30:15	—	When read, these bits return the written value.	R
31	ICSATS	Internal Clock Supply Architecture Type Select 0: Internal Clock Supply Architecture Type B 1: Internal Clock Supply Architecture Type A	R

Note 1. When the boot swap is set, the address of this register changes. Therefore, set 0x0000_2404 and 0x0000_0404 to the same value if boot swap is used.
Note 2. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.
Note 3. See section 35, Electrical Characteristics for the voltage levels to be detected. Set to 100b if LVD0 is not used.

LVDAS bit (Voltage Detection 0 Circuit Start)

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

VDSEL0[2:0] bits (Voltage Detection 0 Level Select)

The VDSEL0[2:0] bits select the voltage detection level of the voltage detection 0 circuit.

HOCOEN bit (HOCO Oscillation Enable)

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

Note: When the HOCOEN bit is set to 0, the system clock source is not switched to HOCO. The system clock source is only switched to HOCO by setting the Clock Source Select bits (SCKSCR.CKSEL[2:0]). To use the HOCO clock, set the OFS1.HOCOFRQ1 bits to an optimum value.

HOCOFRQ1[2:0] bits (HOCO Frequency Setting 1)

The HOCOFRQ1[2:0] bits select the HOCO frequency after a reset as 24, 32, 48, or 64 MHz.

ICSATS bit (Internal Clock Supply Architecture Type Select)

The ICSATS bit selects the internal clock supply architecture from Type A and Type B after a reset.

Internal Clock Supply Architecture Type A provides the clocks that the frequency of ICLK, PCLKB, PCLKD can be individually set in the System Clock Division Control Register (SCKDIVCR).

In Internal Clock Supply Architecture Type A, a fairly flexible operation frequency relationship between system and peripheral functions can be executed for various applications.

Internal Clock Supply Architecture Type B provides the clocks that the frequency of ICLK, PCLKB, PCLKD is fixed as ICLK = PCLKB = PCLKD regardless of the PCKB[2:0] and PCKD[2:0] settings in the System Clock Division Control Register (SCKDIVCR).

In Internal Clock Supply Architecture Type B, a simple operation frequency relationship between system and peripheral functions can be executed, and therefore, is a more advantageous type for power reduction.

For details of the System Clock Division Control Register (SCKDIVCR), see section 8.2.1. SCKDIVCR : System Clock Division Control Register.

For details of the clock generation circuit block diagram, see section 8.1. Overview.

Note: When Internal Clock Supply Architecture Type B is selected:

- 48 MHz or 64 MHz HOCO frequency setting in OFS1.HOCOFRQ1[2:0] is not allowed. Set the HOCO frequency to 32 MHz or 24 MHz.
- Memory wait setting in the MEMWAIT.MEMWAIT and FLDWAITR.FLDWAIT1 is not allowed. Use the default.

6.2.3 MPU Registers

Table 6.2 shows the registers related to the MPU function. For details, see section 14, Memory Protection Unit (MPU).

The security MPU is disabled on erasure of the flash memory. If incorrect data is written to an MPU register, the MCU might fail to operate. See section 14, Memory Protection Unit (MPU) to set the correct data.

Table 6.2 MPU registers (1 of 2)

Register name	Symbol	Function	Address*1	Size (byte)
Security MPU Program Counter Start Address Register 0	SECMPUPCS0	Specifies the security fetch region of code flash or SRAM.	0x0000_0408	4
Security MPU Program Counter End Address Register 0	SECMPUPCE0	Specifies the security fetch region of code flash or SRAM.	0x0000_040C	4
Security MPU Program Counter Start Address Register 1	SECMPUPCS1	Specifies the security fetch region of code flash or SRAM	0x0000_0410	4
Security MPU Program Counter End Address Register 1	SECMPUPCE1	Specifies the security fetch region of code flash or SRAM.	0x0000_0414	4
Security MPU Region 0 Start Address Register	SECMPUS0	Specifies the secure program and data of code flash	0x0000_0418	4
Security MPU Region 0 End Address Register	SECMPUE0	Specifies the secure program and data of code flash.	0x0000_041C	4
Security MPU Region 1 Start Address Register	SECMPUS1	Specifies the secure program and data of SRAM.	0x0000_0420	4
Security MPU Region 1 End Address Register	SECMPUE1	Specifies the secure program and data of SRAM.	0x0000_0424	4

Table 6.2 MPU registers (2 of 2)

Register name	Symbol	Function	Address*1	Size (byte)
Security MPU Region 2 Start Address Register	SECMPUS2	Specifies the secure data of security function.	0x0000_0428	4
Security MPU Region 2 End Address Register	SECMPUE2	Specifies the secure data of security function.	0x0000_042C	4
Security MPU Region 3 Start Address Register	SECMPUS3	Specifies the secure data of security function.	0x0000_0430	4
Security MPU Region 3 End Address Register	SECMPUE3	Specifies the secure data of security function.	0x0000_0434	4
Security MPU Access Control Register	SECMPUAC	Specifies the security enabled/disabled region.	0x0000_0438	4

Note 1. When the boot swap is set, the address of these registers change. Therefore, set (0x0000_2408 to 0x0000_243B) and (0x0000_0408 to 0x0000_043B) to the same value if boot swap is used.

6.2.4 AWS : Access Window Setting Register

Address: 0x0101_0010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BTFLG	—	—	—	—	FAWE[10:0]										
Value after reset:	User setting															

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

Bit	Symbol	Function	R/W
10:0	FAWS[10:0]	Access Window Start Block Address These bits specify the start block address for the access window. They do not represent the block number of the access window. The access window is only valid in the program flash area. The block address specifies the first address of the block and consists of the address bits [21:11].	R
14:11	—	When read, these bits return the written value.	R
15	FSPR	Protection of Access Window and Startup Area Select Function This bit controls the programming of the write/erase protection for the access window, the Startup Area Select Flag (BTFLG), and the temporary boot swap control. When this bit is set to 0, it cannot be changed to 1. 0: Executing the configuration setting command for programming the access window (FAWE[10:0], FAWS[10:0]) and the Startup Area Select Flag (BTFLG) is invalid 1: Executing the configuration setting command for programming the access window (FAWE[10:0], FAWS[10:0]) and the Startup Area Select Flag (BTFLG) is valid	R
26:16	FAWE[10:0]	Access Window End Block Address These bits specify the end block address for the access window. They do not represent the block number of the access window. The access window is only valid in the program flash area. The end block address for the access window is the next block to the acceptable programming and erasure region defined by the access window. The block address specifies the first address of the block and consists of the address bits [21:11].	R
30:27	—	When read, these bits return the written value.	R

Bit	Symbol	Function	R/W
31	BTFLG	Startup Area Select Flag This bit specifies whether the address of the startup area is exchanged for the boot swap function. 0: First 8-KB area (0x0000_0000 to 0x0000_1FFF) and second 8-KB area (0x0000_2000 to 0x0000_3FFF) are exchanged 1: First 8-KB area (0x0000_0000 to 0x0000_1FFF) and second 8-KB area (0x0000_2000 to 0x0000_3FFF) are not exchanged	R

Issuing the program or erase command to an area outside the access window causes a command-locked state. The access window is only valid in the program flash area. The access window provides protection in self-programming mode, serial programming mode, and on-chip debug mode. The access window can be locked by the FSPR bit.

The access window is specified in both the FAWS[10:0] bits and the FAWE[10:0] bits. The settings for the FAWS[10:0] and FAWE[10:0] bits are as follows:

FAWE[10:0] = FAWS[10:0]: The P/E command is allowed to execute in the full program flash area.

FAWE[10:0] > FAWS[10:0]: The P/E command is only allowed to execute in the window from the block pointed to by the FAWS[10:0] bits to the block one lower than the block pointed to by the FAWE[10:0] bits.

FAWE[10:0] < FAWS[10:0]: The P/E command is not allowed to execute in the program flash area.

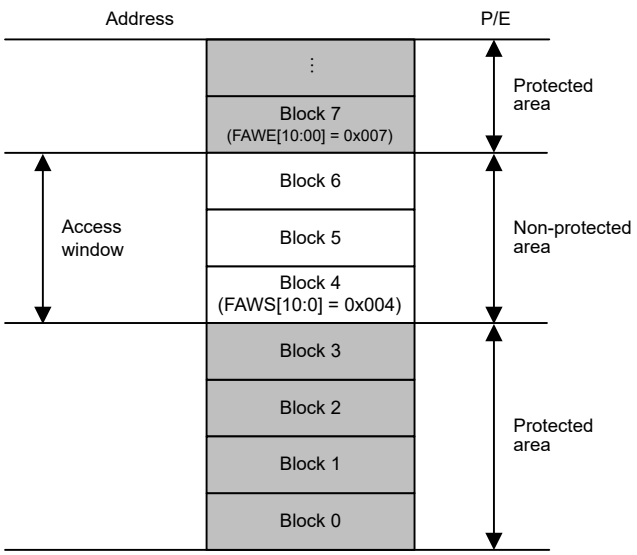


Figure 6.2 Access window overview

6.2.5 OSIS : OCD/Serial Programmer ID Setting Register

The OSIS register stores the ID for ID code protection of the OCD/serial programmer. When connecting the OCD/serial programmer, write values so that the MCU can determine whether to permit the connection. Use this register to check whether a code transmitted from the OCD/serial programmer matches the ID code in the option-setting memory. When the ID codes match, connection with the OCD/serial programmer is permitted, if not, connection with the OCD/serial programmer is not possible. The OSIS register must be set in 32-bit words.

Address: 0x0101_0018, 0x0101_0020, 0x0101_0028, 0x0101_0030

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

These fields hold the ID for use in ID authentication for the OCD/serial programmer.

ID code bits [127] and [126] determine whether the ID code protection is enabled, and the authentication method to use with the host. Table 6.3 shows how the ID code determines the authentication method.

Table 6.3 Specifications for ID code protection

Operating mode on boot up	ID code	State of protection	Operations on connection to programmer or on-chip debugger
Serial programming mode (SCI boot mode) On-chip debug mode (SWD boot mode)	0xFF, …, 0xFF (all bytes are 0xFF)	Protection disabled	The ID code is not checked, the ID code always matches, and the connection to the serial programmer or on-chip debugger*1 is permitted.
	Bit [127] = 1, bit [126] = 1, and at least one of the 16 bytes is not 0xFF	Protection enabled	Matching ID code indicates that authentication is complete and connection to the serial programmer or the on-chip debugger is permitted. Mismatching ID code indicates transition to the ID code protection wait state. When the ID code sent from the serial programmer or the on-chip debugger is ALeRASE in ASCII code (0x414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFF), the content of the user flash area is erased and all bits in the OSIS register are 1. However, when the AWS.FSPR bit is 0 or security MPU is enabled, the content of the user flash area is not erased.
	Bit [127] = 1 and bit [126] = 0	Protection enabled	Matching ID code indicates that authentication is complete and connection to the serial programmer or the on-chip debugger is permitted. Mismatching ID code indicates transition to the ID code protection wait state.
	Bit [127] = 0	Protection enabled	The ID code is not checked, the ID code is always mismatching, the connection to the serial programmer or the on-chip debugger is prohibited. When the ID code sent from the on-chip debugger is ALeRASE in ASCII code (0x414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFF), the content of the user flash area is erased and all bits in the OSIS register are 1. However, when the AWS.FSPR bit is 0 or security MPU is enabled, the content of the user flash area is not erased.

Note 1. Never send the ID code from on-chip debugger. Or send ID code 0xFF, …, 0xFF (all bytes 0xFF) from on-chip debugger.

6.3 Setting Option-Setting Memory

6.3.1 Allocation of Data in Option-Setting Memory

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

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

6.3.2 Setting Data for Programming Option-Setting Memory

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

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

Use the programming command to write data to the program flash area. Use the configuration setting command to write data to the option-setting memory in the configuration setting area. In addition, use the startup area select function to safely update the boot program that includes the option-setting memory.

For details of the programming command, the configuration setting command, and the startup area select function, see section 33, Flash Memory.

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

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

The MCU provides two setting procedures:

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

6.4 Usage Notes

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

When reserved areas and reserved bits in the option-setting memory are within the scope of programming, write 1 to all bits of reserved areas and all reserved bits. If 0 is written to these bits, normal operation cannot be guaranteed.

6.4.2 Note on FSPR Bit

The AWS.FSPR bit cannot be changed to 1 once it is set to 0. At that time, access window and startup area selection cannot be set again.

7. Low Voltage Detection (LVD)

7.1 Overview

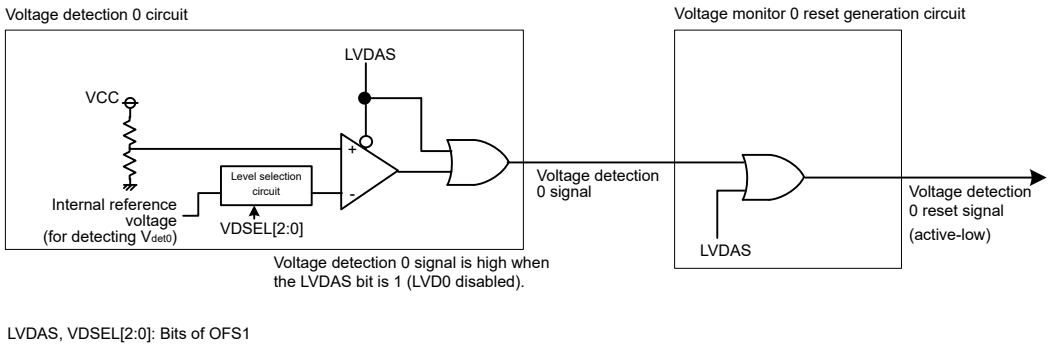
The Low Voltage Detection (LVD) module monitors the voltage level input to the VCC pin. The detection level can be selected by register settings. The LVD module consists of three separate voltage level detectors (LVD0, LVD1, LVD2). LVD0, LVD1, and LVD2 measure the voltage level input to the VCC pin. LVD registers allow your application to configure detection of VCC changes at various voltage thresholds.

Voltage monitor registers are used to configure the LVD to trigger an interrupt, event link output, or reset when the thresholds are crossed.

Table 7.1 lists the LVD specifications. Figure 7.1 shows a block diagram of the voltage monitor 0 reset generation circuit. Figure 7.2 shows a block diagram of the voltage monitor 1 interrupt and reset circuit, and Figure 7.3 shows a block diagram of the voltage monitor 2 interrupt and reset circuit.

Table 7.1 LVD specifications

Parameter		Voltage monitor 0	Voltage monitor 1	Voltage monitor 2
Means for setting up operation		OFS1 register	Registers	Registers
Target for monitoring		VCC pin input voltage	VCC pin input voltage	VCC pin input voltage
Monitored voltage		V_{det0}	V_{det1}	V_{det2}
Detected event		Voltage falls past V_{det0}	Voltage rises or falls past V_{det1}	Voltage rises or falls past V_{det2}
Detection voltage		Selectable from 5 different levels in the OFS1.VDSEL0[2:0] bits	Selectable from 16 different levels in the LVDLVLR.LVD1LVL[4:0] bits	Selectable from 4 different levels in the LVDLVLR.LVD2LVL[2:0] bits
Monitoring flag		None	LVD1SR.MON flag: Monitors whether voltage is higher or lower than V_{det1}	LVD2SR.MON flag: Monitors whether voltage is higher or lower than V_{det2}
			LVD1SR.DET flag: V_{det1} passage detection	LVD2SR.DET flag: V_{det2} passage detection
Process on voltage detection	Reset	Voltage monitor 0 reset	Voltage monitor 1 reset	Voltage monitor 2 reset
		Reset when V_{det0} > VCC CPU restart after specified time with VCC > V_{det0}	Reset when V_{det1} > VCC CPU restart timing selectable: after specified time with VCC > V_{det1} or V_{det1} > VCC	Reset when V_{det2} > VCC CPU restart timing selectable: after specified time with either VCC > V_{det2} or V_{det2} > VCC
	Interrupt	No interrupt	Voltage monitor 1 interrupt	Voltage monitor 2 interrupt
			Non-maskable or maskable interrupt selectable	Non-maskable or maskable interrupt selectable
			Interrupt request issued when V_{det1} > VCC and VCC > V_{det1} or either	Interrupt request issued when V_{det2} > VCC and VCC > V_{det2} or either
Event link function		None	Available Output of event signals on detection of V_{det1} crossings	Available Output of event signals on detection of V_{det2} crossings



Note: For details of OFS1.LVDAS and OFS1.VDSEL0[2:0], see section 6, Option-Setting Memory.

Figure 7.1 Block diagram of voltage monitor 0 reset generation circuit

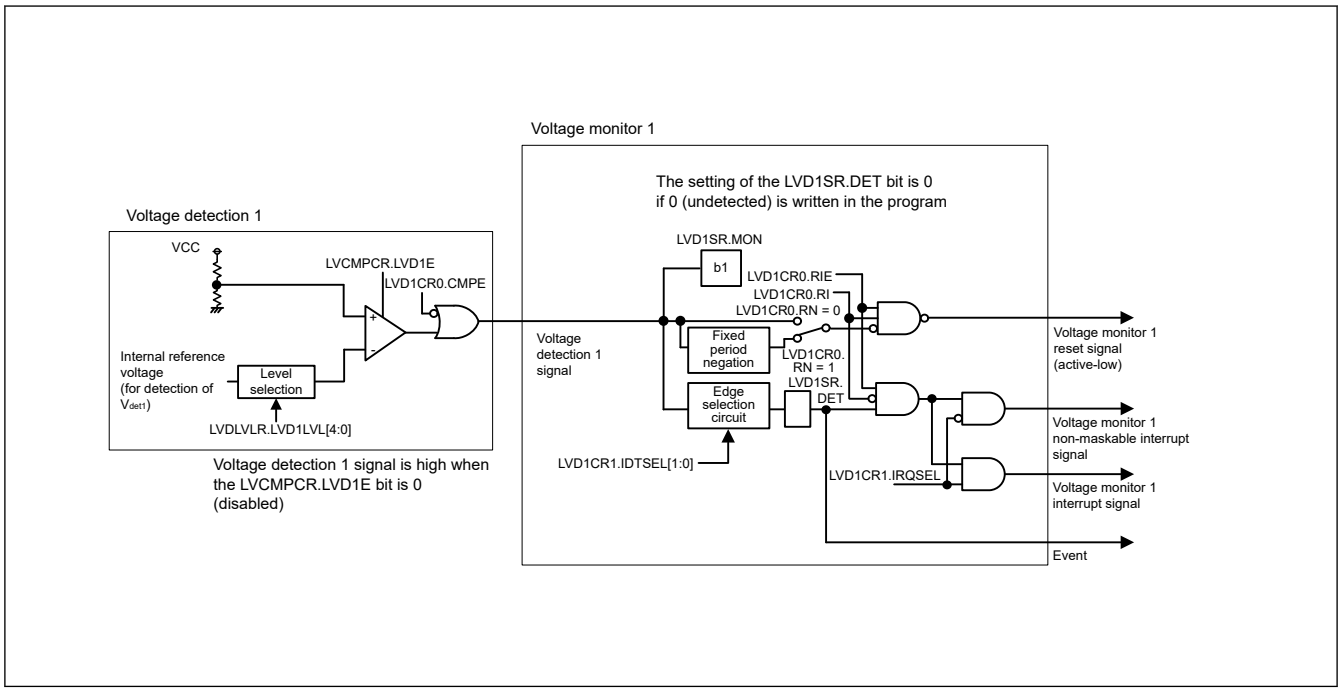


Figure 7.2 Block diagram of voltage monitor 1 interrupt and reset circuit

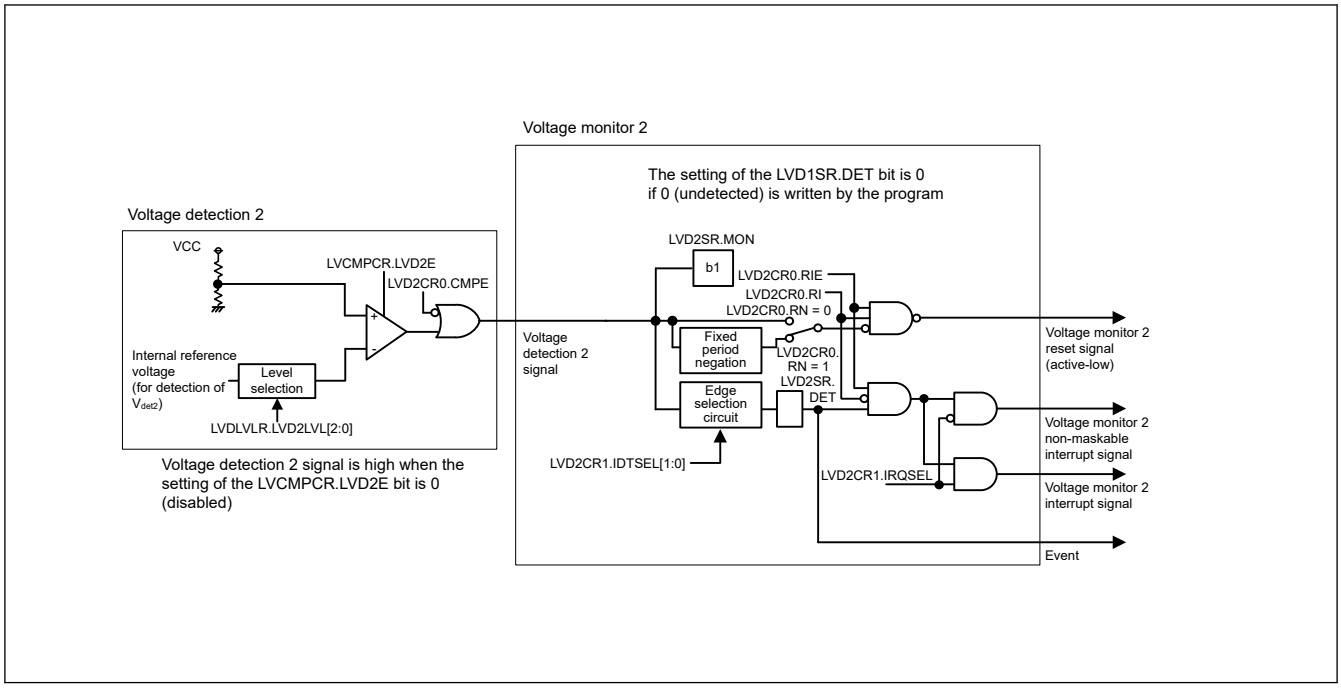


Figure 7.3 Block diagram of voltage monitor 2 interrupt and reset circuit

7.2 Register Descriptions

7.2.1 LVCMPCR : Voltage Monitor Circuit Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x417

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	LVD2E	LVD1E	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 0. The write value should be 0.	R/W
5	LVD1E	Voltage Detection 1 Enable 0: Voltage detection 1 circuit disabled 1: Voltage detection 1 circuit enabled	R/W
6	LVD2E	Voltage Detection 2 Enable 0: Voltage detection 2 circuit disabled 1: Voltage detection 2 circuit enabled	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

LVD1E bit (Voltage Detection 1 Enable)

When using voltage detection 1 interrupt/reset or the LVD1SR.MON flag, set the LVD1E bit to 1. The voltage detection 1 circuit starts when LVD1 operation stabilization time ($t_{d(E-A)}$) elapses after the LVD1E bit value is changed from 0 to 1. For details on $t_{d(E-A)}$, see section 35, Electrical Characteristics.

LVD2E bit (Voltage Detection 2 Enable)

When using voltage detection 2 interrupt/reset or the LVD2SR.MON flag, set the LVD2E bit to 1. The voltage detection 2 circuit starts when LVD2 operation stabilization time $t_{d(E-A)}$ elapses after the LVD2E bit value is changed from 0 to 1. For details on LVD2 operation stabilization time $t_{d(E-A)}$, see section 35, Electrical Characteristics.

7.2.2 LVDLVLR : Voltage Detection Level Select Register

Base address: SYSC = 0x4001_E000

Offset address: 0x418

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LVD2LVL[2:0]			LVD1LVL[4:0]				
Value after reset:	0	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
4:0	LVD1LVL[4:0]	Voltage Detection 1 Level Select (Standard voltage during fall in voltage)*1 0x00: V_{det1_0} 0x01: V_{det1_1} 0x02: V_{det1_2} 0x03: V_{det1_3} 0x04: V_{det1_4} 0x05: V_{det1_5} 0x06: V_{det1_6} 0x07: V_{det1_7} 0x08: V_{det1_8} 0x09: V_{det1_9} 0x0A: V_{det1_A} 0x0B: V_{det1_B} 0x0C: V_{det1_C} 0x0D: V_{det1_D} 0x0E: V_{det1_E} 0x0F: V_{det1_F} Others: Setting prohibited	R/W
7:5	LVD2LVL[2:0]	Voltage Detection 2 Level Select (Standard voltage during fall in voltage)*1 0 0 0: V_{det2_0} 0 0 1: V_{det2_1} 0 1 0: V_{det2_2} 0 1 1: V_{det2_3} Others: Setting prohibited	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

Note 1. See section 35, Electrical Characteristics for the voltage levels to be detected. Keep the initial value if LVD1 is not used. When using LVD0, set the detection voltage of LVD1 higher than the detection voltage of LVD0. The LVD1LVL [4:0] bits can be rewritten only once after reset.

The contents of the LVDLVLR register can only be changed if the LVCMPCR.LVD1E and LVCMPCR.LVD2E bits (voltage detection n circuit disable, n = 1, 2) are both 0. Do not set LVD detectors 1 and 2 to the same voltage detection level.

7.2.3 LVD1CR0 : Voltage Monitor 1 Circuit Control Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x41A

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

Bit	Symbol	Function	R/W
0	RIE	Voltage Monitor 1 Interrupt/Reset Enable 0: Disable 1: Enable	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
2	CMPE	Voltage Monitor 1 Circuit Comparison Result Output Enable 0: Disable voltage monitor 1 circuit comparison result output 1: Enable voltage monitor 1 circuit comparison result output	R/W
3	—	The read value is undefined. The write value should be 1.	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	RI	Voltage Monitor 1 Circuit Mode Select 0: Generate voltage monitor 1 interrupt on V_{det1} crossing 1: Enable voltage monitor 1 reset when the voltage falls to and below V_{det1}	R/W
7	RN	Voltage Monitor 1 Reset Negate Select 0: Negate after a stabilization time (t_{LVD1}) when VCC > V_{det1} is detected 1: Negate after a stabilization time (t_{LVD1}) on assertion of the LVD1 reset	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

RIE bit (Voltage Monitor 1 Interrupt/Reset Enable)

The RIE bit enables or disables voltage monitor 1 interrupt/reset. Ensure that neither a voltage monitor 1 interrupt nor a voltage monitor 1 reset is generated during programming or erasure of the flash memory.

CMPE bit (Voltage Monitor 1 Circuit Comparison Result Output Enable)

The CMPE bit enables or disables voltage monitor 1 circuit comparison result output. Set the CMPE bit to 1 after the voltage detection 1 circuit enables and stabilization time ($t_{d(E-A)}$) elapses. When stopping the voltage detection 1 circuit, disable the voltage detection 1 circuit after setting the CMPE bit is 0.

RN bit (Voltage Monitor 1 Reset Negate Select)

If the RN bit is set to 1 (negation follows a stabilization time on assertion of the LVD1 reset signal), set the MOCOCR.MCSTP bit to 0 (the MOCO operates). In addition, for a transition to Software Standby mode, the only possible value for the RN bit is 0 (negation follows stabilization time when VCC > V_{det1} is detected). Do not set the RN bit to 1 when this is the case.

7.2.4 LVD2CR0 : Voltage Monitor 2 Circuit Control Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x41B

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

Bit	Symbol	Function	R/W
0	RIE	Voltage Monitor 2 Interrupt/Reset Enable 0: Disable 1: Enable	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	CMPE	Voltage Monitor 2 Circuit Comparison Result Output Enable 0: Disable voltage monitor 2 circuit comparison result output 1: Enable voltage monitor 2 circuit comparison result output	R/W
3	—	The read value is undefined. The write value should be 1.	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	RI	Voltage Monitor 2 Circuit Mode Select 0: Generate voltage monitor 2 interrupt on V_{det2} crossing 1: Enable voltage monitor 2 reset when the voltage falls to and below V_{det2}	R/W

Bit	Symbol	Function	R/W
7	RN	Voltage Monitor 2 Reset Negate Select 0: Negate after a stabilization time (t_{LVD2}) when VCC > V_{det2} is detected 1: Negate after a stabilization time (t_{LVD2}) on assertion of the LVD2 reset	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

RIE bit (Voltage Monitor 2 Interrupt/Reset Enable)

The RIE bit enables or disables the voltage monitor 2 interrupt/reset. Ensure that neither a voltage monitor 2 interrupt nor a voltage monitor 2 reset is generated during programming or erasure of the flash memory.

CMPE bit (Voltage Monitor 2 Circuit Comparison Result Output Enable)

The CMPE bit enables or disables voltage monitor 2 circuit comparison result output. Set the CMPE bit to 1 after the voltage detection 2 circuit enables and stabilization time ($t_{d(E-A)}$) elapses. When stopping the voltage detection 2 circuit, disable the voltage detection 2 circuit after setting the CMPE bit is 0.

RN bit (Voltage Monitor 2 Reset Negate Select)

If the RN bit is set to 1 (negating LVD2 reset in a specified time after its assertion), set the MOCOCR.MCSTP bit to 0 (the MOCO operates). Additionally, for a transition to Software Standby mode, the only possible value for the RN bit is 0 (negation follows a stabilization time when VCC > V_{det2} is detected). Do not set the RN bit to 1 (negation follows a stabilization time after assertion of the LVD2 reset signal) when this is the case.

7.2.5 LVD1CR1 : Voltage Monitor 1 Circuit Control Register 1

Base address: SYSC = 0x4001_E000

Offset address: 0x0E0

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

Bit	Symbol	Function	R/W
1:0	IDTSEL[1:0]	Voltage Monitor 1 Interrupt Generation Condition Select 0 0: When VCC ≥ V_{det1} (rise) is detected 0 1: When VCC < V_{det1} (fall) is detected 1 0: When fall and rise are detected 1 1: Settings prohibited	R/W
2	IRQSEL	Voltage Monitor 1 Interrupt Type Select 0: Non-maskable interrupt 1: Maskable interrupt*1	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (writing enabled) before rewriting this register.
Note 1. When enabling maskable interrupts, do not change the NMIER.LVD1EN bit value in the ICU from the reset state.

7.2.6 LVD1SR : Voltage Monitor 1 Circuit Status Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0E1

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

Bit	Symbol	Function	R/W
0	DET	Voltage Monitor 1 Voltage Variation Detection Flag 0: Not detected 1: V_{det1} crossing is detected	R/W*1
1	MON	Voltage Monitor 1 Signal Monitor Flag 0: VCC < V_{det1} 1: VCC ≥ V_{det1} or MON is disabled	R
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, 2 system clock cycles are required for the bit to be read as 0.

DET flag (Voltage Monitor 1 Voltage Variation Detection Flag)

The DET flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.CMPE bit is 1 (voltage monitor 1 circuit comparison result output enabled).

When detecting V_{det1}, set the DET flag to 0 after setting LVD1CR0.RIE is 0 (disabled). When setting LVD1CR0.RIE bit to 1 (enabled) after setting it to 0, wait for 2 or more PCLKB cycles which have elapsed.

MON flag (Voltage Monitor 1 Signal Monitor Flag)

The MON flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.CMPE bit is 1 (voltage monitor 1 circuit comparison result output enabled).

7.2.7 LVD2CR1 : Voltage Monitor 2 Circuit Control Register 1

Base address: SYSC = 0x4001_E000

Offset address: 0x0E2

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

Bit	Symbol	Function	R/W
1:0	IDTSEL[1:0]	Voltage Monitor 2 Interrupt Generation Condition Select 0 0: When VCC ≥ V_{det2} (rise) is detected 0 1: When VCC < V_{det2} (fall) is detected 1 0: When fall and rise are detected 1 1: Settings prohibited	R/W
2	IRQSEL	Voltage Monitor 2 Interrupt Type Select 0: Non-maskable interrupt 1: Maskable interrupt*1	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (writing enabled) before rewriting this register.
Note 1. When enabling maskable interrupts, do not change the NMIER.LVD2EN bit value in the ICU from the reset state.

7.2.8 LVD2SR : Voltage Monitor 2 Circuit Status Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0E3

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

Bit	Symbol	Function	R/W
0	DET	Voltage Monitor 2 Voltage Variation Detection Flag 0: Not detected 1: V_{det2} crossing is detected	R/W*1
1	MON	Voltage Monitor 2 Signal Monitor Flag 0: VCC < V_{det2} 1: VCC ≥ V_{det2} or MON is disabled	R
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, 2 system clock cycles are required for the bit to be read as 0.

DET flag (Voltage Monitor 2 Voltage Variation Detection Flag)

The DET flag is enabled when the LVCMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.CMPE bit is 1 (voltage monitor 2 circuit comparison result output enabled).

When detecting V_{det2}, set the DET flag to 0 after setting LVD2CR0.RIE is 0 (disabled). When setting LVD2CR0.RIE bit to 1 (enabled) after setting it to 0, wait for 2 or more PCLKB cycles which have elapsed.

MON flag (Voltage Monitor 2 Signal Monitor Flag)

The MON flag is enabled when the LVCMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.CMPE bit is 1 (voltage monitor 2 circuit comparison result output enabled).

7.3 VCC Input Voltage Monitor

7.3.1 Monitoring V_{det0}

The comparison results from voltage monitor 0 are not available for reading.

7.3.2 Monitoring V_{det1}

Table 7.2 shows the procedures to set up monitoring against V_{det1}. After the settings are complete, the comparison results from voltage monitor 1 can be monitored with the LVD1SR.MON flag.

Table 7.2 Procedures to set up monitoring against V_{det1}

Step		Monitoring the comparison results from voltage monitor 1
Setting up the voltage detection 1 circuit	1	Set LVCMPCR.LVD1E = 0 to disable voltage detection 1 before writing to the LVDLVLR.LVD1LVL[4:0] bits.
	2	Select the detection voltage in the LVDLVLR.LVD1LVL[4:0] bits.
	3	Set LVCMPCR.LVD1E = 1 to enable the voltage detection 1 circuit.
	4	Wait for at least $t_{d\ (E-A)}$ for the LVD1 operation stabilization time after LVD1 is enabled.
Enabling output	5	Set LVD1CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 1.

7.3.3 Monitoring V_{det2}

Table 7.3 shows the procedures to set up monitoring against V_{det2}. After the settings are complete, the comparison results from voltage monitor 2 can be monitored in the LVD2SR.MON flag.

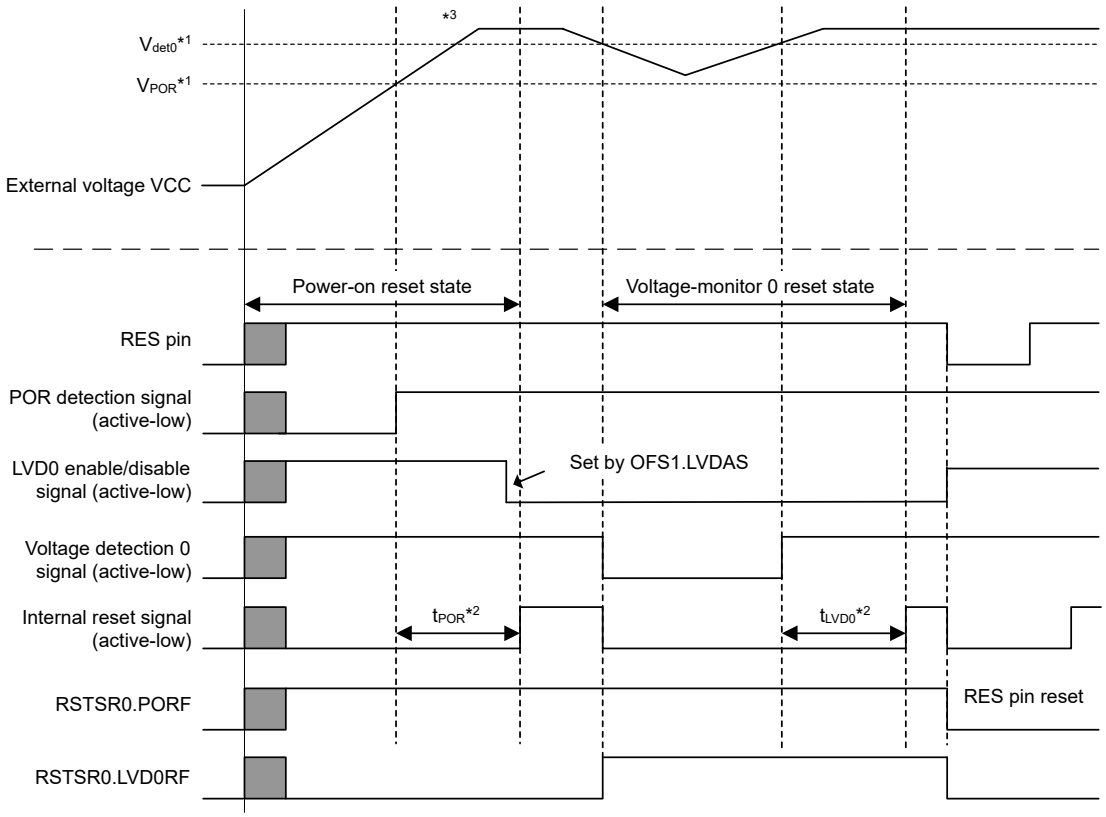
Table 7.3 Procedures to set up monitoring against V_{det2}

Step		Monitoring the results of comparison by voltage monitor 2
Setting up the voltage detection 2 circuit	1	Set LVCMPCR.LVD2E = 0 to disable voltage detection 2 before writing to the LVDLVLR.LVD2LVL[2:0] bits.
	2	Select the detection voltage in the LVDLVLR.LVD2LVL[2:0] bits.
	3	Set LVCMPCR.LVD2E = 1 to enable the voltage detection 2 circuit.
	4	Wait for at least $t_{d\ (E-A)}$ for the LVD2 operation stabilization time after LVD2 is enabled.
Enabling output	5	Set LVD2CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 2.

7.4 Reset from Voltage Monitor 0

When using the reset from voltage monitor 0, clear the OFS1.LVDAS bit to 0 to enable the voltage monitor 0 reset after a reset. However, at boot mode, the reset from voltage monitor 0 is disabled regardless of the value of the OFS1.LVDAS bit.

Figure 7.4 shows an example of operations for a voltage monitor 0 reset.



Note: For details of the electrical characteristics, see section 35, Electrical Characteristics.
Note 1. V_{POR} indicates the detection level for a power-on reset and V_{det0} indicates the detection level for a voltage monitor 0 reset.
Note 2. t_{POR}: the period until the power-on reset is released.
t_{LVD0}: the period until the LVD0 reset is released.
Note 3. At power-on, the POR reset is released after VCC has reached the minimum guaranteed voltage.

Figure 7.4 Example of voltage monitor 0 reset operation

7.5 Interrupt and Reset from Voltage Monitor 1

An interrupt or reset can be generated in response to the comparison results from the voltage monitor 1 circuit.

Table 7.4 shows the procedures for setting bits related to the voltage monitor 1 interrupt/reset so that voltage monitoring occurs. Table 7.5 shows the procedures for setting bits related to the voltage monitor 1 interrupt/reset so that voltage monitoring stops. Figure 7.5 shows an example of operations for a voltage monitor 1 interrupt. For the operation of the voltage monitor 1 reset, see Figure 5.2 in section 5, Resets.

When using the voltage monitor 1 circuit in Software Standby mode, set up the circuit using the procedures in this section.

(1) Setting in Software Standby mode

- When VCC > V_{det1} is detected, negate the voltage monitor 1 reset signal (LVD1CR0.RN = 0) following a stabilization time.

Table 7.4 Procedures for setting bits related to voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitoring occurs

Step		Voltage monitor 1 interrupt (voltage monitor 1 ELC event output)	Voltage monitor 1 reset
Setting up the voltage detection 1 circuit	1	Set LVCMPCR.LVD1E = 0 to disable voltage detection 1 before writing to the LVDLVLR register.	
	2	Select the detection voltage in the LVDLVLR.LVD1LVL[4:0] bits.	
	3	Set LVCMPCR.LVD1E = 1 to enable the voltage detection 1 circuit.	
	4	Wait for at least $t_{d\ (E-A)}$ for the LVD1 operation stabilization time after LVD1 is enabled.*1	
Setting up the voltage monitor 1 interrupt or reset	5	Set LVD1CR0.RI = 0 to select the voltage monitor 1 interrupt.	• Set LVD1CR0.RI = 1 to select the voltage monitor 1 reset. • Select the type of reset negation in the LVD1CR0.RN bit.
	6	• Select the interrupt request condition in the LVD1CR1.IDTSEL[1:0] bits. • Select the interrupt type in the LVD1CR1.IRQSEL bit.	—
Enabling output	7	Set LVD1SR.DET = 0.	
	8	Set LVD1CR0.RIE = 1 to enable the voltage monitor 1 interrupt or reset.*2	
	9	Set LVD1CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 1.	

Note 1. Steps 5 to 8 can be performed during the wait time in step 4. For details on $t_{d\ (E-A)}$, see section 35, Electrical Characteristics.
Note 2. Step 8 is not required if only the ELC event signal is to be output.

Table 7.5 Procedures for setting bits related to voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitoring stops

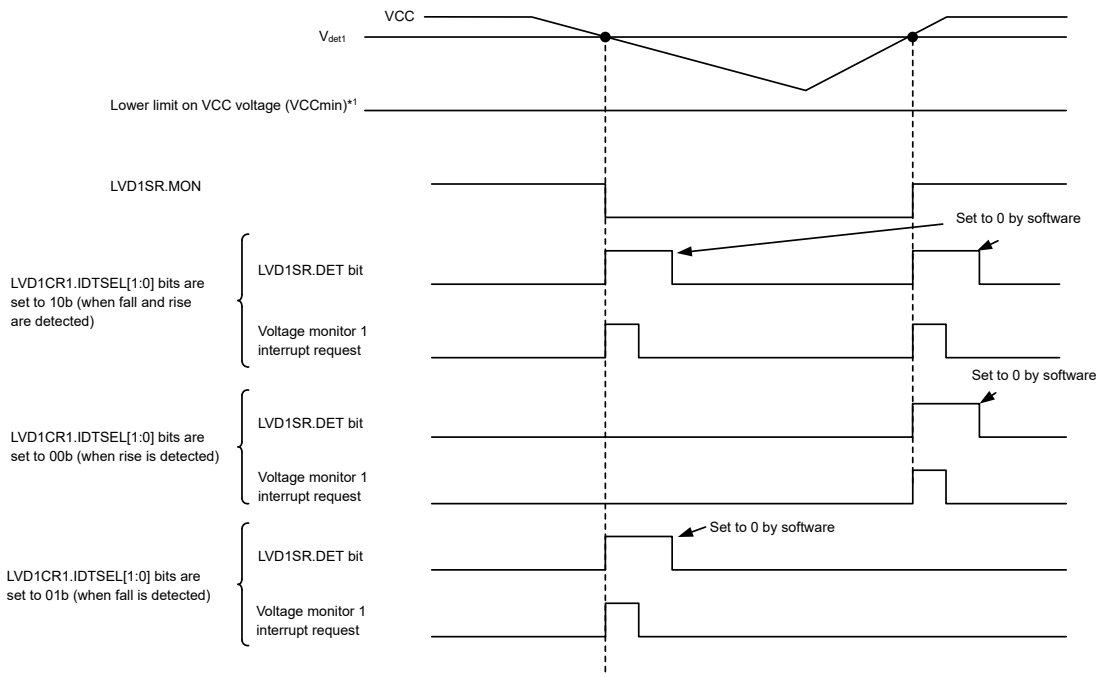
Step		Voltage monitor 1 interrupt (voltage monitor 1 ELC event output), voltage monitor 1 reset
Stopping the enabling output	1	Set LVD1CR0.CMPE = 0 to disable output of the comparison results from voltage monitor 1.
	2	Set LVD1CR0.RIE = 0 to disable the voltage monitor 1 interrupt or reset.*1
Stopping the voltage detection 1 circuit	3	Set LVCMPCR.LVD1E = 0 to disable the voltage detection 1 circuit.

Note 1. Step 2 is not required if only the ELC event signal is to be output.

If the voltage monitor 1 interrupt or reset setting is to be made again after it is used and stopped once, you can omit the following steps in the procedures for stopping and setting, depending on the conditions:

- Setting the voltage detection 1 circuit is not required if the settings for the circuit do not change.
- Setting the voltage monitor 1 interrupt or reset is not required if the settings for the voltage monitor 1 interrupt or voltage monitor 1 reset do not change.

Figure 7.5 shows an example of the voltage monitor 1 interrupt operation.



Note 1. When the voltage monitor 0 reset is not in use, use at VCC ≥ VCCmin.

Figure 7.5 Example of voltage monitor 1 interrupt operation

7.6 Interrupt and Reset from Voltage Monitor 2

An interrupt or reset can be generated in response to the comparison results from the voltage monitor 2 circuit.

Table 7.6 shows the procedures for setting bits related to the voltage monitor 2 interrupt/reset so that voltage monitoring occurs. Table 7.7 shows the procedures for setting bits related to the voltage monitor 2 interrupt/reset so that voltage monitoring stops. Figure 7.6 shows an example of operations for a voltage monitor 2 interrupt. For the operation of the voltage monitor 2 reset, see Figure 5.2 in section 5, Resets.

When using the voltage monitor 2 circuit in Software Standby mode, set up the circuit with the following procedures.

(1) Setting in Software Standby mode

- When VCC > V_{det2} is detected, negate the voltage monitor 2 reset signal (LVD2CR0.RN = 0) following a LVD2 stabilization time.

Table 7.6 Procedures for setting bits related to voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitoring occurs (1 of 2)

Step		Voltage monitor 2 interrupt (voltage monitor 2 ELC event output)	Voltage monitor 2 reset
Setting up the voltage detection 2 circuit	1	Set LVCMPCR.LVD2E = 0 to disable voltage detection 2 before writing to the LVDLVLR register.	
	2	Select the detection voltage in the LVDLVLR.LVD2LVL[2:0] bits.	
	3	Set LVCMPCR.LVD2E = 1 to enable the voltage detection 2 circuit.	
	4	Wait for at least $t_{d\ (E-A)}$ for the LVD2 operation stabilization time after LVD2 is enabled.*1	

Table 7.6 Procedures for setting bits related to voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitoring occurs (2 of 2)

Step		Voltage monitor 2 interrupt (voltage monitor 2 ELC event output)	Voltage monitor 2 reset
Setting up the voltage monitor 2 interrupt or reset	5	Set LVD2CR0.RI = 0 to select the voltage monitor 2 interrupt.	• Set LVD2CR0.RI = 1 to select the voltage monitor 2 reset. • Select the type of reset negation in the LVD2CR0.RN bit.
	6	• Select the interrupt request condition in the LVD2CR1.IDTSEL[1:0] bits. • Select the interrupt type in the LVD2CR1.IRQSEL bit.	—
Enabling output	7	Set LVD2SR.DET = 0.	
	8	Set LVD2CR0.RIE = 1 to enable the voltage monitor 2 interrupt or reset.*2	
	9	Set LVD2CR0.CMPE = 1 to enable output of the comparison results from voltage monitor 2.	

Note 1. Steps 5 to 8 can be performed during the wait time in step 4. For details on $t_{d(E-A)}$, see section 35, Electrical Characteristics.
Note 2. Step 8 is not required if only the ELC event signal is to be output.

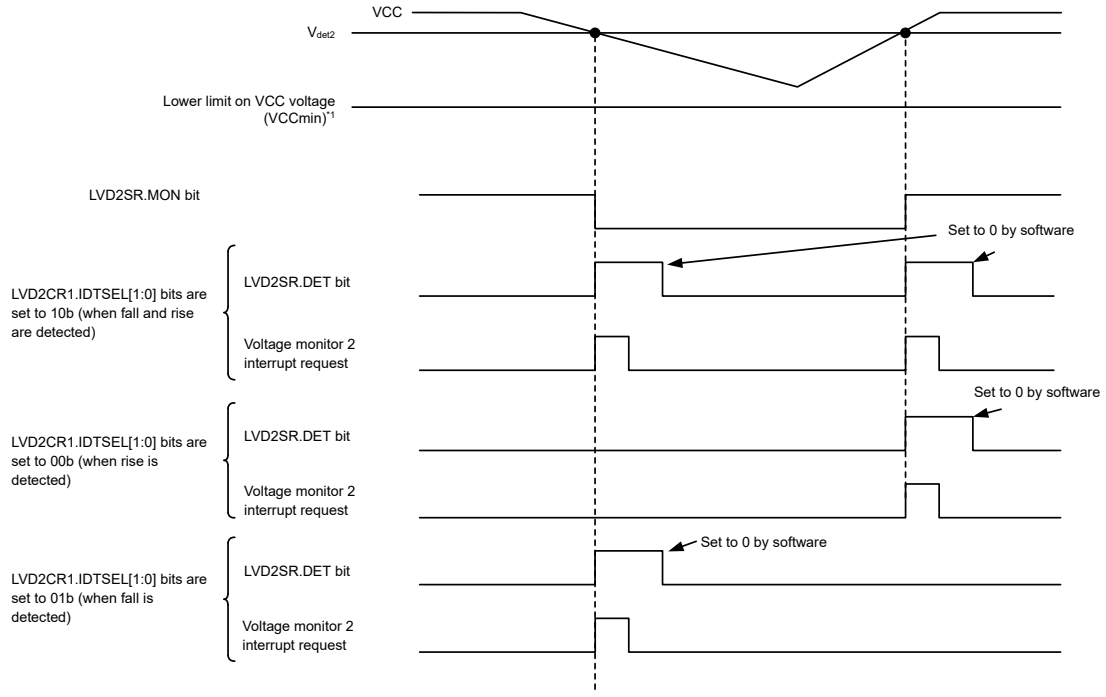
Table 7.7 Procedures for setting bits related to voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitoring stops

Step		Voltage monitor 2 interrupt (voltage monitor 2 ELC event output), voltage monitor 2 reset
Settings to stop enabling output	1	Set LVD2CR0.CMPE = 0 to disable output of the comparison results from voltage monitor 2.
	2	Set LVD2CR0.RIE = 0 to disable the voltage monitor 2 interrupt or reset.*1
Stopping the voltage detection 2 circuit	3	Set LVCMPCR.LVD2E = 0 to disable the voltage detection 2 circuit.

Note 1. Step 2 is not required if only the ELC event signal is to be output.

If the voltage monitor 2 interrupt or reset setting is to be made again after it is used and stopped once, you can omit the following steps in the procedures for stopping and setting, depending on the conditions:

- Setting the voltage detection 2 is not required if the settings for the circuit do not change.
- Setting the voltage monitor 2 interrupt or reset is not required if the settings for the voltage monitor 2 interrupt or voltage monitor 2 reset do not change.



Note 1. When the voltage monitor 0 reset is not in use, use at VCC ≥ VCCmin.

Figure 7.6 Example of voltage monitor 2 interrupt operation

7.7 Event Link Controller (ELC) Output

The LVD can output the event signals to the Event Link Controller (ELC).

(1) V_{det1} Crossing Detection Event

The LVD outputs the event signal when it detects that the voltage has passed the V_{det1} voltage while both the voltage detection 1 circuit and the voltage monitor 1 circuit comparison result output are enabled.

(2) V_{det2} Crossing Detection Event

The LVD outputs the event signal when it detects that the voltage has passed the V_{det2} voltage while both the voltage detection 2 circuit and the voltage monitor 2 circuit comparison result output are enabled.

When enabling the event link output function of the LVD, you must enable the LVD before enabling the LVD event link function of the ELC. To stop the event link output function of the LVD, you must stop the LVD before disabling the LVD event link function of the ELC.

7.7.1 Interrupt Handling and Event Linking

The LVD provides bits to separately enable or disable the voltage monitor 1 and 2 interrupts. When an interrupt source is generated and the interrupt is enabled by the interrupt enable bit, the interrupt signal is output to the CPU.

In contrast, as soon as an interrupt source is generated, an event link signal is output as the event signal to the other module through the ELC, regardless of the state of the interrupt enable bit.

It is possible to output voltage monitor 1 and 2 interrupts in Software Standby mode.

- When a V_{det1} or V_{det2} passage events is detected in Software Standby mode, event signals are not generated for the ELC because the clock is not supplied in Software Standby mode. Because the V_{det1} and V_{det2} passage detection flags

are saved, when the clock supply resumes after returning from Software Standby mode, the event signals for the ELC are output based on the state of the V_{det1} and V_{det2} detection flags.

8. Clock Generation Circuit

8.1 Overview

The MCU provides a clock generation circuit. Table 8.1 and Table 8.2 list the clock generation circuit specifications. Figure 8.1 and Figure 8.2 show a block diagram, and Table 8.3 lists the I/O pins.

Table 8.1 Clock generation circuit specifications for the clock sources

Clock source	Description	Specification
Main clock oscillator (MOSC)	Resonator frequency	1 MHz to 20 MHz
	External clock input frequency	Up to 20 MHz
	External resonator or additional circuit	ceramic resonator, crystal
	Connection pins	EXTAL, XTAL
	Drive capability switching	Available
	Oscillation stop detection function	Available
High-speed on-chip oscillator (HOCO)	Oscillation frequency	24/32/48/64 MHz
	User trimming	Available
Middle-speed on-chip oscillator (MOCO)	Oscillation frequency	8 MHz
	User trimming	Available
Low-speed on-chip oscillator (LOCO)	Oscillation frequency	32.768 kHz
	User trimming	Available
IWDT-dedicated on-chip oscillator (IWDTLOCO)	Oscillation frequency	15 kHz
	User trimming	No
External clock input for SWD (SWCLK)	Input clock frequency	Up to 12.5 MHz

Table 8.2 Clock generation circuit specifications for the internal clocks (1 of 2)

Item	Clock source	Clock supply	Specification
System clock (ICLK)	MOSC*1/HOCO/MOCO/ LOCO	CPU, DTC, Flash, Flash-IF, SRAM	Up to 64 MHz Division ratios: 1/2/4/8/16/32/64 1 MHz to 64 MHz (P/E)
Peripheral module clock B (PCLKB)	MOSC*1/HOCO/MOCO/ LOCO	Peripheral modules (CAC, ELC, I/O Ports, KINT, POEG, GPT, AGTW, ACMPHS, WDT, IWDT, SCI, IIC, SPI, CRC, ADC12, DOC)	Up to 32 MHz Division ratios:1/2/4/8/16/32/64
Peripheral module clock D (PCLKD)	MOSC*1/HOCO/MOCO/ LOCO	Peripheral modules (GPT count clock, ADC12 conversion clock)	Up to 64 MHz Division ratios: 1/2/4/8/16/32/64
AGTW clock (AGTLCLK)	LOCO	AGTW	32.768 kHz
CAC Main clock (CACMCLK)	MOSC*1	CAC	Up to 20 MHz
CAC LOCO clock (CACLCLK)	LOCO	CAC	32.768 kHz
CAC MOCO clock (CACMOCLK)	MOCO	CAC	8 MHz
CAC HOCO clock (CACHCLK)	HOCO	CAC	24/32/48/64 MHz
CAC IWDTLOCO clock (CACILCLK)	IWDTLOCO	CAC	15 kHz
IWDT clock (IWDTCLK)	IWDTLOCO	IWDT	15 kHz
SysTick timer clock (SYSTICCLK)	LOCO	SysTick timer	32.768 kHz

Table 8.2 Clock generation circuit specifications for the internal clocks (2 of 2)

Item	Clock source	Clock supply	Specification
Clock/buzzer output (CLKOUT)	MOSC*1/LOCO/MOCO/ HOCO	CLKOUT pin	Up to 16 MHz Division ratios: 1/2/4/8/16/32/64/128
Serial wire clock (SWCLK)	SWCLK pin	OCD	Up to 12.5 MHz

Note: Restrictions on setting clock frequency: ICLK ≥ PCLKB, PCLKD ≥ PCLKB
PCLKB Restrictions on clock frequency ratio: (N: integer, and up to 64)
ICLK:PCLKB = N:1, ICLK:PCLKD = N:1 or 1:N
Minimum ICLK frequency is 1 MHz in Programming/Erasure (P/E) mode.
Note 1. Using MOSC as a clock source in the Low-speed mode is prohibited.

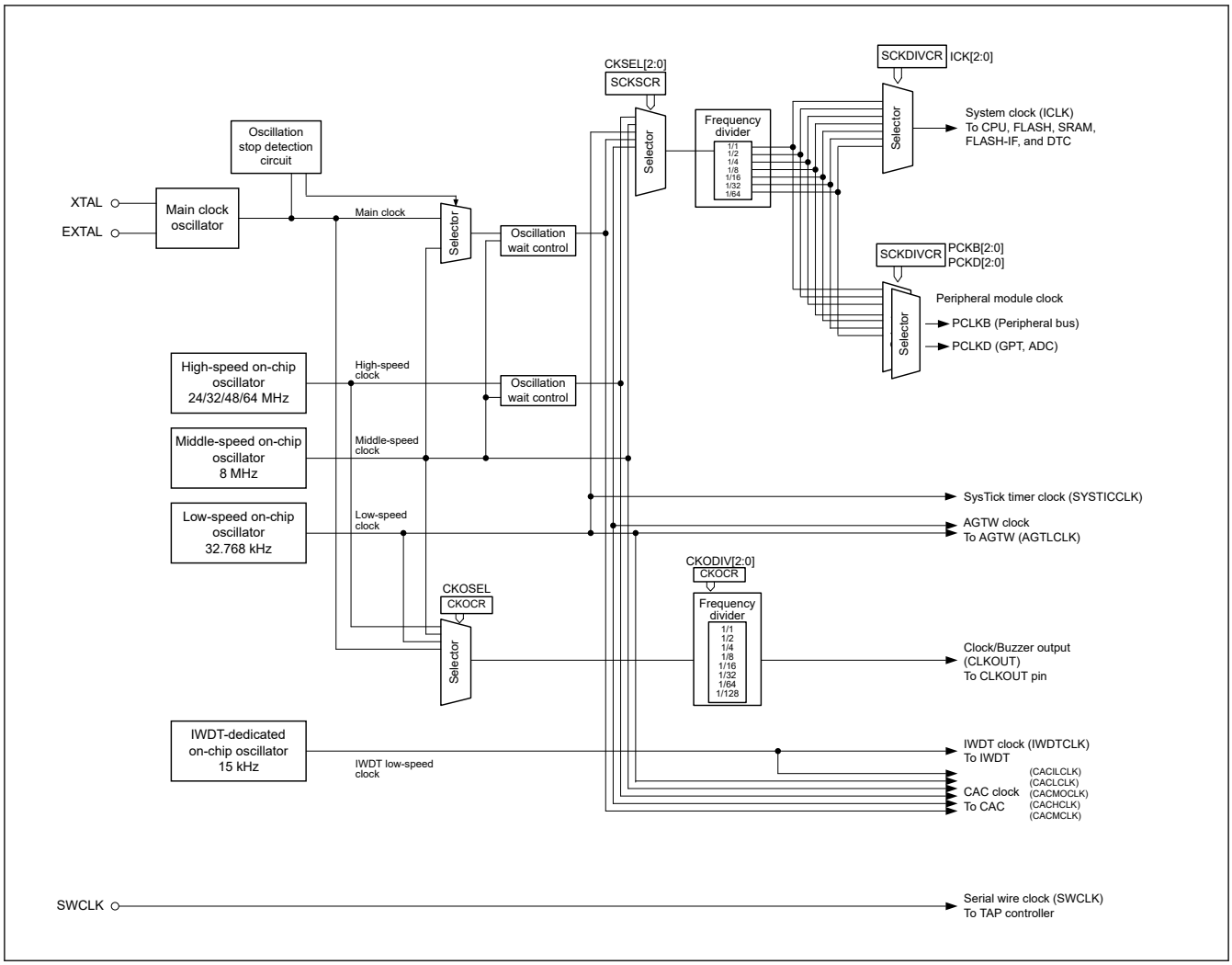


Figure 8.1 Clock generation circuit block diagram (Internal Clock Supply Architecture Type A)

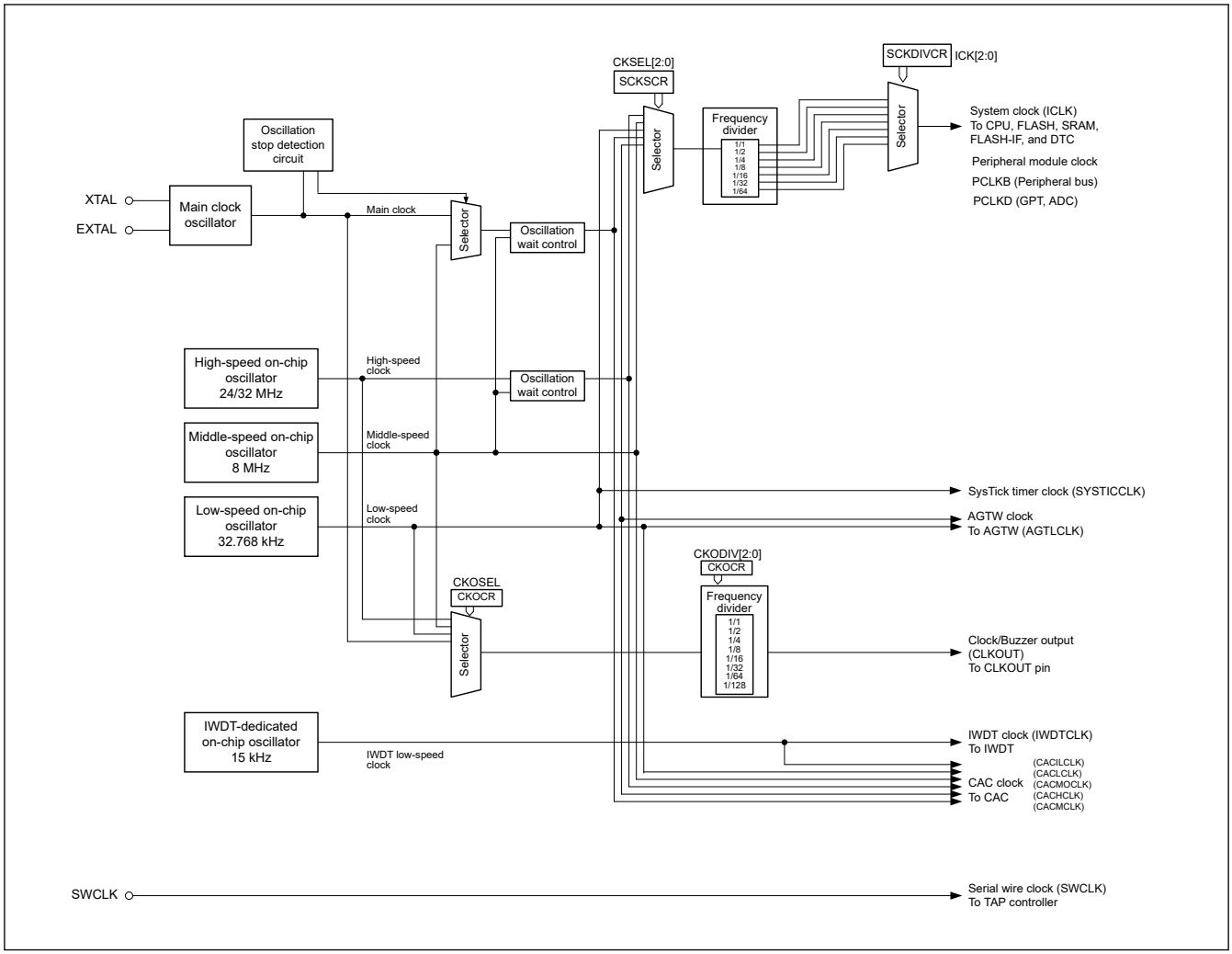


Figure 8.2 Clock generation circuit block diagram (Internal Clock Supply Architecture Type B)

Table 8.3 Clock generation circuit input/output pins

Pin name	I/O	Description
XTAL	Output	These pins are used to connect a crystal resonator. The EXTAL pin can also be used to input an external clock. For details, see section 8.3.2. External Clock Input.
EXTAL	Input	
CLKOUT	Output	This pin is used to output the CLKOUT/BUZZER clock
SWCLK	Input	This pin is used to input from the SWD

8.2 Register Descriptions

8.2.1 SCKDIVCR : System Clock Division Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	ICK[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	PCKD[2:0]	Peripheral Module Clock D (PCLKD) Select*2 *3 0 0 0: × 1/1 0 0 1: × 1/2 0 1 0: × 1/4 0 1 1: × 1/8 1 0 0: × 1/16 1 0 1: × 1/32 1 1 0: × 1/64 Others: Settings prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	PCKB[2:0]	Peripheral Module Clock B (PCLKB) Select*1 *3 0 0 0: × 1/1 0 0 1: × 1/2 0 1 0: × 1/4 0 1 1: × 1/8 1 0 0: × 1/16 1 0 1: × 1/32 1 1 0: × 1/64 Others: Settings prohibited	R/W
23:11	—	These bits are read as 0. The write value should be 0.	R/W
26:24	ICK[2:0]	System Clock (ICLK) Select*1 *2 *3 0 0 0: × 1/1 0 0 1: × 1/2 0 1 0: × 1/4 0 1 1: × 1/8 1 0 0: × 1/16 1 0 1: × 1/32 1 1 0: × 1/64 Others: Settings prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: For Internal Clock Supply Architecture Type B, the association between the frequencies of the system clock (ICLK), the peripheral module clock (PCLKB), and the peripheral module clock (PCLKD) should be ICLK : PCLKB : PCLKD = 1 : 1 : 1.
Write the same value to ICK[2:0], PCKB[2:0], and PCKD[2:0] when setting SCKDIVCR in Internal Clock Supply Architecture Type B.

Note 1. The association between the frequencies of the system clock (ICLK) and the peripheral module clock (PCLKB) should be ICLK:PCLKB = N:1 (N: integer).

Note 2. The association between the frequencies of the system clock (ICLK) and the peripheral module clock (PCLKD) should be ICLK:PCLKD = N:1 or 1:N (N: integer).

Note 3. Selecting division by 1 to ICLK is prohibited when SCKSCR.CKSEL[2:0] bits select the system clock source that is faster than 32 MHz and MEMWAIT.MEMWAIT = 0.

The SCKDIVCR register selects the frequencies of the system clock (ICLK), and peripheral module clock (PCLKB, PCLKD).

8.2.2 SCKSCR : System Clock Source Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x026

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

Bit	Symbol	Function	R/W
2:0	CKSEL[2:0]	Clock Source Select*1 0 0 0: HOCO 0 0 1: MOCO 0 1 0: LOCO 0 1 1: Main clock oscillator (MOSC)*2 Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Selecting a system clock source that is faster than 32 MHz (system clock source > 32 MHz) is prohibited when the SCKDIVCR.ICK[2:0] bits select division by 1 and MEMWAIT.MEMWAIT = 0.
Note 2. If the operating power control mode is the Low-speed mode, selecting the main clock oscillator as system clock source is prohibited.

The SCKSCR register selects the clock source for the system clock.

CKSEL[2:0] bits (Clock Source Select)

The CKSEL[2:0] bits select the source for the following modules:

- System clock (ICLK)
- Peripheral module clocks (PCLKB and PCLKD)

The bits select from one of the following sources:

- Low-speed on-chip oscillator (LOCO)
- Middle-speed on-chip oscillator (MOCO)
- High-speed on-chip oscillator (HOCO)
- Main clock oscillator (MOSC)

The operating state of each clock source is controlled not only by the clock oscillation enable settings but also by the operating modes of the product. Some clock sources might be forcibly stopped depending on the product operating mode being used.

Check the operation state of clock sources in each product operating mode, and do not select the clock source to be stopped in SCKSCR. The clock sources should be switched when there are no occurring internal asynchronous interrupt. For details, see section 10, Low Power Modes.

8.2.3 MEMWAIT : Memory Wait Cycle Control Register for Code Flash

Base address: SYSC = 0x4001_E000

Offset address: 0x031

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

Bit	Symbol	Function	R/W
0	MEMWAIT	Memory Wait Cycle Select for Code Flash 0: No wait 1: Wait	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/(W)

Note: Writing 0 to the MEMWAIT bit is prohibited when SCKDIVCR.ICK bit selects division by 1 and SCKSCR.CKSEL[2:0] bits select the system clock source that is faster than 32 MHz (ICLK > 32 MHz).
Note: For Internal Clock Supply Architecture Type B selected by OFS1.ICSATS bit, the MEMWAIT settings is prohibited.

The MEMWAIT register controls the wait cycle of code flash read access.

MEMWAIT bit (Memory Wait Cycle Select for Code Flash)

This bit selects the wait cycle of code flash read access. The wait cycle of code flash access is set to no wait (MEMWAIT = 0) after a reset is released.

Before writing to the MEMWAIT bit, check the ICLK frequency and operation power control mode. The following restrictions apply when setting the ICLK and operation power control mode, and the MEMWAIT bit:

- When setting the ICLK to faster than 32 MHz (ICLK > 32 MHz), set MEMWAIT to 1 while ICLK is 32 MHz or less (ICLK ≤ 32 MHz) and the operation power control mode is High-speed mode (OPCCR.OPCM[1:0] = 00b). Setting MEMWAIT to 1 is prohibited in operation modes other than High-speed mode. Setting the ICLK faster than 32 MHz is prohibited while MEMWAIT = 0.
- When setting the ICLK from 32 MHz or faster (ICLK > 32 MHz) to 32 MHz or less (ICLK ≤ 32 MHz), the ICLK frequency must be set to 32 MHz or less while MEMWAIT = 1. Setting MEMWAIT to 0 is prohibited while ICLK is faster than 32 MHz. Setting MEMWAIT to 1 is prohibited in operation modes other than High-speed mode. MEMWAIT can be set to 0 while the ICLK frequency is 32 MHz or less and operation power control mode is High-speed mode (OPCCR.OPCM[1:0] = 00b).

Table 8.4 MEMWAIT bit setting

MEMWAIT bit	MCU operation power control		
	Mode: except High-speed mode	High-speed mode	
		ICLK ≤ 32 MHz	ICLK > 32 MHz
0	✓	✓	—
1	—	✓	✓

Note: ✓ : Setting is possible.
— : Setting is not possible.

8.2.4 FLDWAITR : Memory Wait Cycle Control Register for Data Flash

Base address: FLCN = 0x407E_C000

Offset address: 0x3FC4

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	FLDWAIT1
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FLDWAIT1	Memory Wait Cycle Select for Data Flash 0: 1 wait access (Default) 1: 2 wait access	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Writing 0 to the FLDWAIT1 bit is prohibited when SCKDIVCR.ICK bit selects division by 1 and SCKSCR.CKSEL[2:0] bits select the system clock source that is faster than 32 MHz (ICLK > 32 MHz).
Note: For Internal Clock Supply Architecture Type B selected by OFS1.ICSATS bit, setting FLDWAIT1 is prohibited.
Note: There is no need to set FLDWAIT1 if data flash is not used.

The FLDWAITR register controls the wait cycle of data flash read access.

FLDWAIT1 bit (Memory Wait Cycle Select for Data Flash)

This bit selects the wait cycle of data flash read access. The wait cycle of data flash access is set to 1 wait (FLDWAIT1 = 0) after a reset is released.

The FLDWAIT1 settings for the data flash read access wait cycle are as follows:

- FLDWAIT1 = 0: 1 wait cycle
- FLDWAIT1 = 1: 2 wait cycles

Before writing to the FLDWAIT1 bit, check the ICLK frequency and operation power control mode. The following restrictions apply when setting the ICLK and operation power control mode, and the FLDWAIT1 bit:

- When setting the ICLK faster than 32 MHz (ICLK > 32 MHz), set FLDWAIT1 to 1 while ICLK is 32 MHz or less (ICLK ≤ 32 MHz) and the operation power control mode is High-speed mode (OPCCR.OPCM[1:0] = 00b). Setting FLDWAIT1 to 1 is prohibited in operation modes other than High-speed mode. Setting the ICLK faster than 32 MHz is prohibited while FLDWAIT1 = 0.
- When setting the ICLK from 32 MHz or faster (ICLK > 32 MHz) to 32 MHz or less (ICLK ≤ 32 MHz), the ICLK frequency must be set to 32 MHz or less while FLDWAIT1 = 1. Setting FLDWAIT1 to 0 is prohibited while ICLK is faster than 32 MHz. Setting FLDWAIT1 to 1 is prohibited in operation modes other than High-speed mode. FLDWAIT1 can be set to 0 while the ICLK frequency is 32 MHz or less and operation power control mode is High-speed mode (OPCCR.OPCM[1:0] = 00b).

Table 8.5 FLDWAIT1 bit setting

FLDWAIT1 bit	MCU operation power control		
	Mode: except High-speed mode	High-speed mode	
		ICLK ≤ 32 MHz	ICLK > 32 MHz
0	✓	✓	—
1	—	✓	✓

Note: ✓ : Setting is possible.
— : Setting is not possible.

Figure 8.3 shows an example flow when setting ICLK faster than 32 MHz.

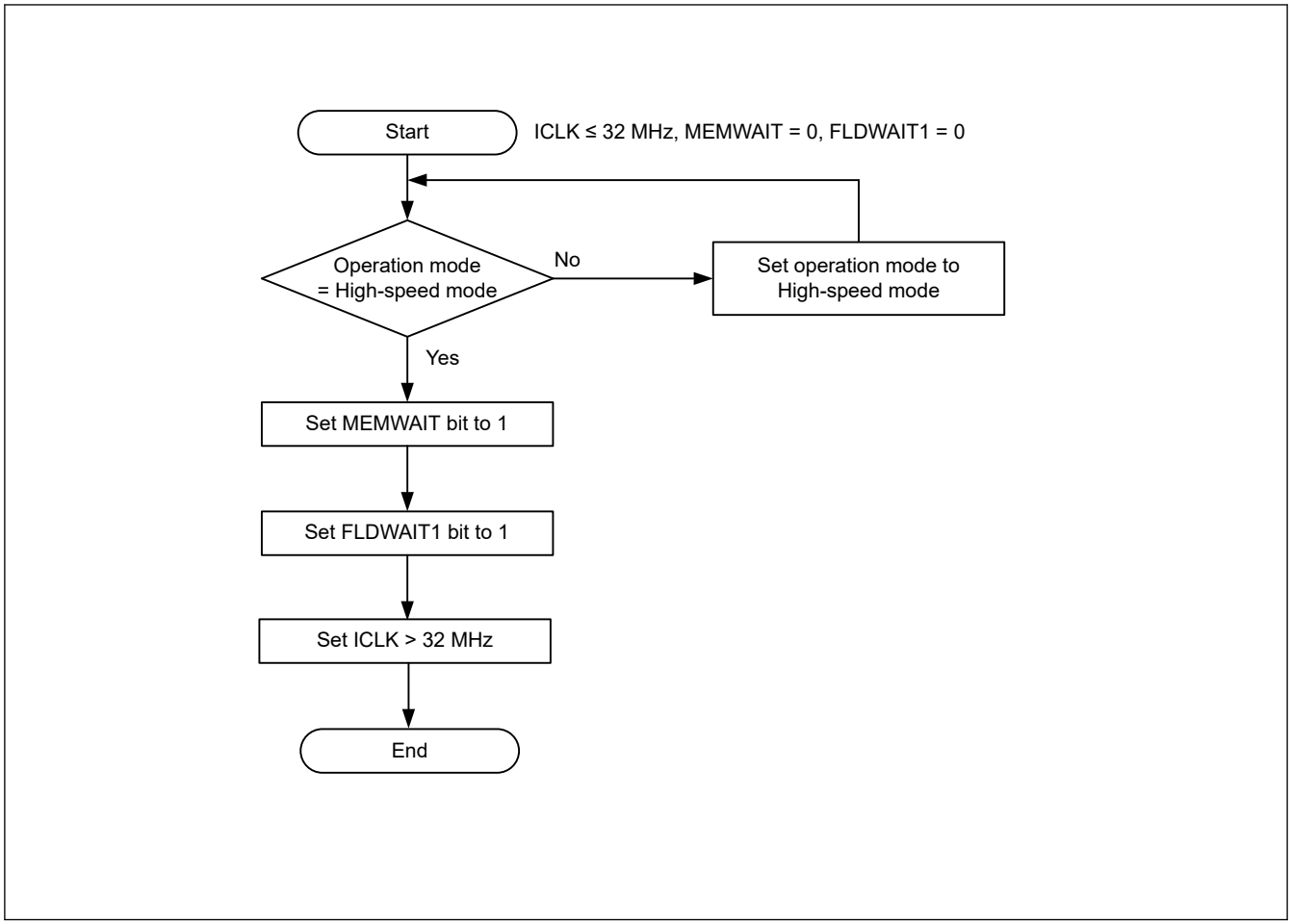


Figure 8.3 When setting ICLK > 32 MHz

Figure 8.4 shows an example flow when setting ICLK less than or equal to 32 MHz when ICLK is greater than 32 MHz.

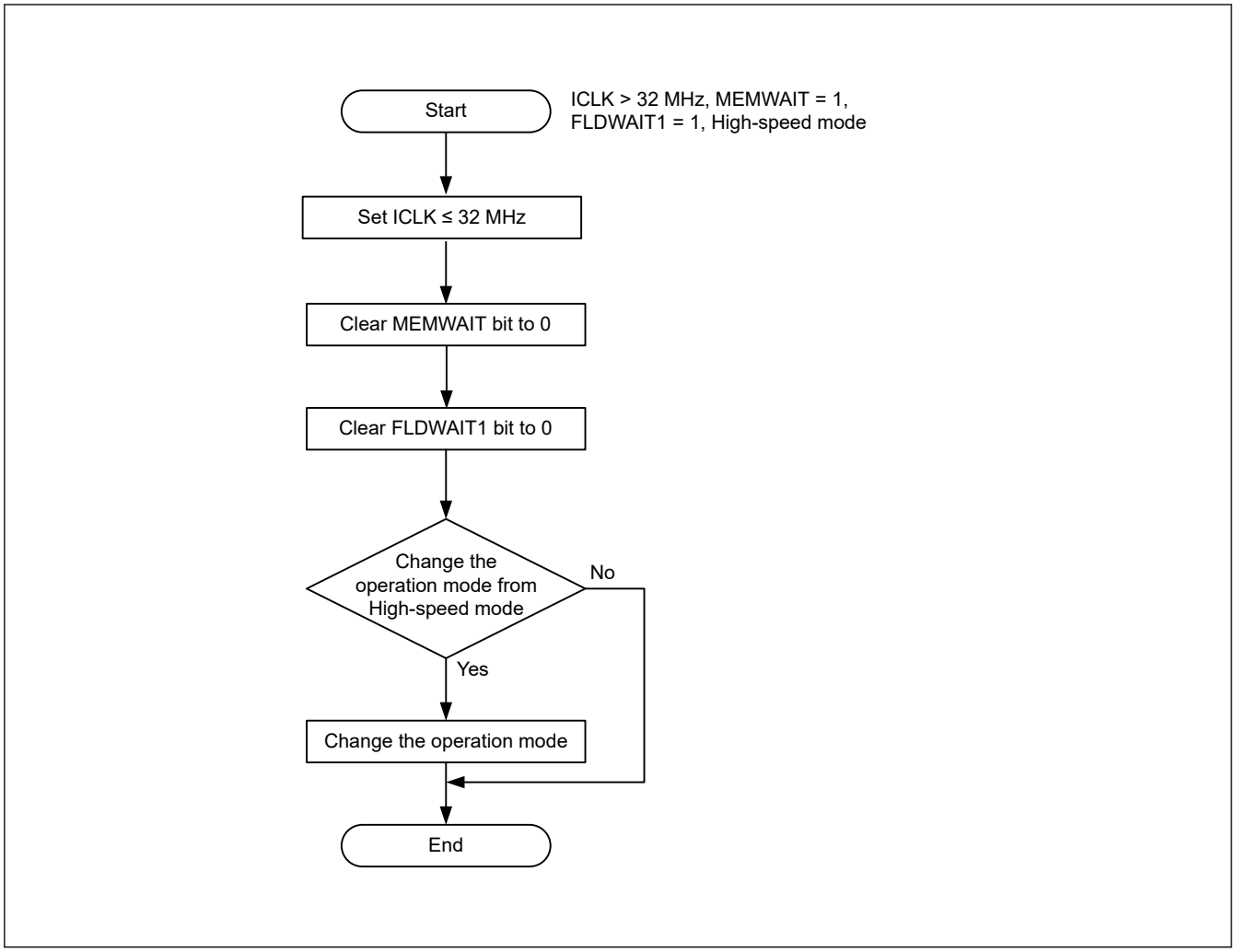


Figure 8.4 When setting ICLK ≤ 32 MHz from ICLK > 32 MHz

8.2.5 MOSCCR : Main Clock Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x032

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

Bit	Symbol	Function	R/W
0	MOSTP	Main Clock Oscillator Stop 0: Operate the main clock oscillator*1 1: Stop the main clock oscillator	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. MOMCR register must be set before setting MOSTP to 0.

The MOSCCR register controls the main clock oscillator.

MOSTP bit (Main Clock Oscillator Stop)

The MOSTP bit starts or stops the main clock oscillator.

When changing the value of the MOSTP bit, execute subsequent instructions only after reading the bit to check that the value is updated.

When using the main clock, the Main Clock Oscillator Mode Oscillation Control Register (MOMCR) and the Main Clock Oscillator Wait Control Register (MOSCWTCR) must be set before setting MOSTP to 0. After setting the MOSTP bit to 0, confirm that the OSCSF.MOSCSF bit is set to 1 before using the main clock oscillator.

A fixed stabilization wait time is required after setting the main clock oscillator to start operation. A fixed wait time is also required for oscillation to stop after stopping the main clock oscillator.

The following restrictions apply when starting and stopping operation:

- After stopping the main clock oscillator, confirm that the OSCSF.MOSCSF bit is 0 before restarting the main clock oscillator
- Confirm that the main clock oscillator operates and that the OSCSF.MOSCSF bit is 1 before stopping the main clock oscillator
- Regardless of whether the main clock oscillator is selected as the system clock, confirm that the OSCSF.MOSCSF bit is set to 1 before executing a WFI instruction to place the MCU in Software Standby after operating the main clock oscillator mode.
- When a transition to Software Standby mode is to follow the setting to stop the main clock oscillator, confirm that the OSCSF.MOSCSF bit is set to 0 before executing the WFI instruction.

Writing 1 to MOSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC).

Operation of the main clock oscillator (MOSTP = 0) is prohibited in the Low-speed mode.

8.2.6 LOCOCR : Low-Speed On-Chip Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x490

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

Bit	Symbol	Function	R/W
0	LCSTP	LOCO Stop 0: Operate the LOCO clock 1: Stop the LOCO clock	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The LOCOCR register controls the LOCO clock.

LCSTP bit (LOCO Stop)

The LCSTP bit starts or stops the LOCO clock.

After setting the LCSTP bit to 0 to start the LOCO clock, only use the clock after the LOCO clock-oscillation stabilization wait time (t_{LOCOWT}) elapses. A fixed stabilization wait time is required after setting the LOCO clock to start operation. A fixed wait time is also required after setting the LOCO clock to stop.

The following restrictions apply when starting and stopping operation:

- After stopping the LOCO clock, allow a stop interval of at least 5 LOCO clock cycles before restarting it
- Confirm that LOCO oscillation is stable before stopping the LOCO clock
- Regardless of whether the LOCO is selected as the system clock, confirm that LOCO oscillation is stable before executing a WFI instruction to place the MCU in Software Standby mode

- When a transition to Software Standby mode is to follow the setting to stop the LOCO clock, wait for at least 3 LOCO cycles before executing the WFI instruction.

Writing 1 to LCSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 010b (system clock source = LOCO).

Because the LOCO clock measures the wait time for other oscillators, it continues to oscillate while measuring this time, regardless of the setting in LOCOCR.LCSTP. As a result, the LOCO clock might be unintentionally supplied even when the LCSTP is set to stop.

8.2.7 HOCOCR : High-Speed On-Chip Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x036

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

Bit	Symbol	Function	R/W
0	HCSTP	HOCO Stop 0: Operate the HOCO clock *3 1: Stop the HOCO clock	R/W*2
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note: Set the HOCOWTCR.HSTS[2:0] bits to 011b before starting the HOCO clock by setting this register.
Note: Writing to OPCCR.OPCM[1:0] is prohibited while HOCOCR.HCSTP = 0 and OSCSF.HOCOSF = 0 (HOCO is in stabilization wait counting).
Note 1. The HCSTP bit value after a reset is 0 when the OFS1.HOCOEN bit is 0. It is 1 when the OFS1.HOCOEN bit is 1.
Note 2. Writing HCSTP is prohibited while OPCCR.OPCMTSF = 1 or SOPCCR.SOPCMTSF = 1 (during transition of operating power control mode)
Note 3. If you are using the HOCO (HCSTP = 0), set the OFS1.HOCOFRQ1[2:0] bit to an optimum value.

The HOCOCR register controls the HOCO clock.

HCSTP bit (HOCO Stop)

The HCSTP bit starts or stops the HOCO clock.

After setting the HCSTP bit to 0 to start the HOCO clock, confirm that the OSCSF.HOCOSF is set to 1 before using the clock. When OFS1.HOCOEN is set to 0, confirm that OSCSF.HOCOSF is also set to 1 before using the HOCO clock. A fixed stabilization wait time is required after setting the HOCO clock to start operation. A fixed wait time is also required after setting the HOCO clock to stop.

The following limitations apply when starting and stopping operation:

- After stopping the HOCO clock, confirm that the OSCSF.HOCOSF is 0 before restarting the HOCO clock.
- Confirm that the HOCO clock operates and that the OSCSF.HOCOSF is 1 before stopping the HOCO clock.
- Regardless of whether the HOCO clock is selected as the system clock, confirm that the OSCSF.HOCOSF is set to 1 before executing a WFI instruction to place the MCU in Software Standby mode after setting HOCO operation with the HCSTP bit.
- When a transition to Software Standby mode is to follow the setting of the HOCO clock to stop, confirm that the OSCSF.HOCOSF is set to 0 after setting the HOCO clock and before executing the WFI instruction.

Writing 1 to HCSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 000b (system clock source = HOCO).

8.2.8 MOCOCR : Middle-Speed On-Chip Oscillator Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x038

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

Bit	Symbol	Function	R/W
0	MCSTP	MOCO Stop 0: MOCO clock is operating 1: MOCO clock is stopped	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The MOCOCR register controls the MOCO clock.

MCSTP bit (MOCO Stop)

The MCSTP bit starts or stops the MOCO clock.

After setting MCSTP to 0, use the MOCO clock only after the MOCO clock oscillation stabilization time (t_{MOCOWT}) elapses. A fixed stabilization wait time is required after setting the MOCO clock to start operation. A fixed wait time is also required for oscillation to stop after setting the MOCO clock to stop operation.

The following restrictions apply when starting and stopping the oscillator:

- After stopping the MOCO clock, allow a stop interval of at least 5 MOCO clock cycles before restarting it
- Confirm that MOCO clock oscillation is stable before stopping the MOCO clock
- Regardless of whether the MOCO clock is selected as the system clock, confirm that MOCO clock oscillation is stable before executing a WFI instruction to place the MCU in Software Standby mode
- When a transition to Software Standby mode is to follow the setting to stop the MOCO clock, wait for at least 3 MOCO clock cycles before executing the WFI instruction.

Writing 1 to MCSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 001b (system clock source = MOCO).

Writing 1 to the MCSTP bit (stopping the MOCO) is prohibited if oscillation stop detection is enabled in the Oscillation Stop Detection Control Register (OSTDCR.OSTDE).

8.2.9 OSCSF : Oscillation Stabilization Flag Register

Base address: SYSC = 0x4001_E000

Offset address: 0x03C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MOSCSF	—	—	HOCOSF
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Function	R/W
0	HOCOSF	HOCO Clock Oscillation Stabilization Flag 0: The HOCO clock is stopped or is not yet stable 1: The HOCO clock is stable, so is available for use as the system clock	R
2:1	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
3	MOSCSF	Main Clock Oscillation Stabilization Flag 0: The main clock oscillator is stopped (MOSTP = 1) or is not yet stable*2 1: The main clock oscillator is stable, so is available for use as the system clock	R
7:4	—	These bits are read as 0.	R

Note 1. The value after reset depends on the OFS1.HOCOEN setting.
When OFS1.HOCOEN = 1 (disable HOCO), the value after reset of HOCOSF is 0.
When OFS1.HOCOEN = 0 (enable HOCO), the HOCOSF value is set to 0 immediately after reset is released, and the HOCOSF value is set to 1 after the HOCO oscillation stabilization wait time elapses.
Note 2. This is true when an appropriate value is set in the Wait Control register for the main clock oscillator. If the wait time value is not sufficient, the oscillation stabilization flag is set to 1 and supply of the clock signal to the internal circuits starts before oscillation is stable.

The OSCSF register contains flags to indicate the operating status of the counters in the oscillation stabilization wait circuits for the individual oscillators. After oscillation starts, these counters measure the wait time until each oscillator output clock is supplied to the internal circuits. An overflow of a counter indicates that the clock supply is stable and available for the associated circuit.

HOCOSF flag (HOCO Clock Oscillation Stabilization Flag)

The HOCOSF flag indicates the operating status of the counter that measures the wait time for the high-speed clock oscillator (HOCO). When OFS1.HOCOEN is set to 0, confirm that OSCSF.HOCOSF is set to 1 before using the HOCO clock.

[Setting condition]

- When the HOCO clock is stopped and the HOCOCR.HCSTP bit is set to 0, and then the HOCO oscillation stabilization time is counted by the MOCO clock and supply of the HOCO clock within the MCU is started. For the HOCO oscillation stabilization time, see section 35, Electrical Characteristics.

[Clearing condition]

- When the HOCO clock is operating and then is deactivated because the HOCOCR.HCSTP bit is set to 1.

MOSCSF flag (Main Clock Oscillation Stabilization Flag)

The MOSCSF flag indicates the operating status of the counter that measures the wait time for the main clock oscillator.

[Setting condition]

- When the main clock oscillator is stopped and the MOSCCR.MOSTP bit is set to 0, and then the number of MOCO clock cycles corresponding to the setting of the MOSCWTCR register is counted and supply of the main clock within the MCU is started.

[Clearing condition]

- When the main clock oscillator is operating and then is deactivated because the MOSCCR.MOSTP bit is set to 1.

8.2.10 OSTDCR : Oscillation Stop Detection Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x040

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

Bit	Symbol	Function	R/W
0	OSTDIE	Oscillation Stop Detection Interrupt Enable 0: Disable oscillation stop detection interrupt (do not notify the POEG) 1: Enable oscillation stop detection interrupt (notify the POEG)	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	OSTDE	Oscillation Stop Detection Function Enable 0: Disable oscillation stop detection function 1: Enable oscillation stop detection function	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The OSTDCR register controls the oscillation stop detection function.

OSTDIE bit (Oscillation Stop Detection Interrupt Enable)

The OSTDIE bit enables the oscillation stop detection function interrupt. It also controls whether oscillation stop detection is reported to the POEG.

If the Oscillation Stop Detection flag in the Oscillation Stop Detection Status Register (OSTDSR.OSTDF) requires clearing, set the OSTDIE bit to 0 before clearing OSTDF. Wait for at least 2 PCLKB cycles before setting the OSTDIE bit to 1. By reading the I/O register whose access cycle number is defined by PCLKB, it is possible to secure waiting time of 2 or more cycles of PCLKB.

OSTDE bit (Oscillation Stop Detection Function Enable)

The OSTDE bit enables the oscillation stop detection function.

When the OSTDE bit is 1 (enabled), the MOCO stop bit (MOCOCR.MCSTP) is set to 0 and the MOCO operation starts. The MOCO clock cannot be stopped while the oscillation stop detection function is enabled. Writing 1 to the MOCOCR.MCSTP bit (MOCO stopped) is invalid.

When the Oscillation Stop Detection flag in the Oscillation Stop Detection Status Register (OSTDSR.OSTDF) is 1 (main clock oscillation stop detected), writing 0 to the OSTDE bit is invalid.

The OSTDE bit must be set to 0 before transitioning to Software Standby mode. To transition to Software Standby mode, first set the OSTDE bit to 0, then execute the WFI instruction.

The following restrictions apply when using the oscillation stop detection function:
In low-speed mode, selecting division by 1, 2 for ICLK, PCLKB, and PCLKD is prohibited

8.2.11 OSTDSR : Oscillation Stop Detection Status Register

Base address: SYSC = 0x4001_E000

Offset address: 0x041

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

Bit	Symbol	Function	R/W
0	OSTDF	Oscillation Stop Detection Flag 0: Main clock oscillation stop not detected 1: Main clock oscillation stop detected	R/W*1
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. This bit can only be set to 0. This bit is cleared to 0 by writing 0 after reading it as 1.

The OSTDSR register indicates the stop detection status of the main clock oscillator.

OSTDF flag (Oscillation Stop Detection Flag)

The OSTDF flag indicates the main clock oscillator status. When this flag is 1, it indicates that the main clock oscillation stop was detected. After this stop is detected, the OSTDF flag is not set to 0 even when the main clock oscillation is restarted. The OSTDF bit is cleared to 0 by writing 0 after reading it as 1.

At least 3 ICLK cycles of wait time are required between writing 0 to OSTDF and reading it as 0. If the OSTDF flag is set to 0 when the main clock oscillation is stopped, the OSTDF flag becomes 0 then returns to 1.

The OSTDF flag cannot be set to 0 under the following conditions:

- SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC).

The OSTDF flag must be set to 0 after switching the clock source to sources other than the main clock oscillator.

[Setting condition]

- The main clock oscillator is stopped when OSTDCR.OSTDE = 1 (oscillation stop detection function enabled).

[Clearing condition]

- 1 is read and then 0 is written when the SCKSCR.CKSEL[2:0] bits are neither 011b (system clock is MOSC) nor 101b.

8.2.12 MOSCWTCR : Main Clock Oscillator Wait Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0A2

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

Bit	Symbol	Function	R/W
3:0	MSTS[3:0]	Main Clock Oscillator Wait Time Setting 0x0: Wait time = 2 cycles (0.25 µs) 0x1: Wait time = 1024 cycles (128 µs) 0x2: Wait time = 2048 cycles (256 µs) 0x3: Wait time = 4096 cycles (512 µs) 0x4: Wait time = 8192 cycles (1024 µs) 0x5: Wait time = 16384 cycles (2048 µs) 0x6: Wait time = 32768 cycles (4096 µs) 0x7: Wait time = 65536 cycles (8192 µs) 0x8: Wait time = 131072 cycles (16384 µs) 0x9: Wait time = 262144 cycles (32768 µs) Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

MSTS[3:0] bits (Main Clock Oscillator Wait Time Setting)

The MSTS[3:0] bits specify the oscillation stabilization wait time for the main clock oscillator.

Set the main clock oscillation stabilization time to a period longer than or equal to the stabilization time recommended by the oscillator manufacturer. When the main clock is input externally, set these bits to 0x0 because the oscillation stabilization time is not required.

The wait time set in these bits is counted using: 1 cycle (µs) = 1/(f_{MOCO}_typ [MHz]) = 1/8 = 0.125 (µs) (typ.) (f_{MOCO}_typ: typical frequency for MOCO) The MOCO clock automatically oscillates when necessary, regardless of the value of the MOCO.MCSTP bit. After the specified wait time elapses, supply of the main clock starts internally in the MCU, and the OSCSF.MOSCSF flag is set to 1. If the specified wait time is short, supply of the main clock starts before oscillation of the clock becomes stable.

Only rewrite the MOSCWTCR register when the MOSCCR.MOSTP bit is 1 and the OSCSF.MOSCSF flag is 0. Do not rewrite this register under any other conditions.

8.2.13 HOCOWTCR : High-Speed On-Chip Oscillator Wait Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0A5

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

Bit	Symbol	Function	R/W
2:0	HSTS[2:0]	HOCO Wait Time Setting 1 0 1: Value after reset. 0 1 1: Before starting high-speed on-chip oscillator by setting HOCOCR.HCSTP bit, the HSTS[2:0] bits must be set to 011b beforehand. Wait time = 46 cycles (5.75 µs) Wait time is calculated at MOCO = 8 MHz (typically 0.125 µs). Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

HOCOWTCR controls the wait time until output of the signal from the high-speed on-chip oscillator to the internal circuits starts. Only write to HOCOWTCR when the HOCOCR.HCSTP bit is 1 or the OSCSF.HOCOSF flag is 1. Do not write to this register under any other conditions.

HSTS[2:0] bits (HOCO Wait Time Setting)

The oscillation stabilization wait circuit measures the wait time and controls the clock supply in the MCU. When the high-speed on-chip oscillator starts, the oscillation stabilization wait circuit starts counting cycles of the middle-speed clock set in the HOCOWTCR register. The MCU clock supply is disabled until counting of the set number of cycles is complete. After counting completes, supply of the clock signal in the MCU starts and the OSCSF.HOCOSF flag is set to1.

The oscillation stabilization wait circuit continues to count the middle-speed clock cycles regardless of the MOCOCR.MCSTP bit setting. Hardware automatically controls the running and stopping of the middle-speed on-chip oscillator for wait time measurement.

8.2.14 MOMCR : Main Clock Oscillator Mode Oscillation Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x413

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	MOSEL	—	—	MODRV1	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	MODRV1	Main Clock Oscillator Drive Capability 1 Switching 0: 10 MHz to 20 MHz 1: 1 MHz to 10 MHz	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	MOSEL	Main Clock Oscillator Switching 0: Resonator 1: External clock input	R/W
7	—	These bits are read as 0. The write value should be 0.	R/W

Note: The EXTAL/XTAL pins are also used as ports. In the initial state, the pin is set as a port.
Note: The MOSCCR.MOSTP bit must be 1 (MOSC is stopped) before changing this register.

MODRV1 bit (Main Clock Oscillator Drive Capability 1 Switching)

The MODRV1 bit switches the drive capability of the main clock oscillator.

MOSEL bit (Main Clock Oscillator Switching)

The MOSEL bit switches the source for the main clock oscillator.

8.2.15 CKOCR : Clock Out Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x03E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CKOEN	CKODIV[2:0]			—	CKOSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CKOSEL[2:0]	Clock Out Source Select 0 0 0: HOCO (value after reset) 0 0 1: MOCO 0 1 0: LOCO 0 1 1: MOSC Others: Setting prohibited	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	CKODIV[2:0]	Clock Output Frequency Division Ratio 0 0 0: × 1/1 0 0 1: × 1/2 0 1 0: × 1/4 0 1 1: × 1/8 1 0 0: × 1/16 1 0 1: × 1/32 1 1 0: × 1/64 1 1 1: × 1/128	R/W
7	CKOEN	Clock Out Enable 0: Disable clock out 1: Enable clock out	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

CKOSEL[2:0] bits (Clock Out Source Select)

The CKOSEL[2:0] bits select the source of the clock to be output from the CLKOUT pin. When changing the clock source, set the CKOEN bit to 0.

CKODIV[2:0] bits (Clock Output Frequency Division Ratio)

The CKODIV[2:0] bits specify the clock division ratio. Set the CKOEN bit to 0 when changing the division ratio. The division ratio of the output clock frequency must be set to a value no higher than the characteristics of the CLKOUT pin output frequency. For details on the characteristics of the CLKOUT pin, see section 35, Electrical Characteristics.

CKOEN bit (Clock Out Enable)

The CKOEN bit enables output from the CLKOUT pin.

When this bit is set to 1, the selected clock is output. When this bit is set to 0, low is output. When changing this bit, confirm that the clock out source clock selected in the CKOSEL[2:0] bits is stable. Otherwise, a glitch might be generated in the output.

Clear this bit before entering Software Standby mode if the selecting clock out source clock is stopped in that mode.

8.2.16 LOCOUTCR : LOCO User Trimming Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x492

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

Bit	Symbol	Function	R/W
7:0	LOCOUTRM[7:0]	LOCO User Trimming 0xF8: -8 0xF9: -7 ⋮ 0xFF: -1 0x00: 0 0x01: +1 ⋮ 0x06: +6 0x07: +7	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The LOCOUTCR register is added to the original LOCO trimming data.

The LOCO frequency can be trimmed with a resolution of approximately 3.9% per bit by setting a trimming value in the LOCOUTCR register.

Increasing the trimming value makes the LOCO frequency higher.

Decreasing the trimming value makes the LOCO frequency lower.

MCU operation is not guaranteed when LOCOUTCR is set to a value that causes the LOCO frequency to be outside of the specification range. When LOCOUTCR is modified, the frequency stabilization time corresponds to the frequency stabilization time at the start of MCU operation. When the ratio of the LOCO frequency and the other oscillation frequency is an integer value, changing the LOCOUTCR value is prohibited.

Note: The frequency will vary if the temperature and the power supply voltage change after frequency trimming. In such case, it is essential to perform trimming regularly or before high frequency accuracy is required.

8.2.17 MOCOUTCR : MOCO User Trimming Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x061

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

Bit	Symbol	Function	R/W
7:0	MOCOUTRM[7:0]	MOCO User Trimming 0xF0: -16 0xF1: -15 ⋮ 0xFF: -1 0x00: 0 0x01: +1 ⋮ 0x0E: +14 0x0F: +15	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The MOCOUTCR register is added to the original MOCO trimming data.

The MOCO frequency can be trimmed with a resolution of approximately 1.2% per bit by setting a trimming value in the MOCOUTCR register.

Increasing the trimming value makes the MOCO frequency higher.

Decreasing the trimming value makes the MOCO frequency lower.

MCU operation is not guaranteed when MOCOUTCR is set to a value that causes the MOCO frequency to be outside of the specification range. When MOCOUTCR is modified, the frequency stabilization wait time corresponds to the frequency stabilization wait time at the start of the MCU operation. When the ratio of the MOCO frequency and the other oscillation frequency is an integer value, changing the MOCOUTCR value is prohibited.

Note: The frequency will vary if the temperature and the power supply voltage change after frequency trimming. In such case, it is essential to perform trimming regularly or before high frequency accuracy is required.

8.2.18 HOCOUTCR : HOCO User Trimming Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x062

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

Bit	Symbol	Function	R/W
7:0	HOCOUTRM[7:0]	HOCO User Trimming 0xE0: -32 0xE1: -31 ⋮ 0xFF: -1 0x00: 0 0x01: +1 ⋮ 0x1E: +30 0x1F: +31	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The HOCOUTCR register is added to the original HOCO trimming data.

The HOCO frequency can be trimmed with a resolution of approximately 0.028% per bit by setting a trimming value in the HOCOUTCR register.

Increasing the trimming value makes the HOCO frequency higher.

Decreasing the trimming value makes the HOCO frequency lower.

MCU operation is not guaranteed when HOCOUTCR is set to a value that causes the HOCO frequency to be outside of the specification range. When HOCOUTCR is modified, the frequency stabilization wait time corresponds to the frequency stabilization wait time at the start of the MCU operation.

Note: The frequency will vary if the temperature and the power supply voltage change after frequency trimming. In such case, it is essential to perform trimming regularly or before high frequency accuracy is required.

8.3 Main Clock Oscillator

To supply the clock signal to the main clock oscillator, use one of the following ways:

- Connect an oscillator
- Connect the input of an external clock signal.

8.3.1 Connecting a Crystal Resonator

Figure 8.5 shows an example of connecting a crystal resonator. A damping resistor (Rd) can be added, if required.

Because the resistor values vary according to the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If the manufacturer recommends using an external feedback resistor (Rf), insert an Rf between EXTAL and XTAL by following the instructions.

When connecting a resonator to supply the clock, the frequency of the resonator must be in the frequency range of the resonator for the main clock oscillator as described in Table 8.1.

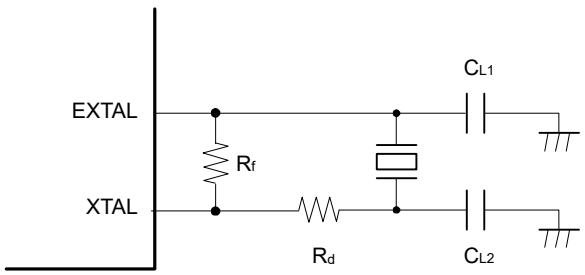


Figure 8.5 Example of crystal resonator connection

Figure 8.6 shows an equivalent circuit of the crystal resonator.

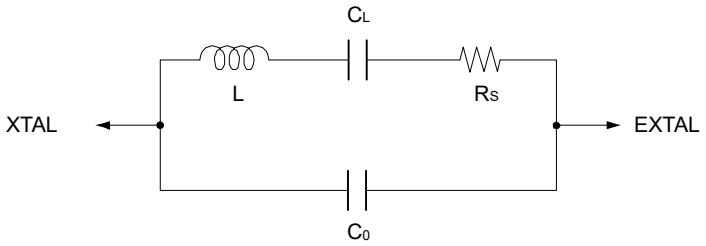


Figure 8.6 Equivalent circuit of the crystal resonator

8.3.2 External Clock Input

Figure 8.7 shows an example of connecting an external clock input. To operate the oscillator with an external clock signal, set the MOMCR.MOSEL bit to 1. The XTAL pin becomes high impedance.

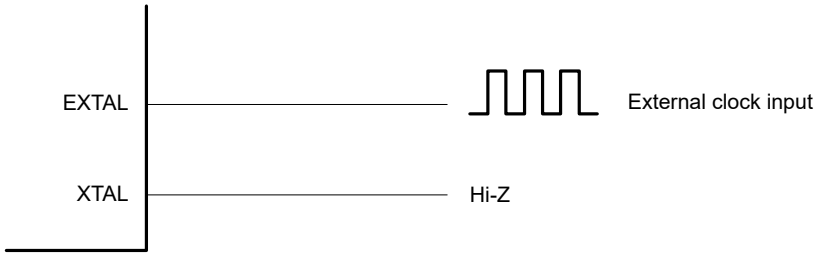


Figure 8.7 Equivalent circuit for external clock

8.3.3 Notes on External Clock Input

The frequency of the external clock input can only be changed when the main clock oscillator is stopped. Do not change the frequency of the external clock input when the setting of the Main Clock Oscillator Stop bit (MOSCCR.MOSTP) is 0.

8.4 Oscillation Stop Detection Function

8.4.1 Oscillation Stop Detection and Operation after Detection

The oscillation stop detection function detects the main clock oscillator stop. When oscillation stop is detected, the system clock switches as follows:

- If an oscillation stop is detected with SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC), the system clock source switches to the MOCO clock.

An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the General PWM Timer (GPT) output can be forced to a high-impedance state on detection.

The main clock oscillation stop is detected when the input clock remains at 0 or 1 for a certain period, for example, when a malfunction occurs in the main clock oscillator. See section 35, Electrical Characteristics.

Switching between the main clock oscillator and the MOCO clock is controlled by the Oscillation Stop Detection Flag (OSTDSR.OSTDF).

OSTDF controls the switched clock as follows:

- When SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC):
  - When OSTDF changes from 0 to 1, the clock source switches to the MOCO clock.
  - When OSTDF changes from 1 to 0, the clock source switches back to MOSC.

Note: When SCKSCR.CKSEL[2:0] ≠ 011b, can not set 0 to OSTDF.

To switch the clock source to the main clock again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock and clear the OSTDF flag to 0. Also, check that the OSTDF flag is not 1, then set the CKSEL[2:0] bits to the main clock after the specified oscillation stabilization time elapses.

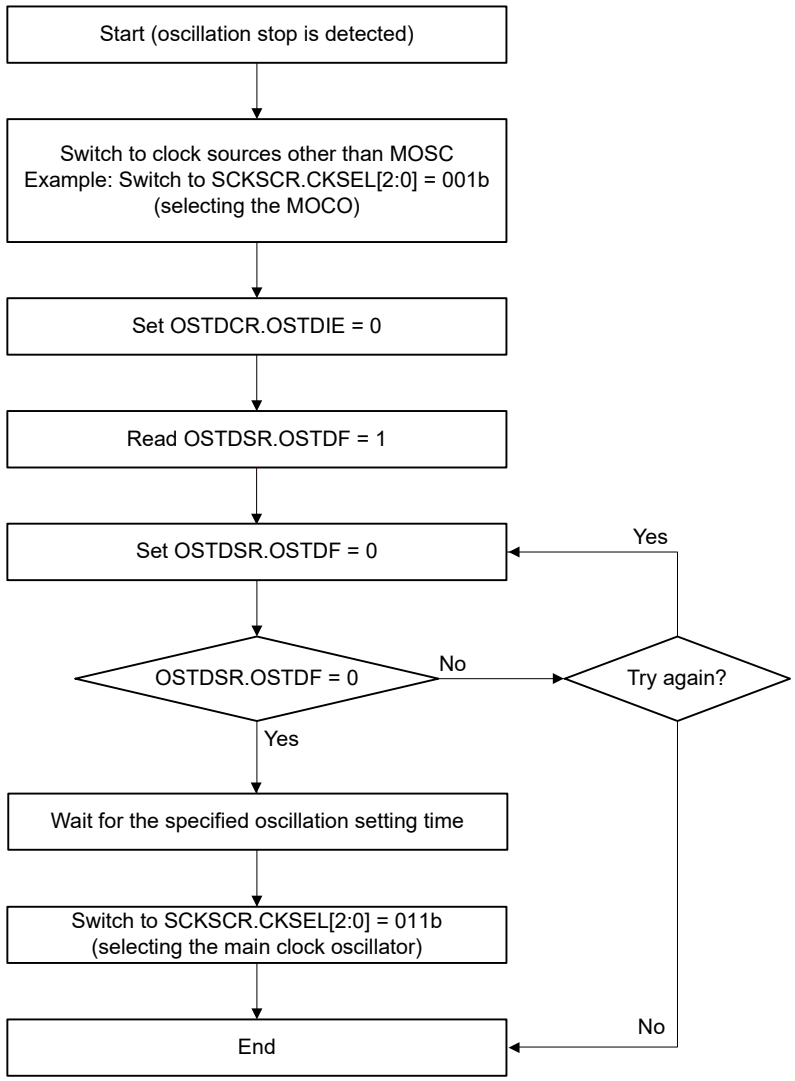
After a reset release, the main clock oscillator is stopped and the oscillation stop detection function is disabled. To enable the oscillation stop detection function, activate the main clock oscillator and write 1 to the oscillation stop detection function enable bit (OSTDCR.OSTDE) after a specified oscillation stabilization time elapses.

The oscillation stop detection function detects when the main clock is stopped by an external cause. Therefore, the oscillation stop detection function must be disabled before the main clock oscillator is stopped by software or a transition is made to Software Standby mode.

The oscillation stop detection function switches all clocks that can be selected as the MOSC clock except CLKOUT to the MOCO (when system clock is MOSC).

The system clock (ICLK) frequency during the MOCO(when system clock is MOSC) operation is specified by the MOCO oscillation frequency and the division ratio set by the system clock select bits (SCKDIVCR.ICK[2:0])

Example of returning when CKSEL[2:0] = 011b (selecting the main clock oscillator) after an oscillation stop is detected



Note: On return from the oscillation-stopped state, the factor responsible for stopping the main clock oscillation circuit must be removed from the system to allow oscillation to resume.

Figure 8.8 Flow of recovery on detection of oscillator stop

8.4.2 Oscillation Stop Detection Interrupts

An oscillation stop detection interrupt (MOSC_STOP) is generated when the Oscillation Stop Detection Flag (OSTDSR.OSTDF) is 1 and the Oscillation Stop Detection Interrupt Enable bit in the Oscillation Stop Detection Control Register (OSTDCR.OSTDIE) is 1 (enabled). The Port Output Enable for GPT (POEG) is notified of the main clock oscillator stop. On receiving the notification, the POEG sets the Oscillation Stop Detection Flag in the POEG Group n Setting Register (POEGGn.OSTPF) to 1 (n = A, B).

After the oscillation stop is detected, wait at least 10 PCLKB clock cycles before writing to the POEGGn.OSTPF flag. When the OSTDSR.OSTDF flag requires clearing, do so after clearing the Oscillation Stop Detection Interrupt Enable bit in the Oscillation Stop Detection Control Register (OSTDCR.OSTDIE). Wait at least 2 PCLKB clock cycles before setting the OSTDCR.OSTDIE bit to 1 again. A longer PCLKB wait time might be required, depending on the number of cycles required to read a given I/O register.

The oscillation stop detection interrupt is a non-maskable interrupt. Because non-maskable interrupts are disabled in the initial state after a reset release, enable non-maskable interrupts through software before using oscillation stop detection interrupts. For details, see section 12, Interrupt Controller Unit (ICU).

8.5 Internal Clock

Clock sources for the internal clock signals include:

- Main clock
- HOCO clock
- MOCO clock
- LOCO clock
- IWDT-dedicated clock

The following internal clocks are produced from these sources.

- Operating clock of the CPU, DTC, Flash, Flash-IF, and SRAM — system clock (ICLK)
- Operating clocks of peripheral modules — PCLKB and PCLKD
- Operating clocks for the CAC — CACCLK
- Operating clock for the IWDT — IWDTCLK
- Operating clock for the AGTW LOCO clock — AGTLCLK
- Operating clock for the SysTick timer — SYSTICCLK
- Clock for external pin output — CLKOUT.

For details of the registers used to set the frequencies of the internal clocks, see section 8.5.1. System Clock (ICLK) to section 8.5.7. External Pin Output Clock (CLKOUT).

If the value of any of these bits is changed, subsequent operation is at the frequency determined by the new value.

8.5.1 System Clock (ICLK)

The system clock (ICLK) is the operating clock for the CPU, DTC, Flash, Flash-IF, and SRAM.

The ICLK frequency is specified by the HOCOFRQ1[2:0] bits in OFS1, the ICK[2:0] bits in SCKDIVCR, and the CKSEL[2:0] bits in SCKSCR.

When the ICLK clock source is switched, the duration of the ICLK clock cycle becomes longer during the clock source transition period. See Figure 8.9 and Figure 8.10.

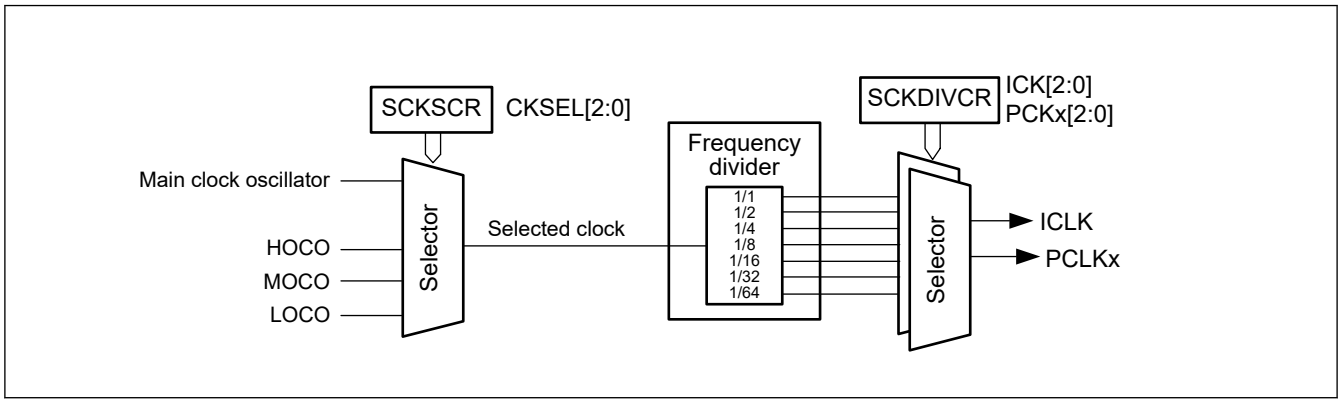


Figure 8.9 Block diagram of clock source selector

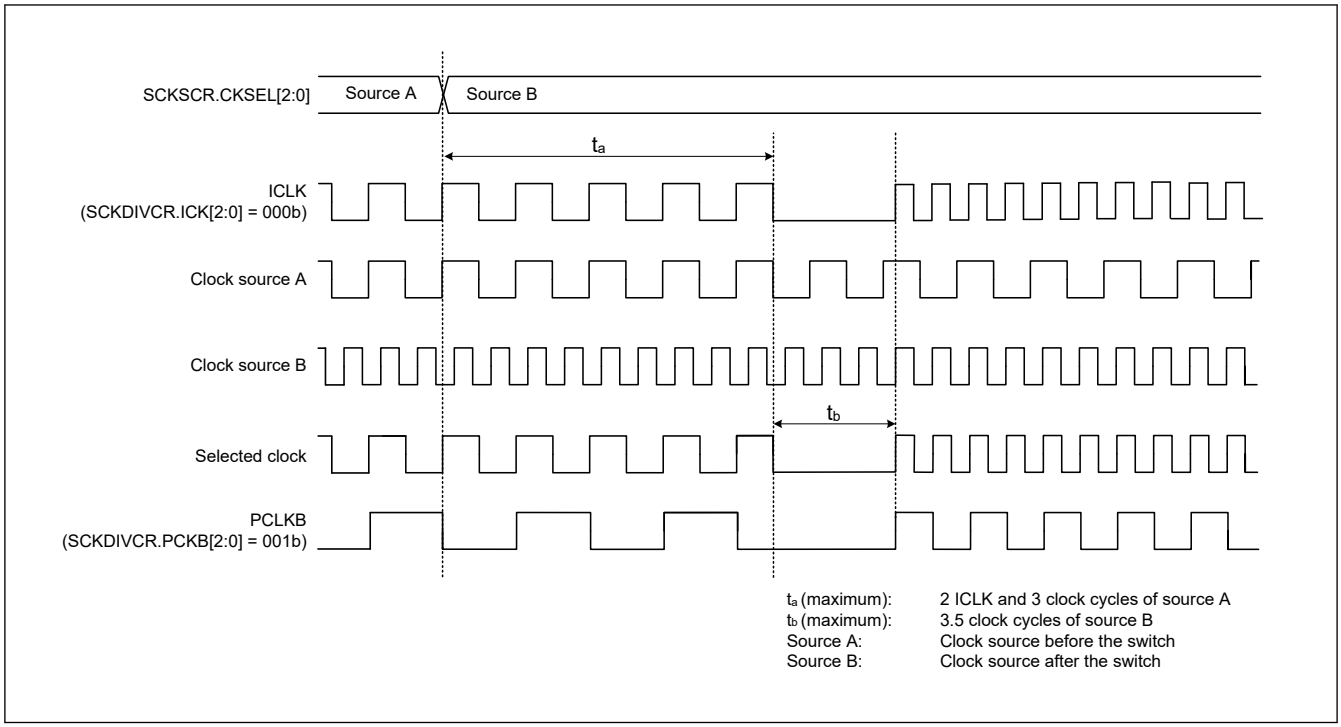


Figure 8.10 Timing of clock source switching

8.5.2 Peripheral Module Clock (PCLKB, PCLKD)

The peripheral module clocks (PCLKB and PCLKD) are the operating clocks for the peripheral modules.

The frequency of the given clock is specified in the following bits:

- HOCOFRQ1[2:0] in OFS1.
- PCKB[2:0] and PCKD[2:0] in SCKDIVCR
- CKSEL[2:0] in SCKSCR

When the clock source of the peripheral module clock is switched, the duration of the peripheral module clock cycle becomes longer during the clock source transition period. See Figure 8.9 and Figure 8.10.

8.5.3 CAC Clock (CACCLK)

The CAC clock (CACCLK) is the operating clock for the CAC. CACCLK is generated by the following oscillators:

- Main clock oscillator
- High-speed clock oscillator (HOCO)
- Middle-speed clock oscillator (MOCO)
- Low-speed on-chip oscillator (LOCO)
- IWDT-dedicated on-chip oscillator. (IWDTLOCO)

8.5.4 IWDT-Dedicated Clock (IWDTCLK)

The IWDT-dedicated clock (IWDTCLK) is the operating clock for the IWDT. IWDTCLK is internally generated by the IWDT-dedicated on-chip oscillator.

8.5.5 AGTW-Dedicated Clock (AGTLCLK)

The AGTW-dedicated clocks (AGTLCLK) are the operating clocks for the AGTW. AGTLCLK is generated by the LOCO clock.

8.5.6 SysTick Timer-Dedicated Clock (SYSTICCLK)

The SysTick timer-dedicated clock (SYSTICCLK) is the operating clock for the SysTick timer. SYSTICCLK is generated by the LOCO clock.

8.5.7 External Pin Output Clock (CLKOUT)

The CLKOUT is output externally from the CLKOUT pin for the clock or buzzer output. The CLKOUT is output to the CLKOUT pin when the CKOCR.CKOEN bit is set to 1. Only change the value in the CKODIV[2:0] bits or CKOSEL[2:0] bits in CKOCR when the CKOCR.CKOEN bit is 0.

The CLKOUT clock frequency is specified in the following bits:

- CKODIV[2:0] bits or CKOSEL[2:0] bits in CKOCR
- HOCOFRQ1[2:0] bits in OFS1

8.6 Usage Notes

8.6.1 Notes on Clock Generation Circuit

The frequency of the following clocks supplied to each module changes according to the setting of the SCKDIVCR register:

- System clock (ICLK)
- Peripheral module clocks (PCLKB and PCLKD)

Each frequency must meet the following conditions:

- Each frequency must be selected within the operation-guaranteed range of the operating frequency (f) specified in the AC characteristics. See section 35, Electrical Characteristics.
- The system clock, peripheral module clock must be set according to Table 8.2.

To ensure correct processing after the clock frequency changes, first write to the relevant Clock Control register to change the frequency, then read the value from the register, and finally perform the subsequent processing.

8.6.2 Notes on Board Design

When using a crystal resonator, place the resonator and its load capacitors as close to the XTAL and EXTAL pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in Figure 8.11 to prevent electromagnetic induction from interfering with correct oscillation. Figure 8.11 shows the case which the main clock oscillator is used.

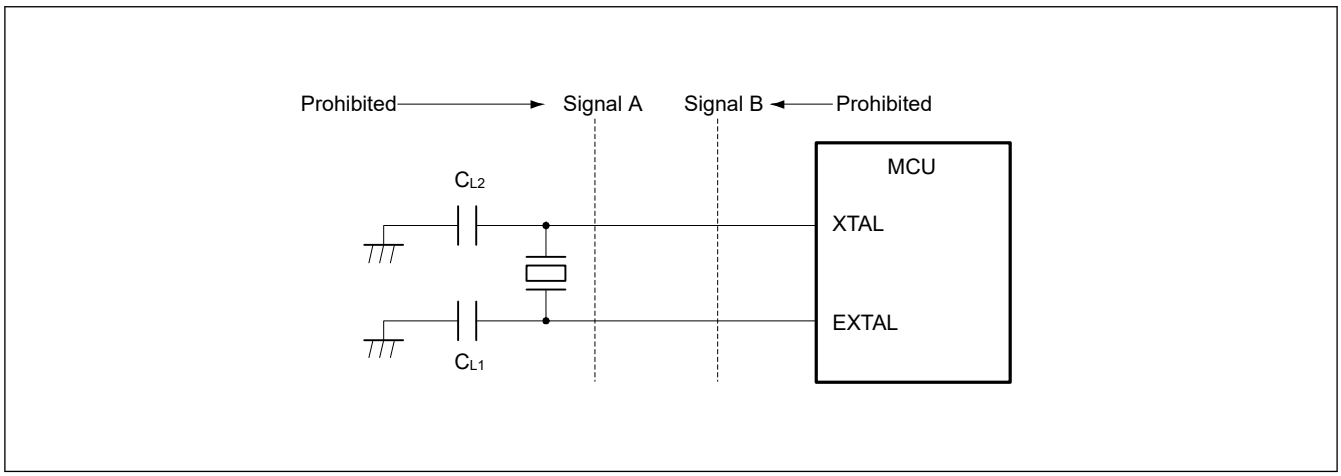


Figure 8.11 Signal routing in board design for oscillation circuit

8.6.3 Notes on Resonator Connect Pin

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports. When these pins are used as general ports, the main clock must be stopped (MOSCCR.MOSTP bit should be set to 1).

9. Clock Frequency Accuracy Measurement Circuit (CAC)

9.1 Overview

The Clock Frequency Accuracy Measurement Circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock selected as the measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range. When measurement is complete or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated.

Table 9.1 lists the CAC specifications, Figure 9.1 shows the CAC block diagram, and Table 9.2 lists the CAC I/O pin.

Table 9.1 CAC specifications

Parameter	Specifications
Measurement target clocks	Frequency can be measured for: • Main clock oscillator • HOCO clock • MOCO • LOCO clock • Peripheral module clock B (PCLKB) • IWDT-dedicated clock
Measurement reference clocks	Frequency can be referenced to: • External clock input to the CACREF pin • Main clock oscillator • HOCO clock • MOCO • LOCO clock • Peripheral module clock B (PCLKB) • IWDT-dedicated clock
Selectable function	Digital filter
Interrupt sources	• Measurement end • Frequency error • Overflow
Module-stop function	Module-stop state can be set to reduce power consumption

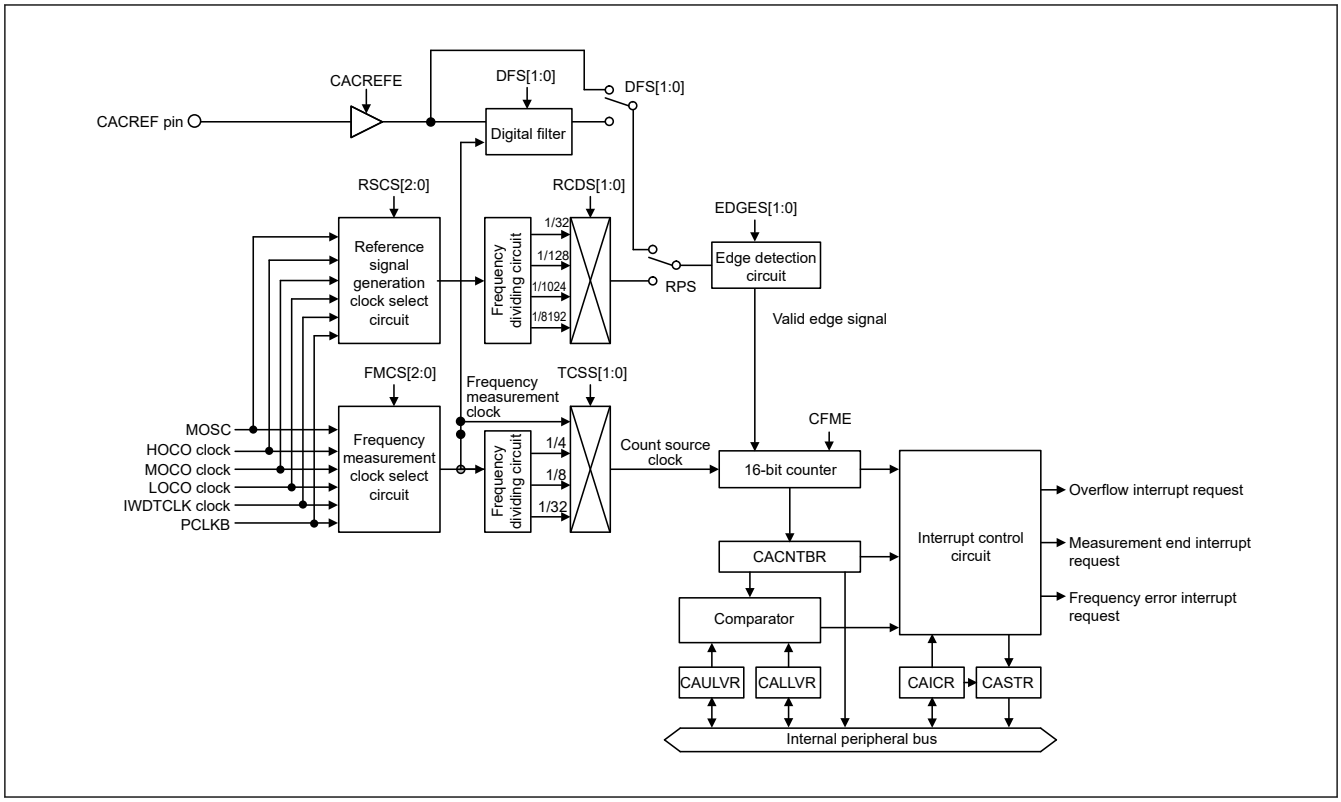


Figure 9.1 CAC block diagram

Table 9.2 CAC I/O pin

Function	Pin name	I/O	Description
CAC	CACREF	Input	Measurement reference clock input pin

9.2 Register Descriptions

9.2.1 CACR0 : CAC Control Register 0

Base address: CAC = 0x4004_4600

Offset address: 0x00

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

Bit	Symbol	Function	R/W
0	CFME	Clock Frequency Measurement Enable 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

CFME bit (Clock Frequency Measurement Enable)

The CFME bit enables clock frequency measurement. Changes made to this bit are not immediately reflected to the internal circuit. Read the bit to confirm that the change has been reflected.

9.2.2 CACR1 : CAC Control Register 1

Base address: CAC = 0x4004_4600

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	EDGES[1:0]		TCSS[1:0]		FMCS[2:0]			CACREFE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CACREFE	CACREF Pin Input Enable 0: Disable 1: Enable	R/W
3:1	FMCS[2:0]	Measurement Target Clock Select 0 0 0: Main clock oscillator 0 1 0: HOCO clock 0 1 1: MOCO clock 1 0 0: LOCO clock 1 0 1: Peripheral module clock B (PCLKB) 1 1 0: IWDT-dedicated clock Others: Setting prohibited	R/W
5:4	TCSS[1:0]	Timer Count Clock Source Select 0 0: No division 0 1: × 1/4 clock 1 0: × 1/8 clock 1 1: × 1/32 clock	R/W
7:6	EDGES[1:0]	Valid Edge Select 0 0: Rising edge 0 1: Falling edge 1 0: Both rising and falling edges 1 1: Setting prohibited	R/W

Note: Set the CACR1 register when the CACR0.CFME bit is 0.

CACREFE bit (CACREF Pin Input Enable)

The CACREFE bit enables the CACREF pin input.

FMCS[2:0] bits (Measurement Target Clock Select)

The FMCS[2:0] bits select the measurement target clock whose frequency is to be measured.

TCSS[1:0] bits (Timer Count Clock Source Select)

The TCSS[1:0] bits select the division ratio of the measurement target clock.

EDGES[1:0] bits (Valid Edge Select)

The EDGES[1:0] bits select the valid edge for the reference signal.

9.2.3 CACR2 : CAC Control Register 2

Base address: CAC = 0x4004_4600

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DFS[1:0]		RCDS[1:0]		RSCS[2:0]			RPS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RPS	Reference Signal Select 0: CACREF pin input 1: Internal clock (internally generated signal)	R/W
3:1	RSCS[2:0]	Measurement Reference Clock Select 0 0 0: Main clock oscillator 0 1 0: HOCO clock 0 1 1: MOCO clock 1 0 0: LOCO clock 1 0 1: Peripheral module clock B (PCLKB) 1 1 0: IWDT-dedicated clock Others: Setting prohibited	R/W
5:4	RCDS[1:0]	Measurement Reference Clock Frequency Division Ratio Select 0 0: × 1/32 clock 0 1: × 1/128 clock 1 0: × 1/1024 clock 1 1: × 1/8192 clock	R/W
7:6	DFS[1:0]	Digital Filter Select 0 0: Disable digital filtering 0 1: Use sampling clock for the digital filter as the frequency measuring clock 1 0: Use sampling clock for the digital filter as the frequency measuring clock divided by 4 1 1: Use sampling clock for the digital filter as the frequency measuring clock divided by 16.	R/W

Note: Set the CACR2 register when the CACR0.CFME bit is 0.

RPS bit (Reference Signal Select)

The RPS bit selects whether to use the CACREF pin input or an internal clock (internally generated signal) as the reference signal.

RSCS[2:0] bits (Measurement Reference Clock Select)

The RSCS[2:0] bits select the reference clock for measurement.

RCDS[1:0] bits (Measurement Reference Clock Frequency Division Ratio Select)

The RCDS[1:0] bits select the frequency-divisor of the reference clock for measurement when an internal reference clock is selected. When RPS = 0 (CACREF pin is used as the reference clock source), the reference clock is not divided.

DFS[1:0] bits (Digital Filter Select)

The DFS[1:0] bits enable or disable the digital filter and selects its sampling clock.

9.2.4 CAICR : CAC Interrupt Control Register

Base address: CAC = 0x4004_4600

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	OVFFCL	MENDFCL	FERRFCL	—	OVFIE	MENDIE	FERRIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FERRIE	Frequency Error Interrupt Request Enable 0: Disable 1: Enable	R/W
1	MENDIE	Measurement End Interrupt Request Enable 0: Disable 1: Enable	R/W

Bit	Symbol	Function	R/W
2	OVFIE	Overflow Interrupt Request Enable 0: Disable 1: Enable	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	FERRFCL	FERRF Clear 0: No effect 1: The CASTR.FERRF flag is cleared	W
5	MENDFCL	MENDF Clear 0: No effect 1: The CASTR.MENDF flag is cleared	W
6	OVFFCL	OVFF Clear 0: No effect 1: The CASTR.OVFF flag is cleared.	W
7	—	This bit is read as 0. The write value should be 0.	R/W

FERRIE bit (Frequency Error Interrupt Request Enable)

The FERRIE bit enables or disables the frequency error interrupt request.

MENDIE bit (Measurement End Interrupt Request Enable)

The MENDIE bit enables or disables the measurement end interrupt request.

OVFIE bit (Overflow Interrupt Request Enable)

The OVFIE bit enables or disables the overflow interrupt request.

FERRFCL bit (FERRF Clear)

Setting the FERRFCL bit to 1 clears the CASTR.FERRF flag.

MENDFCL bit (MENDF Clear)

Setting the MENDFCL bit to 1 clears the CASTR.MENDF flag.

OVFFCL bit (OVFF Clear)

Setting the OVFFCL bit to 1 clears the CASTR.OVFF flag.

9.2.5 CASTR : CAC Status Register

Base address: CAC = 0x4004_4600

Offset address: 0x04

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

Bit	Symbol	Function	R/W
0	FERRF	Frequency Error Flag 0: Clock frequency is within the allowable range 1: Clock frequency has deviated beyond the allowable range (frequency error).	R
1	MENDF	Measurement End Flag 0: Measurement is in progress 1: Measurement ended	R
2	OVFF	Overflow Flag 0: Counter has not overflowed 1: Counter overflowed	R
7:3	—	These bits are read as 0.	R

FERRF flag (Frequency Error Flag)

The FERRF flag indicates a deviation of the clock frequency from the set value (frequency error).

[Setting condition]

- The clock frequency is outside the allowable range defined in the CAULVR and CALLVR registers.

[Clearing condition]

- 1 is written to the FERRFCL bit.

MENDF flag (Measurement End Flag)

The MENDF flag indicates the end of measurement.

[Setting condition]

- Measurement ends.

[Clearing condition]

- 1 is written to the MENDFCL bit.

OVFF flag (Overflow Flag)

The OVFF flag indicates that the counter overflowed.

[Setting condition]

- The counter overflows.

[Clearing condition]

- 1 is written to the CAICR.OVFFCL bit.

9.2.6 CAULVR : CAC Upper-Limit Value Setting Register

Base address: CAC = 0x4004_4600

Offset address: 0x06

Bit position:	15															0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	The Upper Value of the Allowable Range The CAULVR register is a 16-bit read/write register that specifies the upper value of the allowable range. When the counter value exceeds the value specified in this register, a frequency error is detected. Write to this register when the CACR0.CFME bit is 0. The counter value stored in CACNTBR can vary depending on the difference between the phases of the digital filter and edge-detection circuit, and the signal on the CACREF pin. Ensure that this setting allows an adequate margin.	R/W

9.2.7 CALLVR : CAC Lower-Limit Value Setting Register

Base address: CAC = 0x4004_4600

Offset address: 0x08

Bit position:	15															0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	The Lower Value of the Allowable Range The CALLVR register is a 16-bit read/write register that specifies the lower value of the allowable range. When the counter value falls below the value specified in this register, a frequency error is detected. Write to this register when the CACR0.CFME bit is 0. The counter value stored in CACNTBR can vary depending on the difference between the phases of the digital filter and edge-detection circuit, and the signal on the CACREF pin. Ensure that this setting allows an adequate margin.	R/W

9.2.8 CACNTBR : CAC Counter Buffer Register

Base address: CAC = 0x4004_4600

Offset address: 0x0A

Bit position:	15															0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	The Measurement Result The CACNTBR register is a 16-bit read-only register that stores the measurement result.	R

9.3 Operation

9.3.1 Measuring Clock Frequency

The CAC measures the clock frequency using the CACREF pin input or an internal clock as a reference. Figure 9.2 shows an operating example of the CAC.

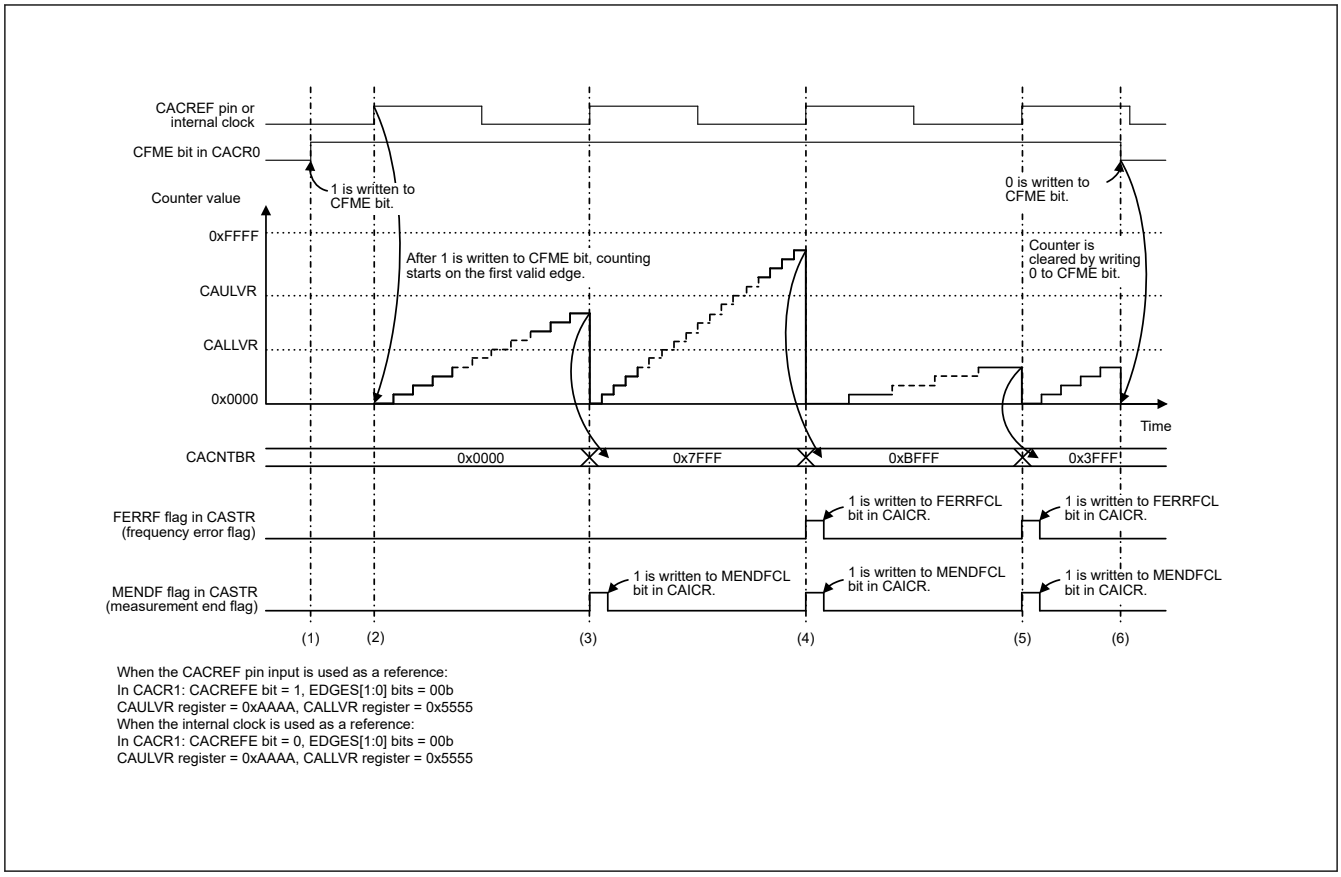


Figure 9.2 CAC operating example

The events in Figure 9.2 are:

1. When the CACREF pin input is used as reference (CACR1.CACREFE = 1), frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is set to 0 and the CACR1.CACREFE bit is set to 1. When the internal clock is used as reference (CACR1.CACREFE = 0), frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is set to 1.
2. When the CACREF pin input is used as reference, after 1 is written to the CFME bit, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits (rising edge (CACR1.EDGES[1:0] = 00b) in Figure 9.2) is input from the CACREF pin. When the internal clock is used as reference, after 1 is written to the CFME bit, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits (rising edge (CACR1.EDGES[1:0] = 00b) in Figure 9.2) is input based on the clock source selected by the CACR2.RSCS[2:0] bits.
3. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If both CACNTBR ≤ CAULVR and CACNTBR ≥ CALLVR are true, only the MENDF flag in CASTR is set to 1, because the clock frequency is correct. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
4. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If CACNTBR > CAULVR, the FERRF flag in CASTR is set to 1, because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. The MENDF flag in CASTR is set to 1 at the end of measurement. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
5. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If CACNTBR < CALLVR, the FERRF flag in CASTR is set to 1, because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. The MENDF flag in CASTR is set to 1 at the end of measurement. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
6. When the CFME bit in CACR0 is 1, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR every time a valid edge is input. Writing 0 to the CFME bit in CACR0 clears the counter and stops up-counting.

9.3.2 Digital Filtering of Signals on CACREF Pin

The CACREF pin has a digital filter, and levels on the CACREF pin are transmitted to the internal circuitry after three consecutive matches in the selected sampling interval. The same level continues to be transmitted internally until the level on the pin has three consecutive matches again. Enabling or disabling of the digital filter and its sampling clock are selectable.

The counter value transferred to CACNTBR might be in error by up to 1 cycle of the sampling clock because of the difference between the phases of the digital filter and the signal input to the CACREF pin. When a frequency dividing clock is selected as a count source clock, the counter value error is obtained using the following formula:

$$\text{Counter value error} = (1 \text{ cycle of the count source clock}) / (1 \text{ cycle of the sampling clock})$$

9.4 Interrupt Requests

The CAC generates three types of interrupt requests:

- Frequency error interrupt
- Measurement end interrupt
- Overflow interrupt

When an interrupt source is generated, the associated status flag is set to 1. Table 9.3 provides information on the CAC interrupt requests.

Table 9.3 CAC interrupt requests

Interrupt request	Interrupt enable bit	Status flag	Interrupt sources
Frequency error interrupt	CAICR.FERRIE	CASTR.FERRF	The result of comparing CACNTBR with CAULVR and CALLVR is either CACNTBR > CAULVR or CACNTBR < CALLVR
Measurement end interrupt	CAICR.MENDIE	CASTR.MENDF	• Valid edge is input from the CACREF pin or internal clock • Measurement end interrupt does not occur at the first valid edge after writing 1 to the CACR0.CFME bit
Overflow interrupt	CAICR.OVFIE	CASTR.OVFF	Counter overflows

9.5 Usage Notes

9.5.1 Settings for the Module-Stop Function

The Module Stop Control Register C (MSTPCRC) can enable or disable CAC operation. The CAC module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

10. Low Power Modes

10.1 Overview

The MCU provides several functions for reducing power consumption, such as setting clock dividers, stopping modules, selecting power control mode in normal mode, and transitioning to low power modes.

Table 10.1 lists the specifications of the low power mode functions. Table 10.2 lists the conditions to transition to low power modes, the states of the CPU and peripheral modules, and the method for canceling each mode. After a reset, the MCU enters the program execution state, but only the DTC and SRAM operate.

Table 10.1 Specifications of the low power mode functions

Item	Specification
Reducing power consumption by switching clock signals	The frequency division ratio can be selected independently for the system clock (ICLK), peripheral module clock (PCLKB and PCLKD)*1
Module stop	Functions can be stopped independently for each peripheral module
Low power modes	• Sleep mode • Software Standby mode • Snooze mode
Power control modes	Power consumption can be reduced in Normal, Sleep, and Snooze mode by selecting an appropriate operating power control mode according to the operating frequency and voltage. Four operating power control modes are available: • High-speed mode • Middle-speed mode • Low-speed mode • Subosc-speed mode

Note 1. For details, see section 8, Clock Generation Circuit

Table 10.2 Operating conditions of each low power mode (1 of 2)

Item	Sleep mode	Software Standby mode	Snooze mode*1
Transition condition	When [Condition 1] or [Condition 2] while SBYCR.SSBY = 0 [Condition 1] • WFI instruction • A valid interrupt request*2 (including a transition from the time WFI instruction is executed to the time the transition to Sleep mode is completed) [Condition 2] • SCR.SLEEPONEXIT = 1 • Complete execution of all exception handlers • A valid interrupt request*2 (including a transition from the time WFI instruction is executed to the time the transition to Sleep mode is completed)	When [Condition 1] or [Condition 2] while SBYCR.SSBY = 1 [Condition 1] • WFI instruction • A valid interrupt request*2 (including a transition from the time WFI instruction is executed to the time the transition to Software Standby mode is completed) [Condition 2] • SCR.SLEEPONEXIT = 1 • Complete execution of all exception handlers • A valid interrupt request*2 (including a transition from the time WFI instruction is executed to the time the transition to Software Standby mode is completed)	Snooze request in Software Standby mode. SNZCR.SNZE = 1
Canceling method	All interrupts. Any reset available in the mode.	Interrupts shown in Table 10.3. Any reset available in the mode.	Interrupts shown in Table 10.3. Any reset available in the mode.
State after cancellation by an interrupt	Program execution state	Program execution state	Program execution state
State after cancellation by a reset	Reset state	Reset state	Reset state
Main clock oscillator	Selectable	Stop	Selectable*3
High-speed on-chip oscillator	Selectable	Stop	Selectable

Table 10.2 Operating conditions of each low power mode (2 of 2)

Item	Sleep mode	Software Standby mode	Snooze mode*1
Middle-speed on-chip oscillator	Selectable	Stop	Selectable
Low-speed on-chip oscillator	Selectable	Selectable	Selectable
IWDT-dedicated on-chip oscillator	Selectable*5	Selectable*5	Selectable*5
Oscillation stop detection function	Selectable	Operation prohibited	Operation prohibited
Clock/buzzer output function	Selectable	Selectable*4	Selectable
CPU	Stop (Retained)	Stop (Retained)	Stop (Retained)
SRAM	Selectable	Stop (Retained)	Selectable
Flash memory	Operating	Stop (Retained)	Stop (Retained)
Data Transfer Controller (DTC)	Selectable	Stop (Retained)	Selectable
Watchdog Timer (WDT)	Selectable*5	Stop (Retained)	Stop (Retained)
Independent Watchdog Timer (IWDT)	Selectable*5	Selectable*5	Selectable*5
Low Power Asynchronous General Purpose Timer (AGTWn, n = 0, 1)	Selectable	Selectable*6	Selectable*6
12-bit A/D Converter (ADC12)	Selectable	Stop (Retained)	Selectable*10
Data Operation Circuit (DOC)	Selectable	Stop (Retained)	Selectable
Serial Communications Interface (SCI0)	Selectable	Stop (Retained)	Selectable*8
Serial Communications Interface (SCIn, n = 1, 2, 9)	Selectable	Stop (Retained)	Operation prohibited
I^2C Bus Interface (IIC0)	Selectable	Selectable*9	Selectable*9
Event Link Controller (ELC)	Selectable	Stop (Retained)	Selectable*7
High-speed Analog Comparator (ACMPHS0/1)	Selectable	Selectable*11	Selectable*11
NMI, IRQn (n = 0 to 7) pin interrupt	Selectable	Selectable	Selectable
Key Interrupt Function (KINT)	Selectable	Selectable	Selectable
Low Voltage Detection (LVD)	Selectable	Selectable	Selectable
Power-on reset circuit	Operating	Operating	Operating
Other peripheral modules	Selectable	Stop (Retained)	Operation prohibited
I/O ports	Operating	Retained	Operating

Note: Selectable means that operating or not operating can be selected by the control registers.
Stop (Retained) means that the contents of the internal registers are retained but the operations are suspended.
Operation prohibited means that the function must be stopped before entering Software Standby mode.
Otherwise, proper operation is not guaranteed in Snooze mode.

Note 1. All modules whose module-stop bits are 0 start as soon as PCLKs are supplied after entering Snooze mode.
To avoid an increasing power consumption in Snooze mode, set the module-stop bit of modules that are not required in Snooze mode to 1 before entering Software Standby mode.

Note 2. Valid interrupt requests are any interrupt/exception that are not masked by the priority level of current exception and the priority level set by BASEPRI. In addition, if the interrupt request is based on IELSRn, the interrupt must be enabled by NVIC_ISERn.

Note 3. When using SCI0 in Snooze mode, the MOSCCR.MOSTP bit must be 1.

Note 4. Stopped when the Clock Output Source Select bits (CKOCR.CKOSEL[2:0]) are set to a value other than 010b (LOCO).

Note 5. In IWDT-dedicated on-chip oscillator and IWDT, operating or stopping is selected by setting the IWDT Stop Control bit (IWDTSTPCTL) in Option Function Select register 0 (OFS0) in IWDT auto start mode. In WDT, operating or stopping is selected by setting the WDT Stop Control bit (WDTSTPCTL) in Option Function Select register 0 (OFS0) in WDT auto start mode or by setting the WDTCSTPR.SLCSTP in WDT register-start mode.

Note 6. AGTW0 operation is possible when 100b (LOCO) is selected in the AGTW0.AGTMR1.TCK[2:0] bits.
AGTW1 operation is possible when 100b (LOCO) or 101 (underflow event signal from AGTW0) is selected in the AGTW1.AGTMR1.TCK[2:0] bits.

Note 7. Event lists the restrictions described in section 10.9.12. ELC Events in Snooze Mode.

Note 8. Serial communication modes of SCI0 is only in asynchronous mode.

Note 9. Only wakeup interrupt is available.

Note 10. When using the 12-bit A/D Converter (ADC12) in Snooze mode, the ADCMPCR.CMPAE and ADCMPCR.CMPBE bit must be 1.

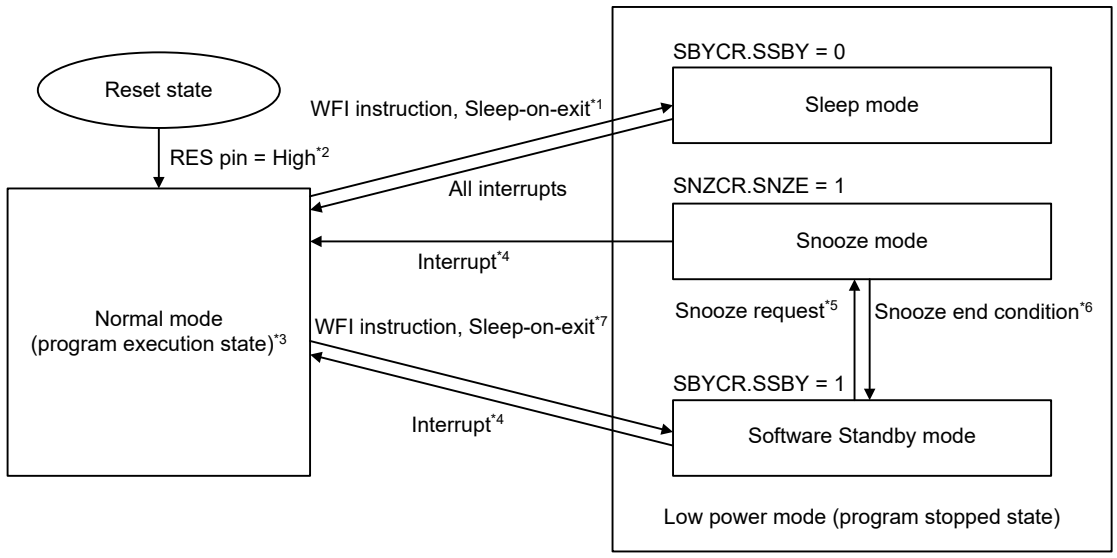
Note 11. Only VCOUT function is permitted. The VCOUT pin operates when ACMPHS uses no digital filter. For details on digital filter, see section 28, High-Speed Analog Comparator (ACMPHS).

Table 10.3 Available interrupt sources to transition to Normal mode from Snooze mode and Software Standby mode

Interrupt source	Name	Software Standby mode	Snooze mode
NMI		Yes	Yes
Port	PORT_IRQn (n = 0 to 7)	Yes	Yes
LVD	LVD_LVD1	Yes	Yes
	LVD_LVD2	Yes	Yes
IWDT	IWDT_NMIUNDF	Yes	Yes
KINT	KEY_INTKR	Yes	Yes
AGTW1	AGT1_AGTI	Yes	Yes*3
	AGT1_AGTCMAI	Yes	Yes
	AGT1_AGTCMBI	Yes	Yes
ACMPHS	ACMP_HS0	Yes	Yes
IIC0	IIC0_WUI	Yes	Yes
ADC120	ADC120_WCMPM	No	Yes with SELSR0*1 *3
	ADC120_WCMPUM	No	Yes with SELSR0*1 *3
SCI0	SCI0_AM	No	Yes with SELSR0*1 *2
	SCI0_RXI_OR_ERI	No	Yes with SELSR0*1 *2
DTC	DTC_COMPLETE	No	Yes with SELSR0*1
DOC	DOC_DOPCI	No	Yes with SELSR0*1

Note 1. To use the interrupt request as a trigger for exiting Snooze mode, the request must be selected in SELSR0. See section 12, Interrupt Controller Unit (ICU). When a trigger selected in SELSR0 occurs after executing a WFI instruction, and during the transition from Normal mode to Software Standby mode, the request can be accepted depending on the timing of the occurrence.
Note 2. Only one of either SCI0_AM or SCI0_RXI_OR_ERI can be selected.
Note 3. Event that is enabled by the SNZEDCR0 register must not be used.

Figure 10.1 shows the transition between Normal mode to low power mode.



Note 1. When an interrupt that acts as a trigger for cancellation of the Sleep mode state is received during a transition to the Sleep mode state, the MCU does not transit to Sleep mode state but got back to Normal mode state.
Note 2. The MOCO is the source of the operating clock following a transition from the reset state to Normal mode.
Note 3. The transition to Normal mode is made from an interrupt in Sleep mode, Software Standby mode, or Snooze mode. The clock source is the same as before entering the low power mode.
Note 4. See Table 10.3.
Note 5. See section 10.5.1. Setting Operating Power Control Mode.
Note 6. See section 10.8.3. Returning from Snooze Mode to Software Standby Mode.
Note 7. When an interrupt that acts as a trigger for cancellation of the Sleep mode state is received during a transition to the Software Standby mode state, the MCU does not transit to Software Standby mode state but go back to Normal mode state.

Figure 10.1 Low power mode transitions

10.2 Register Descriptions

10.2.1 SBYCR : Standby Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x00C

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

Bit	Symbol	Function	R/W
14:0	—	These bits are read as reset value. The write value should be reset value	R/W
15	SSBY	Software Standby Mode Select 0: Sleep mode 1: Software Standby mode.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

SSBY bit (Software Standby Mode Select)

The SSBY bit specifies the transition destination after a WFI instruction is executed.

When the SSBY bit is set to 1, the MCU enters Software Standby mode after execution of a WFI instruction. When the MCU returns to Normal mode from Software Standby mode by an interrupt, the SSBY bit remains 1. The SSBY bit can be cleared by writing 0 to it.

While the OSTDCR.OSTDE bit is 1, setting of the SSBY bit is ignored. Even if SSBY bit is 1, the MCU enters Sleep mode on execution of a WFI instruction.

While the FENTRYR.FENTRY0 bit is 1 setting of the SSBY bit is ignored. Even if SSBY bit is 1, the MCU enters Sleep mode on execution of a WFI instruction.

10.2.2 MSTPCRA : Module Stop Control Register A

Base address: SYSC = 0x4001_E000

Offset address: 0x01C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	MSTPA22	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1

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

Bit	Symbol	Function	R/W
21:0	—	These bits are read as 1. The write value should be 1.	R/W
22	MSTPA22	DTC Module Stop*1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31:23	—	These bits are read as 1. The write value should be 1.	R/W

Note 1. When rewriting the MSTPA22 bit from 0 to 1, disable the DTC before setting the MSTPA22 bit.

10.2.3 MSTPCRB : Module Stop Control Register B

Base address: MSTP = 0x4004_7000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTPB31	MSTPB30	MSTPB29	—	—	—	—	—	—	MSTPB22	—	—	MSTPB19	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	MSTPB9	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
8:0	—	These bits are read as 1. The write value should be 1.	R/W
9	MSTPB9	I^2C Bus Interface 0 Module Stop Target module: IIC0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
18:10	—	These bits are read as 1. The write value should be 1.	R/W
19	MSTPB19	Serial Peripheral Interface 0 Module Stop Target module: SPI0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21:20	—	These bits are read as 1. The write value should be 1.	R/W

Bit	Symbol	Function	R/W
22	MSTPB22	Serial Communication Interface 9 Module Stop Target module: SCI9 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
28:23	—	These bits are read as 1. The write value should be 1.	R/W
29	MSTPB29	Serial Communication Interface 2 Module Stop Target module: SCI2 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
30	MSTPB30	Serial Communication Interface 1 Module Stop Target module: SCI1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31	MSTPB31	Serial Communication Interface 0 Module Stop Target module: SCI0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

10.2.4 MSTPCRC : Module Stop Control Register C

Base address: MSTP = 0x4004_7000

Offset address: 0x004

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MSTPC14	MSTPC13	—	—	—	—	—	—	—	—	—	—	—	MSTPC1	MSTPC0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	MSTPC0	Clock Frequency Accuracy Measurement Circuit Module Stop*1 Target module: CAC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
1	MSTPC1	Cyclic Redundancy Check Calculator Module Stop Target module: CRC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
12:2	—	These bits are read as 1. The write value should be 1.	R/W
13	MSTPC13	Data Operation Circuit Module Stop Target module: DOC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
14	MSTPC14	Event Link Controller Module Stop Target module: ELC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31:15	—	These bits are read as 1. The write value should be 1.	R/W

Note 1. The MSTPC0 bit must be written while the oscillation of the clock to be controlled by this bit is stable. To enter Software Standby mode after writing this bit, wait for 2 cycles of the slowest clock from the clocks output by the oscillators, then execute a WFI instruction.

10.2.5 MSTPCRD : Module Stop Control Register D

Base address: MSTP = 0x4004_7000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	MSTPD28	—	—	—	—	—	—	—	—	—	—	—	MSTPD16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MSTPD14	MSTPD13	—	—	—	—	—	—	MSTPD6	—	—	MSTPD3	MSTPD2	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 1. The write value should be 1.	R/W
2	MSTPD2	Low Power Asynchronous General Purpose Timer 1 Module Stop*1 Target module: AGTW1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
3	MSTPD3	Low Power Asynchronous General Purpose Timer 0 Module Stop*2 Target module: AGTW0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
5:4	—	These bits are read as 1. The write value should be 1.	R/W
6	MSTPD6	General PWM Timer 160 to 163 and PWM Delay Generation Circuit Module Stop Target module: GPT160 to 163, and PWM delay generation circuit 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
12:7	—	These bits are read as 1. The write value should be 1.	R/W
13	MSTPD13	Port Output Enable for GPT 1 Module Stop Target module: POEG1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
14	MSTPD14	Port Output Enable for GPT 0 Module Stop Target module: POEG0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
15	—	This bit is read as 1. The write value should be 1.	R/W
16	MSTPD16	12-bit A/D Converter Module Stop Target module: ADC120 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
27:17	—	These bits are read as 1. The write value should be 1.	R/W
28	MSTPD28	High-Speed Analog Comparator Module Stop Target module: ACMPHS 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31:29	—	These bits are read as 1. The write value should be 1.	R/W

Note 1. When the count source is LOCO, AGTW1 counting does not stop even if MSTPD2 is set to 1. If the count source is LOCO, this bit must be set to 1 except when accessing the AGTW1 registers.

Note 2. When the count source is LOCO, AGTW0 counting does not stop even if MSTPD3 is set to 1. If the count source is LOCO, this bit must be set to 1 except when accessing the AGTW0 registers.

10.2.6 OPCCR : Operating Power Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0A0

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

Bit	Symbol	Function	R/W
1:0	OPCM[1:0]	Operating Power Control Mode Select 0 0: High-speed mode 0 1: Middle-speed mode 1 0: Setting prohibited 1 1: Low-speed mode	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	OPCMTSF	Operating Power Control Mode Transition Status Flag 0: Transition completed 1: During transition	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The OPCCR register is used to reduce power consumption in Normal mode, Sleep mode, and Snooze mode. Power consumption can be reduced according to the operating frequency and operating voltage used by the OPCCR setting. For the procedure to change the operating power control modes, see section 10.5. Function for Lower Operating Power Consumption.

OPCM[1:0] bits (Operating Power Control Mode Select)

The OPCM[1:0] bits select the operating power control mode in Normal mode, Sleep mode, and Snooze mode.

Table 10.4 shows the relationship between the operating power control modes, the OPCM[1:0], and SOPCM bits settings.

Writing to OPCCR.OPCM[1:0] is prohibited while MCU is under the following conditions:

1. HOCOCR.HCSTP and OSCSF.HOCOSF are 0 (the oscillation of the HOCO clock is not yet stable).
2. The MCU is in Sleep or Snooze mode, the MCU transitions to Normal mode from Sleep or Snooze mode, the MCU transitions to Sleep, Snooze, or Software Standby mode from Normal mode, or the MCU is in transfer state when in operating power mode.
3. Flash is in programming mode.
4. The MCU is in Subosc-speed mode (SOPCCR.SOPCM bit is 1).

OPCMTSF flag (Operating Power Control Mode Transition Status Flag)

The OPCMTSF flag indicates the switching control state when the operating power control mode is switched. This flag becomes 1 when the OPCM bit is written, and 0 when mode transition completes. Read this flag and confirm that it is 0 before proceeding.

10.2.7 SOPCCR : Sub Operating Power Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x0AA

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

Bit	Symbol	Function	R/W
0	SOPCM	Sub Operating Power Control Mode Select 0: Other than Subosc-speed mode 1: Subosc-speed mode	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	SOPCMTSF	Operating Power Control Mode Transition Status Flag 0: Transition completed 1: During transition	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The SOPCCR register is used to reduce power consumption in Normal mode, Sleep mode, and Snooze mode. Setting this register initiates entry to and exit from Subosc-speed mode. Subosc-speed mode is available only when using LOCO without dividing the frequency.

For the procedure to change operating power control modes, see section 10.5. Function for Lower Operating Power Consumption.

SOPCM bit (Sub Operating Power Control Mode Select)

The SOPCM bit selects the operating power control mode in Normal mode, Sleep mode, and Snooze mode. Setting this bit to 1 allows transition to Subosc-speed mode. Setting this bit to 0 allows a return to the operating mode (set in OPCCR.OPCM[1:0]) before the transition to Subosc-speed mode.

Writing to SOPCCR.SOPCM is prohibited while MCU is under the following conditions:

1. The MCU is in Sleep or Snooze mode, the MCU transitions to Normal mode from Sleep, Snooze, or Software Standby mode, the MCU transitions to Sleep, Snooze, or Software Standby mode from Normal mode, or the MCU is in transfer state when in operating power mode.
2. Flash is in programming mode.
3. MOSC is operating (MOSCCR.MOSTP bit is 0), HOCO is operating (HOCOCR.HCSTP bit is 0), or MOCO is operating (MOCOCR.MCSTP bit is 0).
4. The value of SCKDIVCR register is not equal to 0x00000000.
5. The data flash is disabled (DFLCTL.DFLEN bit is 0).

Table 10.4 shows the relationship between the operating power control modes, the OPCM[1:0], and SOPCM bits settings.

SOPCMTSF flag (Operating Power Control Mode Transition Status Flag)

The SOPCMTSF flag indicates the switching control state when the operating power control mode is switched to or from Subosc-speed mode. This flag becomes 1 when the SOPCM bit is written, and 0 when mode transition completes. Read this flag and confirm that it is 0 before proceeding.

Table 10.4 shows each operating power control mode.

Table 10.4 Operating power control mode

Operating power control mode	OPCM[1:0] bits	SOPCM bit	Power consumption
High-speed mode	00b	0	High
Middle-speed mode	01b	0	↓
Low-speed mode	11b	0	↓
Subosc-speed mode	xxb	1	Low

10.2.8 SNZCR : Snooze Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x092

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

Bit	Symbol	Function	R/W
0	RXDREQEN	RXD0 Snooze Request Enable 0: Ignore RXD0 falling edge in Software Standby mode 1: Detect RXD0 falling edge in Software Standby mode	R/W
1	SNZDTCEN	DTC Enable in Snooze mode 0: Disable DTC operation 1: Enable DTC operation	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	SNZE	Snooze mode Enable 0: Disable Snooze mode 1: Enable Snooze mode	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

RXDREQEN bit (RXD0 Snooze Request Enable)

The RXDREQEN bit specifies whether to detect a falling edge of the RXD0 pin in Software Standby mode. This bit can be used only when SCI0 is operating in asynchronous mode. To detect a falling edge of the RXD0 pin, set this bit before entering Software Standby mode. When this bit is set to 1, a falling edge of the RXD0 pin in Software Standby mode causes the MCU to enter Snooze mode.

SNZDTCEN bit (DTC Enable in Snooze mode)

The SNZDTCEN bit specifies whether to use the DTC and SRAM in Snooze mode. To use the DTC and SRAM in Snooze mode, set this bit to 1 before entering Software Standby mode. When this bit is set to 1, the DTC can be activated by setting IELSRn register.

SNZE bit (Snooze mode Enable)

The SNZE bit specifies whether to enable a transition from Software Standby mode to Snooze mode. To use Snooze mode, set this bit to 1 before entering Software Standby mode. When this bit is set to 1, a trigger as shown in Table 10.6 in Software Standby mode causes the MCU to enter Snooze mode. After the MCU transitions from Software Standby mode or Snooze mode to Normal mode, set 0 to the SNZE bit once then set it before re-entering Software Standby mode. For details, see section 10.8. Snooze Mode.

10.2.9 SNZEDCR0 : Snooze End Control Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x094

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SCI0UMTED	—	—	AD0UMTED	AD0MATED	DTCNZRED	DTCZRED	AGTUNFED
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AGTUNFED	AGTW1 Underflow Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W

Bit	Symbol	Function	R/W
1	DTCZRED	Last DTC Transmission Completion Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
2	DTCNZRED	Not Last DTC Transmission Completion Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
3	AD0MATED	ADC12 Compare Match Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
4	AD0UMTED	ADC12 Compare Mismatch Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	SCI0UMTED	SCI0 Address Mismatch Snooze End Enable 0: Disable the snooze end request 1: Enable the snooze end request	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The SNZEDCR0 register controls the condition of switching from Snooze mode to Software Standby mode. In order to use a trigger shown in Table 10.7 as a condition to switch from Snooze mode to Software Standby mode, the corresponding bit in the SNZEDCR0 register must be set to 1.

The event that is used to return from snooze mode to normal mode as shown in Table 10.3 must not be enabled in the SNZEDCR0 register.

AGTUNFED bit (AGTW1 Underflow Snooze End Enable)

The AGTUNFED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on an AGTW1 underflow. For details on the trigger conditions, see section 21, Low Power Asynchronous General Purpose Timer (AGTW).

DTCZRED bit (Last DTC Transmission Completion Snooze End Enable)

The DTCZRED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on completion of the last DTC transmission, that is, when CRA or CRB registers in the DTC is 0. For details on the trigger conditions, see section 15, Data Transfer Controller (DTC).

DTCNZRED bit (Not Last DTC Transmission Completion Snooze End Enable)

The DTCNZRED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on completion of each DTC transmission, that is, when CRA or CRB registers in the DTC is not 0. For details on the trigger conditions, see section 15, Data Transfer Controller (DTC).

AD0MATED bit (ADC12 Compare Match Snooze End Enable)

The AD0MATED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on an ADC120 event when a conversion result matches the expected data. For details on the trigger conditions, see section 29, 12-Bit A/D Converter (ADC12).

AD0UMTED bit (ADC12 Compare Mismatch Snooze End Enable)

The AD0UMTED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on an ADC120 event when the conversion result does not match the expected data. For details on the trigger conditions, see section 29, 12-Bit A/D Converter (ADC12).

SCI0UMTED bit (SCI0 Address Mismatch Snooze End Enable)

The SCI0UMTED bit specifies whether to enable a transition from Snooze mode to Software Standby mode on an SCI0 event when an address received in Software Standby mode does not match the expected data. For details on the trigger conditions, see section 24, Serial Communications Interface (SCI). Only set this bit to 1 when SCI0 operates in asynchronous mode.

10.2.10 SNZREQCR0 : Snooze Request Control Register 0

Base address: SYSC = 0x4001_E000

Offset address: 0x098

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	SNZREQEN30	SNZREQEN29	SNZREQEN28	—	—	—	—	SNZREQEN23	—	—	—	—	—	SNZREQEN17	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SNZREQEN7	SNZREQEN6	SNZREQEN5	SNZREQEN4	SNZREQEN3	SNZREQEN2	SNZREQEN1	SNZREQEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SNZREQEN0	Enable IRQ0 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
1	SNZREQEN1	Enable IRQ1 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
2	SNZREQEN2	Enable IRQ2 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
3	SNZREQEN3	Enable IRQ3 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
4	SNZREQEN4	Enable IRQ4 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
5	SNZREQEN5	Enable IRQ5 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
6	SNZREQEN6	Enable IRQ6 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
7	SNZREQEN7	Enable IRQ7 pin snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
16:8	—	These bits are read as 0. The write value should be 0.	R/W
17	SNZREQEN17	Enable KEY_INTKR snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
22:18	—	These bits are read as 0. The write value should be 0.	R/W
23	SNZREQEN23	Enable ACMPHS snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
27:24	—	These bits are read as 0. The write value should be 0.	R/W
28	SNZREQEN28	Enable AGTW1 underflow snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W

Bit	Symbol	Function	R/W
29	SNZREQEN29	Enable AGTW1 compare match A snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
30	SNZREQEN30	Enable AGTW1 compare match B snooze request 0: Disable the snooze request 1: Enable the snooze request	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The SNZREQCR0 register controls which trigger causes the MCU to switch from Software Standby mode to Snooze mode. If a trigger is selected as a request to cancel Software Standby mode by setting the WUPEN register, see section 12, Interrupt Controller Unit (ICU), the MCU enters Normal mode when the trigger is generated while the associated bit of the SNZREQCR0 is 1. The setting of the WUPEN register always has higher priority than the setting of the SNZREQCR0 register. For details, see section 10.8. Snooze Mode and section 12, Interrupt Controller Unit (ICU).

10.2.11 SYOCDCR : System Control OCD Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x040E

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

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	DBGEN	Debugger Enable bit Set to 1 first in on-chip debug mode. 0: On-chip debugger is disabled 1: On-chip debugger is enabled	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DBGEN bit (Debugger Enable bit)

The DBGEN bit enables the on-chip debug mode. This bit must be set to 1 first in the on-chip debugger mode.

[Setting condition]

- Writing 1 to the bit when the debugger is connected.

[Clearing condition]

- Power-on reset is generated
- Writing 0 to the bit.

Note: Certain restrictions apply in terms of the MCU states in which the DBGEN bit can be set to 1. For details, see section 2.7.3. Restrictions on Connecting an OCD emulator.

10.2.12 LSMRWDIS : Low Speed Module R/W Disable Control Register

Base address: MSTP = 0x4004_7000

Offset address: 0x00C

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	WDTDIS	WDT Operation Clock and R/W Disable Control Stop the WDT counter clock and register R/W clock (valid only when LPOPT.LPOPTEN = 1) 0: WDT operates as normal 1: Stop the WDT clock and register R/W clock	R/W
2	IWDTDIS	IWDT Register R/W Disable Control Stop the IWDT register R/W clock (valid only when LPOPT.LPOPTEN = 1) 0: IWDT register R/W clock always on 1: Stop the IWDT register R/W clock	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	WREN	Write Enable for bits [2:0] 0: Write protect for bits [2:0] 1: Write enable for bits [2:0]	R/W
15:8	PRKEY[7:0]	LSMRWDIS Key Code These bits control the write access to the LSMRWDIS register. To modify the LSMRWDIS register, write 0xA5 to the upper 8 bits and the target value to the lower 8 bits as a 16-bit unit.	W

WDTDIS bit (WDT Operation Clock and R/W Disable Control)

[Setting condition]

- This bit can only be modified when WREN is set to 1.
- When LPOPT.LPOPTEN = 1 and this bit is set to 1, this bit stops the WDT operate clock.
- Do not set this bit to 1 when WDT is in auto start mode (OFS0.WDTSTRT = 0).
- Do not set this bit to 1 when WDT is operating.
- Set this bit to 1 to disable register start mode for WDT.

IWDTDIS bit (IWDT Register R/W Disable Control)

[Setting condition]

- This bit can only be modified when WREN is set to 1.
- When LPOPT.LPOPTEN = 1 and this bit is set to 1, this bit stops the IWDT register R/W clock.
- Do not set this bit to 1 when IWDT is in auto start mode (OFS0.IWDTSTRT = 0).
- Do not set this bit to 1 when IWDT is operating.

10.2.13 LPOPT : Lower Power Operation Control Register

Base address: SYSC = 0x4001_E000

Offset address: 0x04C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LPOPTEN	—	—	—	BPFCLKDIS	DCLKDIS[1:0]		MPUDIS
Value after reset:	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MPUDIS	MPU Clock Disable Control Stop the MPU operate clock (valid only when LPOPTEN = 1) 0: MPU operates as normal 1: MPU operate clock stops (MPU function disable).	R/W
2:1	DCLKDIS[1:0]	Debug Clock Disable Control 0 0: Debug clock does not stop Others: Debug clock stops (valid only when LPOPT.LPOPTEN = 1)	R/W

Bit	Symbol	Function	R/W
3	BPFCLKDIS	Flash Register R/W Disable Control Stop the Flash register R/W clock (valid only when LPOPT.LPOPTEN = 1) 0: Flash register R/W clock operates as normal 1: Flash register R/W clock stops	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	—	This bit is read as 1. The write value should be 1.	R/W
7	LPOPTEN	Lower Power Operation Enable 0: All lower power counter measure disable 1: All lower power counter measure enable	R/W

The LPOPT register is protected by the PRCR.PRC0 bit.

MPUDIS bit (MPU Clock Disable Control)

[Setting condition]

- Do not set this bit to 1 when the MPU function is used.
- When LPOPT.LPOPTEN = 1 and this bit is set to 1, this bit stops the MPU operate clock.

DCLKDIS[1:0] bits (Debug Clock Disable Control)

[Setting condition]

- Do not set these bits to 1 when in OCD mode or in SCI boot mode.
- When LPOPT.LPOPTEN = 1 and this bit is set to 1, this bit stops the debug system clock to debug function for the chip.

BPFCLKDIS bit (Flash Register R/W Disable Control)

[Setting condition]

- Do not set this bit to 1 when in OCD mode or SCI boot mode.
- Do not set this bit to 1 when operating code flash or data flash through the Flash register.
- Do not set this bit to 1 when operating data flash.
- Do not set this bit to 1 when system transfers power control mode (for example, transfer High-speed mode to Middle-speed mode or transfer High-speed mode to Low-speed mode).
- When LPOPT.LPOPTEN = 1 and this bit is set to 1, this bit stops the Flash register R/W clock.

LPOPTEN bit (Lower Power Operation Enable)

[Setting condition]

- Setting this bit to 1 decreases the MCU power consumption but creates limitations in the system.

10.3 Reducing Power Consumption by Switching Clock Signals

The clock frequency changes when the SCKDIVCR register is set.

For information on module and clock associations, see section 8.2.1. SCKDIVCR : System Clock Division Control Register.

10.4 Module-Stop Function

The module stop function can stop the clock supply set for each peripheral module.

When the MSTPmi bit (m = A to D, i = 31 to 0) in MSTPCRn (n = A to D) is set to 1, the specified module stops operating and enters the module-stop state, but the CPU continues to operate independently. Setting the MSTPmi bit to 0 cancels the module-stop state, allowing the module to resume operation at the end of the bus cycle.

After a reset is canceled, all modules other than the DTC modules are placed in the module-stop state. Do not access the module while the corresponding MSTPmi bit is 1. Additionally, do not set 1 to the MSTPmi bit while the corresponding module is accessed.

10.5 Function for Lower Operating Power Consumption

By selecting an appropriate operating power consumption control mode according to the operating frequency, power consumption can be reduced in Normal mode, Sleep mode, and Snooze mode.

10.5.1 Setting Operating Power Control Mode

Ensure the operating condition such as the frequency range is always within the specified range before and after switching the operating power control modes.

This section provides example procedures for switching operating power control modes.

Table 10.5 Available oscillators in each mode

Mode	Oscillator				
	High-speed on-chip oscillator	Middle-speed on-chip oscillator	Low-speed on-chip oscillator	Main clock oscillator	IWDT-dedicated on-chip oscillator
High-speed	Available	Available	Available	Available	Available
Middle-speed	Available	Available	Available	Available	Available
Low-speed	Available	Available	Available	N/A	Available
Subosc-speed	N/A	N/A	Available	N/A	Available

(1) Switching from a higher power mode to a lower power mode

Example 1: From High-speed mode to Low-speed mode:

(Operation begins in High-speed mode)

1. Change the oscillator to what is used in Low-speed mode. Set the frequency of each clock lower than or equal to the maximum operating frequency in Low-speed mode.
2. Turn off the oscillator that is not required in Low-speed mode.
3. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
4. Set the OPCCR.OPCM[1:0] bits to 11b (Low-speed mode).
5. Confirm that OPCCR.OPCMTSF flag is 0 (indicates transition completed).

(Operation is now in Low-speed mode)

Example 2: From High-speed mode to Subosc-speed mode

(Operation begins in High-speed mode)

1. Switch the clock source to LOCO. Turn off HOCO, MOCO and main oscillator.
2. Confirm that all clock sources other than LOCO are stopped.
3. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).
4. Set the SOPCCR.SOPCM bit to 1 (Subosc-speed mode).
5. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).

(Operation is now in Subosc-speed mode)

(2) Switching from a lower power mode to a higher power mode

Example 1: From Subosc-speed mode to High-speed mode

(Operation begins in Subosc-speed mode)

1. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).
2. Set the SOPCCR.SOPCM bit to 0 (High-speed mode).
3. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).
4. Turn on the required oscillator in High-speed mode.

5. Set the frequency of each clock lower than or equal to the maximum operating frequency for High-speed mode.

(Operation is now in High-speed mode)

Example 2: From Low-speed mode to High-speed mode

(Operation begins in Low-speed mode)

1. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
2. Set the OPCCR.OPCM[1:0] bits to 00b (High-speed mode).
3. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
4. Turn on any required oscillator in High-speed mode.
5. Set the frequency of each clock lower than or equal to the maximum operating frequency for High-speed mode.

(Operation is now in High-speed mode)

10.5.2 Operating Range

Figure 10.2 to Figure 10.5 show the ICLK operating voltages and frequencies. However, peripheral module clocked by PCLKB and PCLKD is not equal to ICLK.

High-speed mode

The maximum operating frequency during a flash read is 64 MHz for ICLK. The operating voltage range during a flash read is 1.8 to 5.5 V.

During flash programming/erasure, the operating frequency range is 1 to 64 MHz and the operating voltage range is 1.8 to 5.5 V.

Figure 10.2 shows the operating voltages and frequencies in High-speed mode.

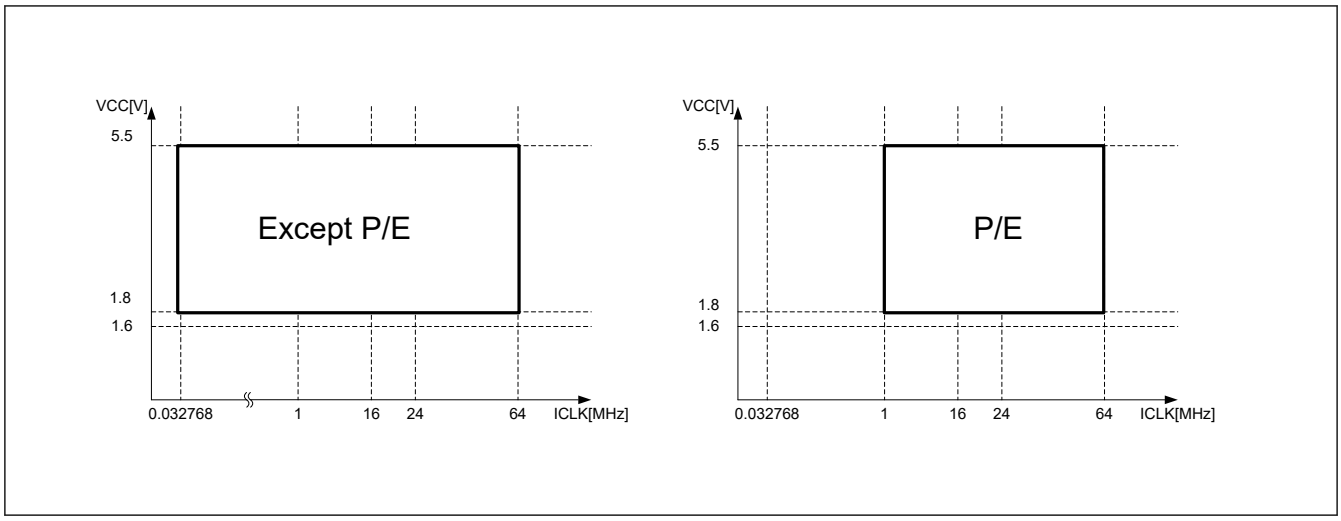


Figure 10.2 Operating voltages and frequencies in High-speed mode

Middle-speed mode

The power consumption of this mode is lower than that of High-speed mode under the same conditions.

The maximum operating frequency during a flash read is 24 MHz for ICLK. The operating voltage range during a flash read is 1.6 to 5.5 V. However, the maximum operating frequency during a flash read is 4 MHz when the operating voltage is 1.6 to 1.8 V.

During flash programming/erasure, the operating frequency range is 1 to 24 MHz and the operating voltage range is 1.6 to 5.5 V. However, the maximum operating frequency during flash programming/erasure is 4 MHz when the operating voltage is 1.6 to 1.8 V.

Figure 10.3 shows the operating voltages and frequencies in Middle-speed mode.

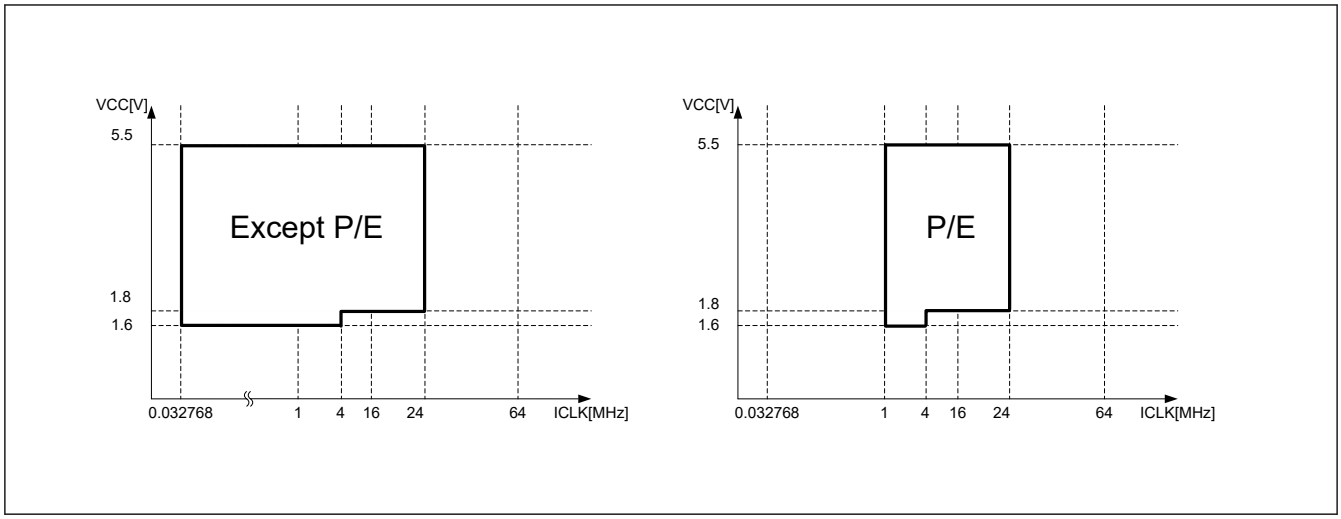


Figure 10.3 Operating voltages and frequencies in Middle-speed mode

Low-speed mode

The maximum operating frequency during a flash read is 2 MHz for ICLK. The operating voltage range during a flash read is 1.6 to 5.5 V.

During flash programming/erasure, the operating frequency range is 1 to 2 MHz and the operating voltage range is 1.6 to 5.5 V.

Figure 10.4 shows the operating voltages and frequencies in Low-speed mode.

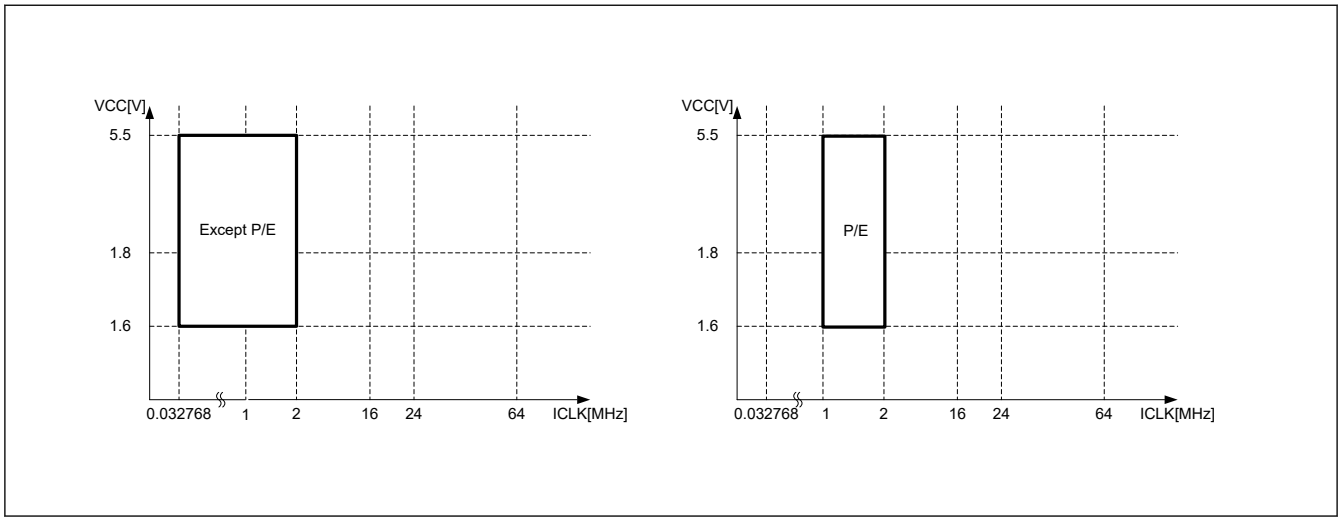


Figure 10.4 Operating voltages and frequencies in Low-speed mode

Note: Selecting the main clock oscillator as system clock source is prohibited if using the Low-speed mode.

Subosc-speed mode

The maximum operating frequency during a flash read is 37.6832 kHz for ICLK. The operating voltage range during a flash read is 1.6 to 5.5 V. P/E operations for flash memory are prohibited.

Using the oscillators other than LOCO is prohibited. Setting the SCKDIVCR register to a value other than 0x00000000 is also prohibited.

Figure 10.5 shows the operating voltages and frequencies in Subosc-speed mode.

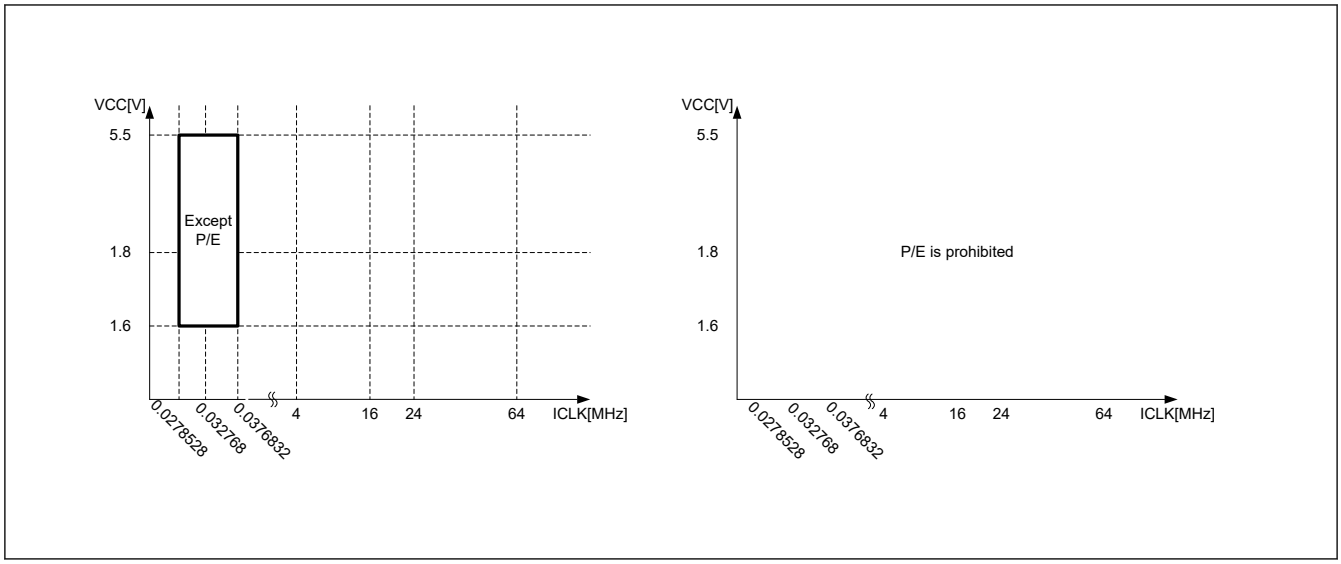


Figure 10.5 Operating voltages and frequencies in Subosc-speed mode

10.6 Sleep Mode

10.6.1 Transitioning to Sleep Mode

When a WFI instruction is executed while SBYCR.SSBY bit is 0, the MCU enters Sleep mode. In this mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop. Available resets or interrupts in Sleep mode cause the MCU to cancel Sleep mode. All interrupt sources are available. If using an interrupt to cancel Sleep mode, you must set the associated IELSRn register before executing a WFI instruction. For details, see section 12, Interrupt Controller Unit (ICU).

Counting by IWDT stops when the MCU enters Sleep mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 1 (IWDT stops in Sleep mode, Software Standby mode, or Snooze mode).

Counting by IWDT continues when the MCU enters Sleep mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 0 (IWDT does not stop in Sleep mode, Software Standby mode, or Snooze mode).

Counting by WDT stops when the MCU enters Sleep mode while the WDT is in auto start mode and the OFS0.WDTSTPCTL bit is 1 (WDT stops in Sleep mode). Similarly, counting by WDT stops when the MCU enters Sleep mode while the WDT is in register start mode and the WDTCSTPR.SLCSTP bit is 1 (WDT stops in Sleep mode).

Counting by WDT continues when the MCU enters Sleep mode while the WDT is in auto start mode and the OFS0.WDTSTPCTL bit is 0 (WDT does not stop in Sleep mode). Similarly, counting by WDT continues when the MCU enters Sleep mode while the WDT is in register start mode and the WDTCSTPR.SLCSTP bit is 0 (WDT does not stop in Sleep mode).

10.6.2 Canceling Sleep Mode

Sleep mode is canceled by:

- An interrupt
- A RES pin reset
- A power-on reset
- A voltage monitor reset
- An SRAM parity error reset
- A bus master MPU error reset
- A bus slave MPU error reset
- A reset caused by an IWDT or a WDT underflow

The operations are as follows:

1. Canceling by an interrupt
   When an interrupt request is generated, Sleep mode is canceled and the MCU starts the interrupt handling.
2. Canceling by RES pin reset
   When the RES pin is driven low, the MCU enters the reset state. Be sure to keep the RES pin low for the time period specified in section 35, Electrical Characteristics. When the RES pin is driven high after the specified time period, the CPU starts the reset exception handling.
3. Canceling by IWDT reset
   - Sleep mode is canceled by an internal reset generated by an IWDT underflow and the MCU starts the reset exception handling. However, IWDT stops in Sleep mode and an internal reset for canceling Sleep mode is not generated in the following conditions:
   - OFS0.IWDTSTRT = 0 and OFS0.IWDTSTPCTL = 1.
4. Canceling by WDT reset
   Sleep mode is canceled by an internal reset generated by a WDT underflow and the MCU starts the reset exception handling. However, WDT stops in Sleep mode even when counting in Normal mode and an internal reset for canceling Sleep mode is not generated in the following conditions:
   - OFS0.WDTSTRT = 0 (auto start mode) and OFS0.WDTSTPCTL = 1
   - OFS0.WDTSTRT = 1 (register start mode) and WDTCSTPR.SLCSTP = 1.
5. Canceling by other resets available in Sleep mode
   Sleep mode is canceled by other resets and the MCU starts the reset exception handling.

Note: For details on proper setting of the interrupts, see section 12, Interrupt Controller Unit (ICU).

10.7 Software Standby Mode

10.7.1 Transition to Software Standby Mode

When a WFI instruction is executed while SBYCR.SSBY bit is 1, the MCU enters Software Standby mode. In this mode, the CPU, most of the on-chip peripheral functions and oscillators stop. However, the contents of the CPU internal registers and SRAM data, the states of on-chip peripheral functions and the I/O ports are retained. Software Standby mode allows a significant reduction in power consumption because most of the oscillators stop in this mode. Table 10.2 shows the status of each on-chip peripheral functions and oscillators. Available resets or interrupts in Software Standby mode cause the MCU to cancel Software Standby mode. See Table 10.3 for available interrupt sources and section 12.2.8. WUPEN : Wakeup Interrupt Enable Register for information on how to wake up the MCU from Software Standby mode. If using an interrupt to cancel Software Standby mode, you must set the associated IELSRn register before executing a WFI instruction. For details, see section 12, Interrupt Controller Unit (ICU).

Counting by IWDT stops when the MCU enters Software Standby mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 1 (IWDT stops in Sleep mode, Software Standby mode, and Snooze mode). Counting by IWDT continues if the MCU enters Software Standby mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 0 (IWDT does not stop in Sleep mode, Software Standby mode, and Snooze mode).

WDT stops counting when the MCU enters Software Standby mode.

Do not enter Software Standby mode while OSTDCR.OSTDE = 1 (oscillation stop detection function is enabled). To enter Software Standby mode, execute a WFI instruction after disabling the oscillation stop detection function (OSTDCR.OSTDE = 0). If executing a WFI instruction while OSTDCR.OSTDE = 1, the MCU enters Sleep mode even when SBYCR.SSBY = 1. In addition, do not enter Software Standby mode while the flash memory performs a programming or erasing procedure. To enter Software Standby mode, execute a WFI instruction after the programming or erasing procedure completes.

10.7.2 Canceling Software Standby Mode

Software Standby mode is canceled by:

- An available interrupt shown in Table 10.3
- A RES pin reset
- A power-on reset
- A voltage monitor reset

- A reset caused by an IWDT underflow.

On exiting Software Standby mode, the oscillators that operate before the transition to the mode restart. After all the oscillators are stabilized, the MCU returns to Normal mode from Software Standby mode. See section 12.2.8. WUPEN : Wakeup Interrupt Enable Register for information on how to wake up the MCU from Software Standby mode.

You can cancel Software Standby mode in any of the following ways:

1. Canceling by an interrupt
   When an available interrupt request (see Table 10.3) is generated, an oscillator that operates before the transition to Software Standby mode restarts. After all the oscillators are stabilized, the MCU returns to Normal mode from Software Standby mode and starts the interrupt handling.
2. Canceling by a RES pin reset
   When the RES pin is driven low, the MCU enters the reset state, and the oscillators whose default status is operating, start the oscillation. Be sure to keep the RES pin low for the time period specified in section 35, Electrical Characteristics. When the RES pin is driven high after the specified time period, the CPU starts the reset exception handling.
3. Canceling by a power-on reset
   Software Standby mode is canceled by a power-on reset and the MCU starts the reset exception handling.
4. Canceling by a voltage monitor reset
   Software Standby mode is canceled by a voltage monitor reset from the voltage detection circuit and the MCU starts the reset exception handling.
5. Canceling by IWDT reset
   Software Standby mode is canceled by an internal reset generated by an IWDT underflow and the MCU starts the reset exception handling. However, IWDT stops in Software Standby mode and an internal reset for canceling Software Standby mode is not generated in the following condition:
   - OFS0.IWDTSTRT = 0 and OFS0.IWDTSTPCTL = 1.

10.7.3 Example of Software Standby Mode Application

Figure 10.6 shows an example of entry to Software Standby mode on detection of a falling edge of the IRQn pin, and exit from Software Standby mode by a rising edge of the IRQn pin.

In this example, an IRQn pin interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 00b (falling edge) in Normal mode, and the IRQCRi.IRQMD[1:0] bits are set to 01b (rising edge). After that, the SBYCR.SSBY bit is set to 1 and a WFI instruction is executed. As a result, entry to Software Standby mode completes and exit from Software Standby mode is initiated by a rising edge of the IRQn pin.

Setting the ICU is also required to exit Software Standby mode. For details, see section 12, Interrupt Controller Unit (ICU). The oscillation stabilization time in Figure 10.6 is specified in section 35, Electrical Characteristics.

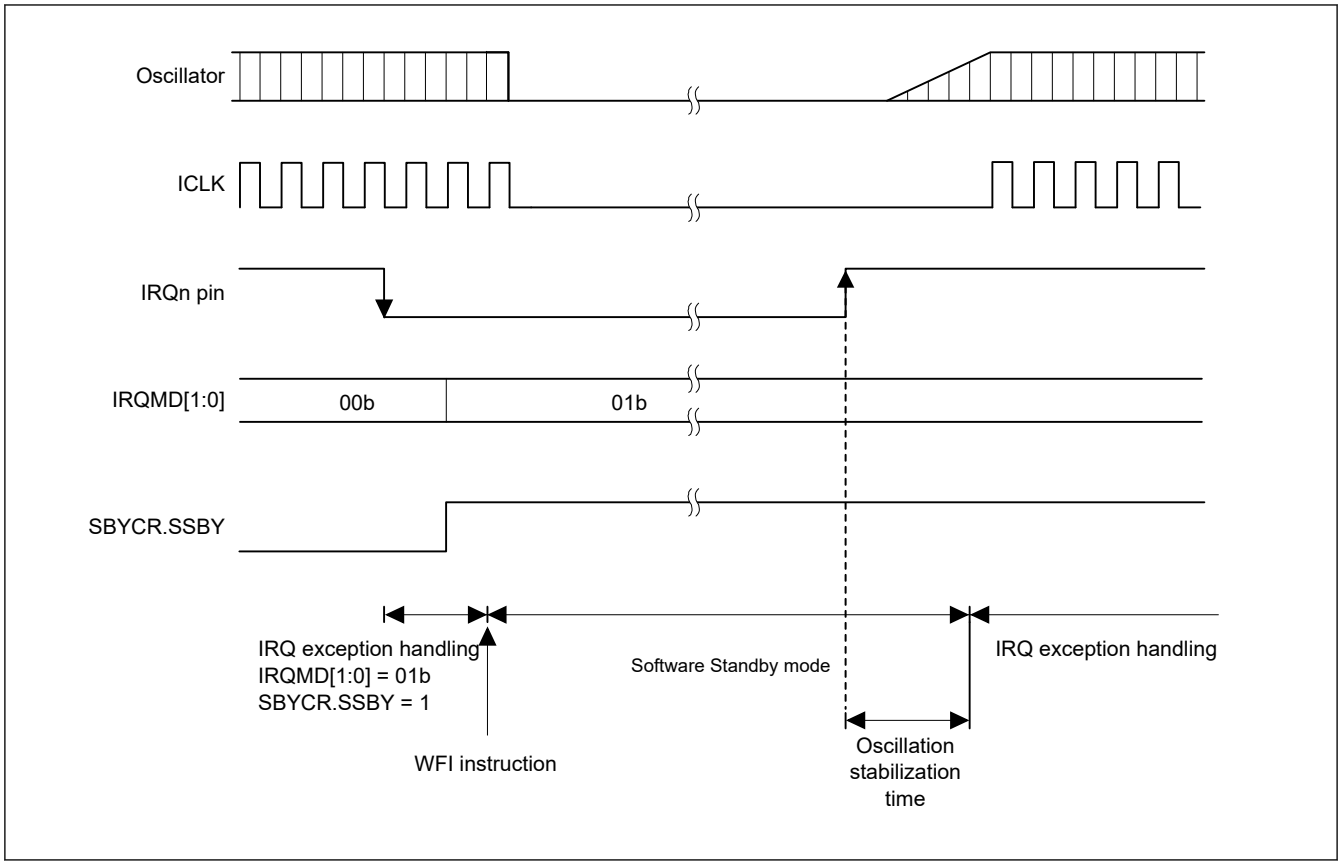
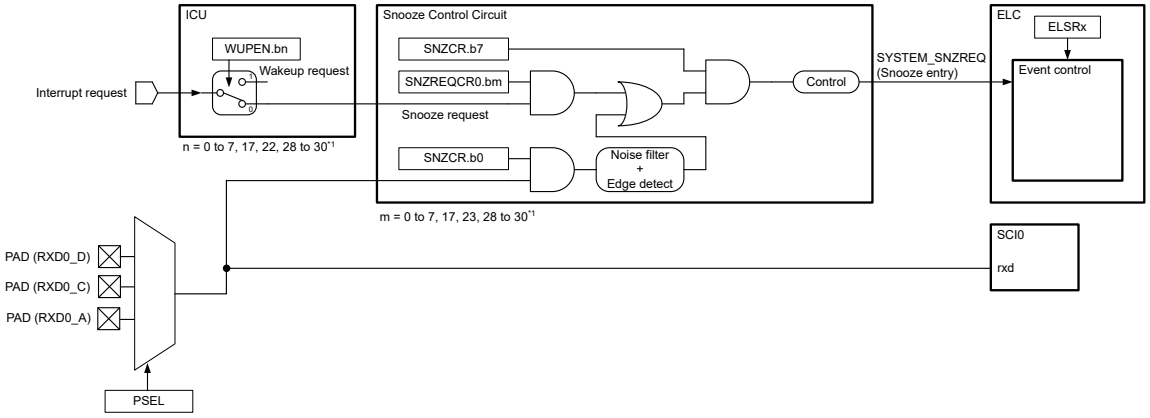


Figure 10.6 Example of Software Standby mode application

10.8 Snooze Mode

10.8.1 Transition to Snooze Mode

Figure 10.7 shows snooze mode entry configuration. When the snooze control circuit receives a snooze request in Software Standby mode, the MCU transfers to Snooze mode. In this mode, some peripheral modules operate without waking up the CPU. Table 10.2 shows the peripheral modules that can operate in Snooze mode. Also, DTC operation in Snooze mode can be selected by setting the SNZCR.SNZDTCEN bit.



Note 1. For ACMPHS the bit of snooze request is bit 23 in the SNZREQCR0 register and of wakeup request is bit 22 in the WUPEN register.

Figure 10.7 Snooze mode entry configuration

Table 10.6 shows the snooze requests to switch the MCU from Software Standby mode to Snooze mode. To use the listed snooze requests as a trigger to switch to Snooze mode, you must set the associated SNZREQENn bit of the SNZREQCR0 register or RXDREQEN bit of the SNZCR register before entering Software Standby mode.

Note: Do not enable multiple snooze requests at the same time.

Table 10.6 Available snooze requests to switch to Snooze mode

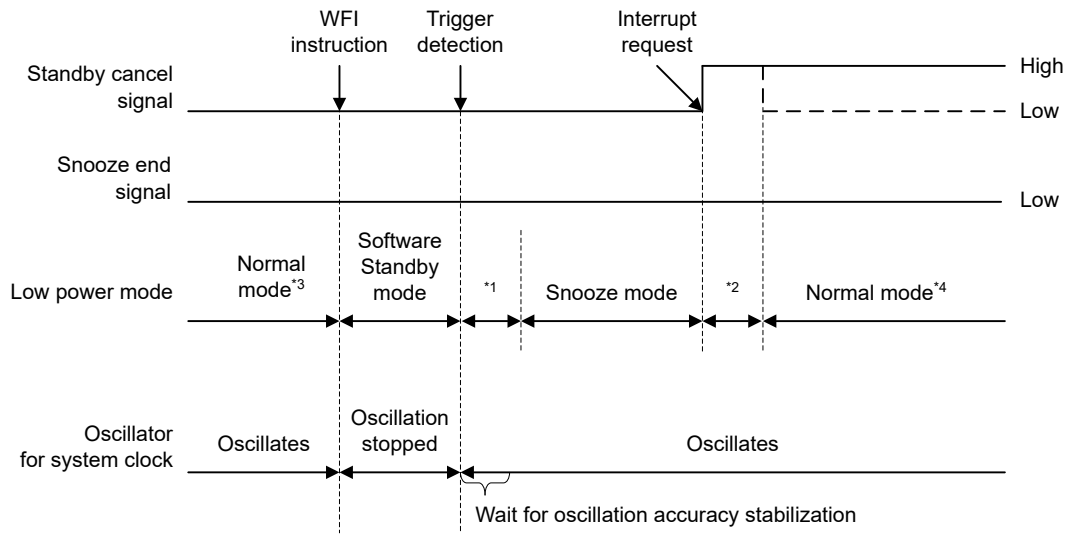
Snooze request	Control Register	
	Register	Bit
PORT_IRQn (n = 0 to 7)	SNZREQCR0	SNZREQENn (n = 0 to 7)
KEY_INTKR	SNZREQCR0	SNZREQEN17
ACMP_HS0	SNZREQCR	SNZREQEN23
AGT1_AGTI	SNZREQCR0	SNZREQEN28
AGT1_AGTCMAI	SNZREQCR0	SNZREQEN29
AGT1_AGTCMBI	SNZREQCR0	SNZREQEN30
RXD0 falling edge	SNZCR	RXDREQEN*1

Note 1. RXDREQEN bit must not be set to 1 unless in asynchronous mode.

Clear the DTCST.DTCST bit to 0 before executing a WFI instruction except when using DTC in Snooze mode. If DTC is required in Snooze mode, set the DTCST.DTCST bit to 1 before executing a WFI instruction.

10.8.2 Canceling Snooze Mode

Snooze mode is canceled by an interrupt request that is available in Software Standby mode or a reset. Table 10.3 shows the requests that can be used to exit each mode. After canceling the Snooze mode, the MCU enters Normal mode and proceeds with exception processing for the given interrupt or reset. The action triggered by the interrupt requests, selected in SELSR0, cancels Snooze mode. Interrupt canceling Snooze mode must be selected in IELSRn to link to the NVIC for the corresponding interrupt handling. See section 12, Interrupt Controller Unit (ICU) for information on SELSR0 and IELSRn registers.



Note 1. Transition time from Software Standby mode to Snooze mode.
Note 2. Transition time from Snooze mode to Normal mode.
Note 3. Enable Snooze mode (SNZCR.SNZE = 1) immediately before switching to Software Standby mode.
Note 4. Disable Snooze mode (SNZCR.SNZE = 0) immediately after returning to Normal mode.

Figure 10.8 Canceling of Snooze mode when an interrupt request signal is generated

10.8.3 Returning from Snooze Mode to Software Standby Mode

Table 10.7 shows the snooze end request that can be used as triggers to return to Software Standby mode. The snooze end requests are available only in Snooze mode. If the requests are generated when the MCU is not in Snooze mode, they are ignored. When multiple requests are selected, each of the requests invokes transition to Software Standby mode from Snooze mode.

Table 10.8 shows the snooze end conditions that consist of the snooze end requests and the conditions of the peripheral modules. The SCI0, ADC12, and DTC modules can keep the MCU in Snooze mode until they complete the operation. However, an AGTWn (n = 1) underflow as a trigger to return to Software Standby mode cancels Snooze mode without waiting for the completion of SCI0 operation.

Figure 10.9 shows the timing diagram for the transition from Snooze mode to Software Standby mode. This mode transition occurs according to which snooze end requests are set in the SNZEDCR0 register. A snooze request is cleared automatically after returning to Software Standby mode.

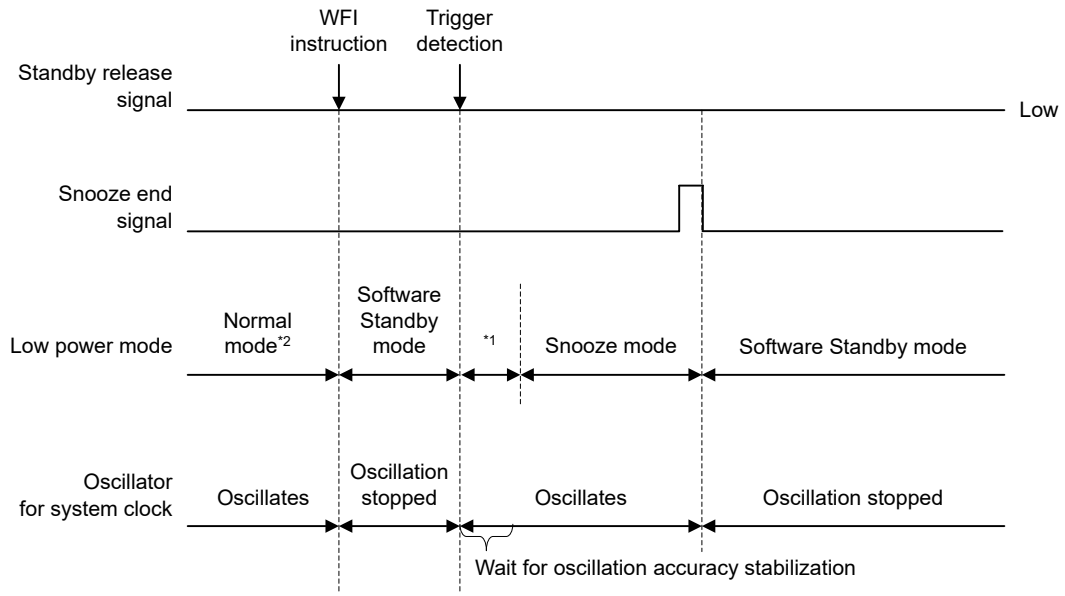
Table 10.7 Available snooze end requests (triggers to return to Software Standby mode)

Peripheral Module	Snooze end request	Enable/Disable Control	
		Register	Symbol
AGTW1	AGTW1 underflow (AGT1_AGTI)	SNZEDCR0	AGTUNFED
DTC	Last DTC transmission completion (DTC_COMPLETE)	SNZEDCR0	DTCZRED
DTC	Not Last DTC Transmission Completion (DTC_TRANSFER)	SNZEDCR0	DTCNZRED
ADC120	Window A/B compare match (ADC120_WCMPM)	SNZEDCR0	AD0MATED
ADC120	Window A/B compare mismatch (ADC120_WCMPUM)	SNZEDCR0	AD0UMTED
SCI0	SCI0 address mismatch (SCI0_DCUF)	SNZEDCR0	SCI0UMTED

Table 10.8 Snooze end conditions

Operating module when a snooze end request occurs	Snooze end request	Other than AGTW1 underflow
DTC	The MCU transfers to the Software Standby mode after all of the modules listed in this table complete operation.	The MCU transfers to the Software Standby mode after all of the modules listed to the left of this column complete the operation.
ADC120		
SCI0	The MCU transfers to the Software Standby mode immediately after a snooze end request is generated.	
Other than specified	The MCU transfers to the Software Standby mode immediately after a snooze end request is generated.	

Note: If the DTC is used to activate the ADC120 or SCI0, the MCU transitions to Software Standby mode after a snooze end request is generated.

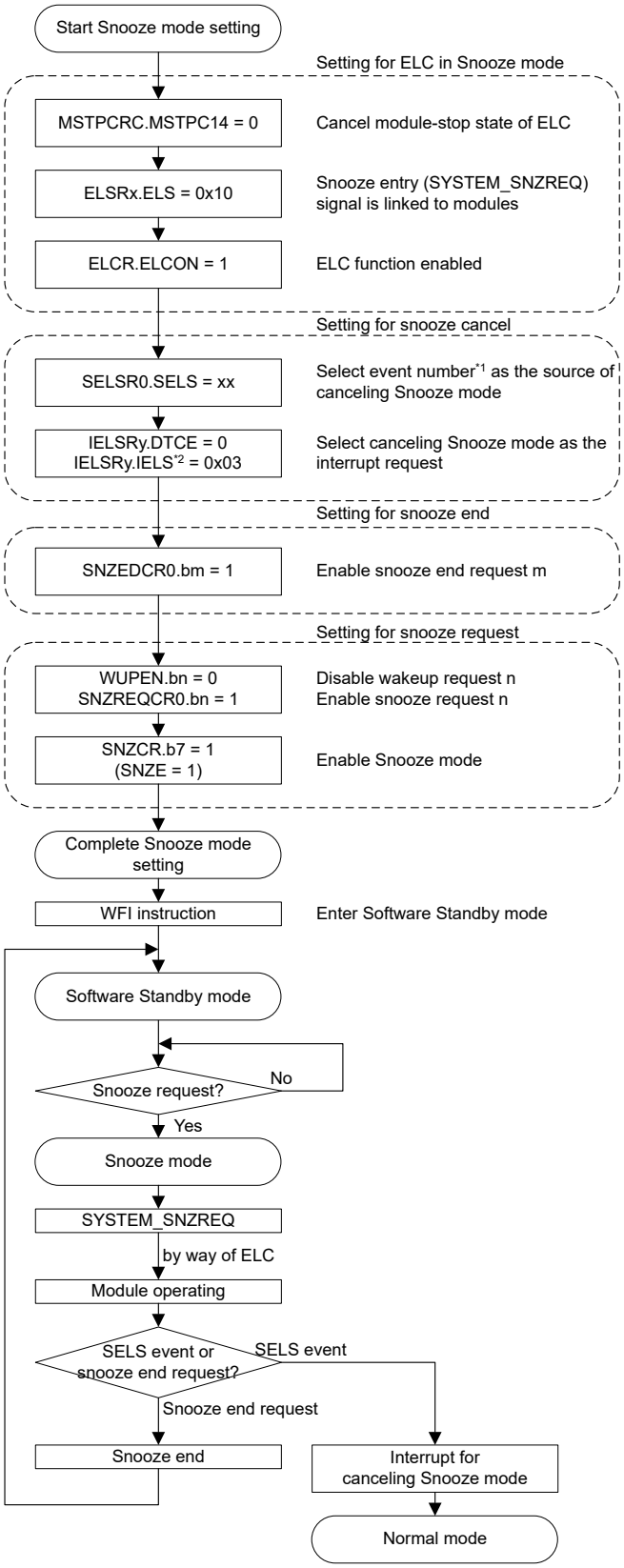


Note 1. Transition time from Software Standby mode to Snooze mode.
Note 2. Enable Snooze mode (SNZCR.SNZE = 1) immediately before switching to Software Standby mode.

Figure 10.9 Canceling of Snooze mode when an interrupt request signal is not generated

10.8.4 Snooze Operation Example

Figure 10.10 shows an example setting for using ELC in Snooze mode.



Note 1. See Table 12.4
Note 2. y = 0, 4, 8, 12, 16, 20, 24, 28

Figure 10.10 Setting example of using ELC in Snooze mode

The MCU can transmit and receive data in SCI0 asynchronous mode without CPU intervention. When using the SCI0 in Snooze mode, use one of the following operating modes:

- High-speed mode
- Middle-speed mode
- Low-speed mode.

Do not use Subosc-speed mode. Table 10.9 shows the maximum transfer rate of SCI0 in Snooze mode. When using the SCI0 in Snooze mode, use the following settings :

- BGDM = 0
- ABCS = 0
- ABCSE = 0.

See section 24, Serial Communications Interface (SCI) for information on these bits.

High-speed mode, Middle-speed mode, Low-speed mode

Table 10.9 HOCO: ± 1 % (Unit: bps)

Maximum division ratio of ICLK, PCLKB, and PCLKD	HOCO frequency			
	24 MHz	32 MHz	48 MHz	64 MHz
1	76800*1	76800*5	—	—
2	76800*2	76800*6	76800*8	76800*12
4	76800*3	76800*7	76800*9	76800*13
8	76800*4	62500	76800*10	76800*14
16	46875	62500	76800*11	62500
32	23437	31250	46875	62500
64	11718	15625	23437	31250

Note 1. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 0x06, SCI0.MDDR = 0xB8 must be used for 76800 bps.
Note 2. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 0x03, SCI0.MDDR = 0xD2 must be used for 76800 bps.
Note 3. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 0x01, SCI0.MDDR = 0xD2 must be used for 76800 bps.
Note 4. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 0x00, SCI0.MDDR = 0xD2 must be used for 76800 bps.
Note 5. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 0x08, SCI0.MDDR = 0xB1 must be used for 76800 bps.
Note 6. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 0x04, SCI0.MDDR = 0xC5 must be used for 76800 bps.
Note 7. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 0x02, SCI0.MDDR = 0xEC must be used for 76800 bps.
Note 8. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 0x06, SCI0.MDDR = 0xB8 must be used for 76800 bps.
Note 9. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 0x03, SCI0.MDDR = 0xD2 must be used for 76800 bps.
Note 10. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 0x01, SCI0.MDDR = 0xD2 must be used for 76800 bps.
Note 11. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 0x00, SCI0.MDDR = 0xD2 must be used for 76800 bps.
Note 12. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 0x08, SCI0.MDDR = 0xB1 must be used for 76800 bps.
Note 13. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 0x04, SCI0.MDDR = 0xC5 must be used for 76800 bps.
Note 14. SCI0.SMR.CKS[1:0] = 00b, SCI0.SEMR.BRME = 1, SCI0.BRR = 0x02, SCI0.MDDR = 0xEC must be used for 76800 bps.

Figure 10.11 shows a setting example for using SCI0 in Snooze mode entry.

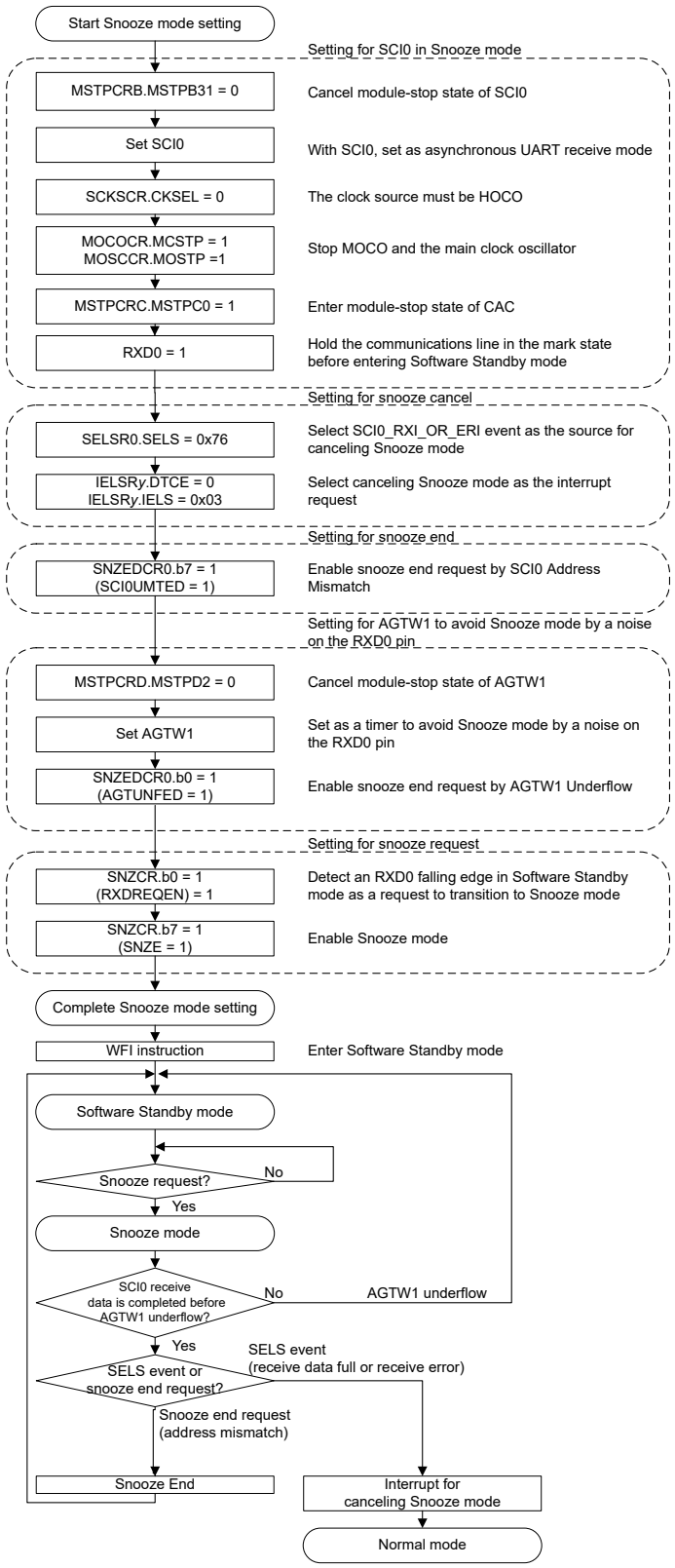


Figure 10.11 Setting example of using SCI0 in Snooze mode entry

10.9 Usage Notes

10.9.1 Register Access

(1) Do not write to registers listed in this section in any of the following conditions:

[Registers]

- All registers with a peripheral name of SYSTEM.

[Conditions]

- OPCCR.OPCMTSF = 1 or SOPCCR.SOPCMTSF = 1 (during transition of the operating power control mode)
- During the time period from executing a WFI instruction to returning to Normal mode
- Flash P/E mode, data flash P/E mode

(2) Valid setting for the clock-related registers

Table 10.10 and Table 10.11 show the valid settings for the clock-related registers in each operating power control mode. Do not write any value other than the valid setting, otherwise it is ignored. Additionally, each register has some prohibited settings under certain conditions other than those related to the operating power control modes. See section 8, Clock Generation Circuit for these other conditions of each register.

Table 10.10 Valid settings for the clock-related registers (1)

Mode	Valid settings					
	SCKSCR.CKSEL[2:0] CKOCR.CKOSEL[2:0]	SCKDIVCR.ICK[2:0]	HOCOCR.HCSTP	MOCOCR.MCSTP	LOCOCR.LCSTP	MOSCCR.MOSTP
High-speed Middle-speed	000b (HOCO) 001b (MOCO) 010b (LOCO) 011b (MOSC)	000b (1/1) 001b (1/2) 010b (1/4) 011b (1/8) 100b (1/16) 101b (1/32) 110b (1/64)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)
Low-speed	000b (HOCO) 001b (MOCO) 010b (LOCO)	000b (1/1) 001b (1/2) 010b (1/4) 011b (1/8) 100b (1/16) 101b (1/32) 110b (1/64)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)	0 (operating) 1 (stopped)	1 (stopped)
Subosc-speed	010b (LOCO)	000b (1/1)	1 (stopped)	1 (stopped)	0 (operating) 1 (stopped)	1 (stopped)

Table 10.11 Valid settings for the clock-related registers (2)

Operating oscillator	Valid settings	
	SOPCCR.SOPCM	OPCCR.OPCM[1:0]
High-speed on-chip oscillator	0	00b, 01b, 11b
Middle-speed on-chip oscillator		
Main clock oscillator		00b, 01b
Low-speed on-chip oscillator	0, 1	00b, 01b, 11b
IWDT-dedicated on-chip oscillator		

(3) Do not write to registers listed in this section for the following condition:

[Registers]

- SCKSCR, OPCCR.

[Condition]

- SOPCCR.SOPCM = 1 (Subosc-speed mode).

(4) Do not write to registers listed in this section by DTC:

[Registers]

- MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD.

(5) Do not write to registers listed in this section in Snooze mode. They must be set before entering Software Standby mode:

[Registers]

- SNZCR, SNZEDCR0, SNZREQCR0.

(6) Write access to registers listed in this section is invalid when PRCR.PRC1 bit is 0:

[Registers]

- SBYCR, SNZCR, SNZEDCR0, SNZREQCR0, OPCCR, SOPCCR.

10.9.2 I/O Port pin states

The I/O port pin states in Software Standby mode and Snooze mode, unless modifying in Snooze mode, are the same before entering the modes. Therefore, power consumption is not reduced while the output signals are held high.

10.9.3 Module-Stop State of DTC

Before writing 1 to MSTPCRA.MSTPA22, clear the DTCST.DTCST bit of the DTC to 0. For details, see section 15, Data Transfer Controller (DTC).

10.9.4 Internal Interrupt Sources

Interrupts do not operate in the module-stop state. If setting the module-stop bit while an interrupt request is generated, a CPU interrupt source or a DTC startup source cannot be cleared. Always disable the associated interrupts before setting the module-stop bits.

10.9.5 Transitioning to Low Power Modes

Because the MCU does not support wakeup by events, do not enter the low power modes such as Sleep mode, Software Standby mode by executing a WFE instruction. Also, do not set the SLEEPDEEP bit of the System Control Register in the Cortex-M23 core because the MCU does not support low power modes by SLEEPDEEP.

10.9.6 Timing of WFI Instruction

It is possible for the WFI instruction to be executed before I/O register write is completed, in which case operation might not be as intended. This can happen if the WFI is placed immediately after a write to an I/O register. To avoid this problem, read back the register that was written to confirm that the write completed.

10.9.7 Writing to the WDT/IWDT Registers by DTC in Sleep Mode or Snooze Mode

Do not write to the WDT or IWDT registers by the DTC while WDT or IWDT is stopped after entering Sleep mode or Snooze mode.

10.9.8 Oscillators in Snooze Mode

Oscillators that stop on entering Software Standby mode automatically restart when a trigger for switching to Snooze mode is generated. The MCU does not enter Snooze mode until all the oscillators stabilize. If in Snooze mode, you must disable oscillators that are not required in Snooze mode before entering Software Standby mode. Otherwise, the transition from Software Standby mode to Snooze mode takes longer.

10.9.9 Snooze Mode Entry by RXD0 Falling Edge

When the SNZCR.RXDREQEN bit is 1, the falling edge of RXD0 pin is used to switch MCU from Software Standby mode to Snooze mode when using UART of SCI0 in Snooze mode. In this case an interrupt such as SCI0_ERI, SCI0_RXI or an address mismatch event is used as the source for canceling Snooze mode. However noise on the RXD0 pin might cause the MCU to transfer from Software Standby mode to Snooze mode unexpectedly. In this case if the MCU does not receive RXD0 data after the noise, an interrupt such as SCI0_ERI or SCI0_RXI, or an address mismatch event is not generated and the MCU stays in Snooze mode. This can be avoided by using AGTWn (n = 1) underflow interrupt to return to Software Standby mode or Normal mode unless otherwise UART receive data is completed before AGTWn (n = 1) underflow. However, do not use the AGTWn (n = 1) underflow as a source to return to Software Standby mode during an UART communication. This causes the UART to stop the operation in a half-finished state.

10.9.10 Using UART of SCI0 in Snooze Mode

When using UART in Snooze mode, ensure that the snooze request (RXD0 falling edge) does not conflict with the wakeup requests set by the WUPEN register, otherwise UART cannot be guaranteed.

When using UART in Snooze mode, the following conditions must be satisfied:

- The clock source must be HOCO
- MOCO, and the main clock oscillator must be stopped before entering Software Standby mode
- The RXD0 pin must be kept high before entering Software Standby mode
- A transition to Software Standby mode must not occur during an SCI0 communication
- The MSTPCRC.MSTPC0 bit must be 1 before entering Software Standby mode.

10.9.11 Conditions of A/D Conversion Start in Snooze Mode

ADC120 can only be triggered by the ELC in Snooze mode. Do not use software trigger or ADTRGn (n = 0) pin.

10.9.12 ELC Events in Snooze Mode

This section lists available ELC events in Snooze mode. Do not use any other events. If starting peripheral modules for the first time after entering Snooze mode, the Event Link Setting Register (ELSRn) must set a Snooze mode entry event (SYSTEM_SNZREQ) as the trigger.

- Snooze mode entry (SYSTEM_SNZREQ)
- DTC transfer end (DTC_DTCEND)
- ADC120 window A/B compare match (ADC120_WCMPM)
- ADC120 window A/B compare mismatch (ADC120_WCMPUM)
- Data operation circuit interrupt (DOC_DOPCI).

10.9.13 Module-Stop Function for ADC120

When entering Software Standby mode, it is recommended that you set the ADC120 module-stop state to reduce power consumption. In this case, the ADC120 can be available in Snooze mode by releasing the ADC120 module-stop using the DTC. Similarly, set the module-stop state using the DTC before returning to Software Standby mode from Snooze mode.

10.9.14 Module-Stop Function for an Unused Circuit

A circuit that is not used in user mode might not be reset, and might operate in an unstable state because the clocks are not supplied during an MCU reset. In this case, when the MCU transitions to Low-speed mode or Software Standby mode, the supply current can be increased to a value greater than that stated in this User's Manual, by up to 600 µA. So, initialize the unused circuit as shown in Figure 10.12.

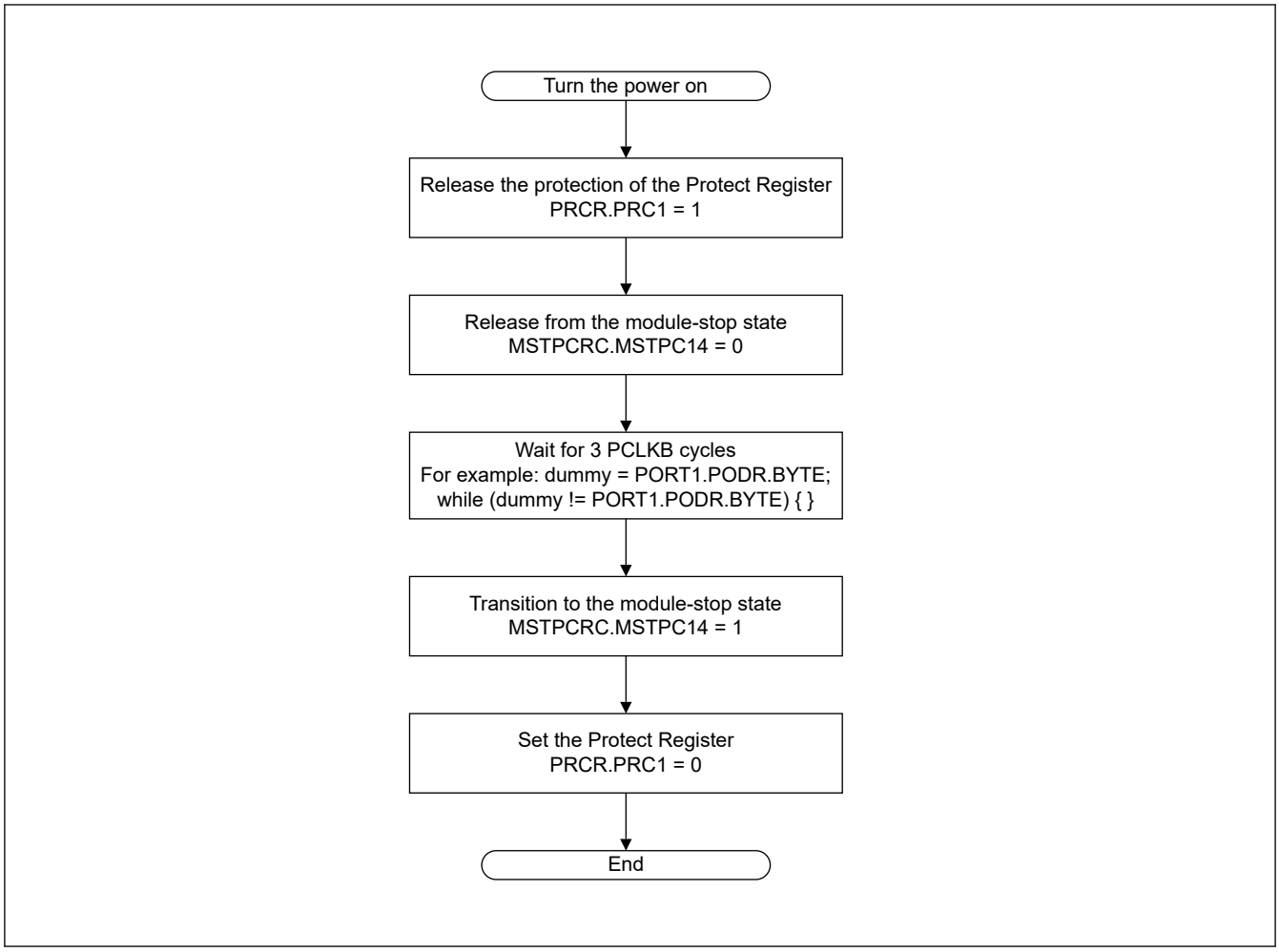


Figure 10.12 Example of initial setting flow for an unused circuit

10.9.15 Notes when Transition to Low Power Modes by Sleep-On-Exit

There are 2 ways to transition to low power modes. One is WFI instruction and the other is Sleep-on-exit. When Sleep-on-exit is used for transition to low power modes, WFI instruction comments written in User's Manual Hardware is applicable to Sleep-on-exit.

10.9.16 Applicable Condition and Notes

(1) Applicable Condition

Transition to Software Standby mode is started by a trigger (WFI instruction or SLEEPONEXIT) with SBYCR.SSBY = 1 set to use Software Standby or Snooze mode.

During the specified interval (ICLK 2 cycle) of transitioning to Software Standby mode, one of the following interrupt requests that is not an interrupt source to return from Software Standby mode is accepted by CPU.

1. SysTick interrupt
   - Exception number 15 of Interrupt vector table
2. Maskable interrupt requests that are not interrupt source to return from Software Standby mode (all of the following are applicable)
   - By WUPEN in exception numbers 16 to 83 in the interrupt vector table those not permitted to return from Software Standby mode
   - Interrupt requests are enabled by Interrupt Set-Enable Register (NVIC_ISERn)
3. Non-maskable interrupt request triggered by the following sources

   SRAM parity error

SRAM ECC failure

Bus master MPU error

Bus slave MPU error

(2) Notes

If the above conditions are met, the MCU will transit to following unintended states. These unintended states can be resolved by a reset or returning to Normal mode with an interrupt request of an interrupt source to return from Software Standby mode.

1. When transitioning to Software Standby mode (SBYCR.SSBY = 1, SNZCR.SNZE = 0)
   Only CPU clock is stopped, and the remaining clocks continue to operate as they were prior to transitioning Software Standby mode.
   - As before the transition to Software Standby mode is started, depending on the setting, timer or other peripherals continue to operate, and an interrupt request related to the peripheral is generated.
   - Because the IWDT and WDT clock-stop function is disabled, a reset or an interrupt for the IWDT and WDT is generated depending on the settings before starting the transition to Software Standby mode.
   - Interrupt requests are held in IR flag (IELSRn, DELSRn).
2. When transitioning to Snooze mode (SBYCR.SSBY = 1, SNZCR.SNZE = 1)
   The transition to Snooze mode is not possible, and the states shown in “1) When transitioning to Software Standby mode” is continued.
   To return to Normal mode by an interrupt source (SELSR0) from Snooze mode depends on whether the interrupt request (SELSR0) to returning from Snooze mode can be generated while DTC operation is disabled.
   If DTC operation is disabled (SNZCR.SNZDTCEN = 0) in Snooze mode, Normal mode can be returned because an interrupt source for the interrupt source (SELSR0) to return from Snooze mode can be generated.
   If DTC operation is enabled (SNZCR.SNZDTCEN = 1) in Snooze mode, the Normal mode cannot be returned because an interrupt request for the interrupt source (SELSR0) to return from Snooze mode cannot be generated.

10.9.17 Workaround

(1) Workaround

To avoid the unintended states described above, apply the following before the terms for transition to Software Standby mode, Snooze mode are met: (For the setting procedure, see "Setting Procedure for Transition to Software Standby Mode or Snooze Mode")

1. Disable SysTick interrupt requests.
   Exception number 15 of Interrupt vector table
2. Disable maskable interrupt requests that are not interrupt sources to return from Software Standby mode.
   Exception number 16 to 83 of Interrupt vector table that WUPEN does not allow to return from Software Standby mode.
3. Stop access from the bus master other than the CPU so that the non-maskable interrupt is not triggered by the following sources.

   SRAM parity error

   SRAM ECC failure

   Bus master MPU error

   Bus slave MPU error

(2) Setting Procedure for Transition to Software Standby Mode or Snooze Mode

This section describes procedures for avoiding unintended states.

The handling of interrupt requests after returning from Software Standby or Snooze mode varies depending on the method used to disable the maskable interrupt request. Either one or the other should be applied.

- Step A: Disable maskable interrupt request acceptance.
  Any interrupt request that occurs while interrupt request acceptance is disabled is discarded.

Table 10.12 Before transitioning to Software Standby Mode or Snooze Mode

Step	Description
1	Stop the bus access from the bus master other than CPU.*1
2	Disable the SysTick interrupt request.*2
3	Clear IELSRn in ICU to disable acceptance of maskable interrupt requests that are not interrupt sources to return from Software Standby mode.
4	Read IELSRn in ICU to confirm that IELSRn in ICU has been cleared.
5	Transition to Software Standby mode (WFI instruction, SLEEPONEXIT).

Note 1. SRAM parity error interrupt, SRAM ECC error interrupt, Bus master MPU error interrupt or Bus slave MPU error interrupt is enabled as a non-maskable interrupt.
Note 2. Disabling a SysTick interrupt request may cause SysTick interrupt request to be delayed by one cycle of SysTick timer without generating the latest SysTick interrupt request.

Table 10.13 After returning from Software Standby or Snooze mode

Step	Description
1	Enable the SysTick interrupt request.
2	Set IELSRn in ICU to enable acceptance of maskable interrupt requests that are not interrupt sources to return from Software Standby mode.
3	Enable bus access from bus masters other than CPU.

- Step B: Disable the maskable interrupt request
The interrupt request generated while the interrupt request is disabled is retained in IELSRn.IR flag. Therefore, after returning from Software Standby or Snooze mode and enabling the maskable interrupt, it is possible to process the interrupt.

Table 10.14 Before transitioning to Software Standby, Snooze, or Deep Software Standby mode

Step	Description
1	Stop the bus access from the bus master other than CPU.*1
2	Disable the SysTick interrupt request.*2
3	Write 1 to the corresponding bit in NVIC_ICERn in CPU to disable maskable interrupt requests that are not interrupt sources to return from Software Standby mode.
4	Execute Data Synchronization Barrier (DSB) instruction.
5	Transition Software Standby mode (WFI instruction, SLEEPONEXIT)

Note 1. SRAM parity error interrupt, SRAM ECC error interrupt, Bus master MPU error interrupt or Bus slave MPU error interrupt is enabled as a non-maskable interrupt.
Note 2. Disabling a SysTick interrupt request may cause SysTick interrupt request to be delayed by one cycle of SysTick timer without generating the latest SysTick interrupt request.

Table 10.15 After returning from Software Standby or Snooze mode

Step	Description
1	Enable the SysTick interrupt request.
2	Write 1 to the corresponding bit in NVIC_ISERn in CPU to enable maskable interrupt requests that are not interrupt sources to return from Software Standby mode.
3	Enable bus access from bus masters other than CPU.

11. Register Write Protection

11.1 Overview

The register write protection function protects important registers from being overwritten due to software errors. The registers to be protected are set with the Protect Register (PRCR).

Table 11.1 lists the association between the bits in the PRCR register and the registers to be protected.

Table 11.1 Association between the bits in the PRCR register and registers to be protected

PRCR bit	Register to be protected
PRC0	• Registers related to the clock generation circuit: SCKDIVCR, SCKSCR, MEMWAIT, MOSCCR, HOCOCR, MOCOCR, CKOCR, OSTDCR, OSTDSR, MOCOUTCR, HOCOUTCR, MOSCWTCR, MOMCR, LOCOCR, LOCOUTCR, HOCOWTCR, LPOPT
PRC1	• Registers related to the low power modes: SBYCR, SNZCR, SNZEDCR0, SNZREQCR0, OPCCR, SOPCCR, SYOCDCR
PRC3	• Registers related to the LVD: LVD1CR1, LVD1SR, LVD2CR1, LVD2SR, LVCMPCR, LVDLVLR, LVD1CR0, LVD2CR0

11.2 Register Descriptions

11.2.1 PRCR : Protect Register

Base address: SYSC = 0x4001_E000

Offset address: 0x3FE

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]								—	—	—	—	PRC3	—	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRC0	Enable writing to the registers related to the clock generation circuit 0: Disable writes 1: Enable writes	R/W
1	PRC1	Enable writing to the registers related to the low power modes 0: Disable writes 1: Enable writes	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	PRC3	Enable writing to the registers related to the LVD 0: Disable writes 1: Enable writes	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	PRC Key Code These bits control the write access to the PRCR register. To modify the PRCR register, write 0xA5 to the upper 8 bits and the target value to the lower 8 bits as a 16-bit unit.	W

PRCn bits (Protect bit n) (n = 0, 1, 3)

The PRCn bits enable or disable writing to the protected registers listed in Table 11.1. Setting the PRCn bits to 1 or 0 enables or disables writing, respectively.

12. Interrupt Controller Unit (ICU)

12.1 Overview

The Interrupt Controller Unit (ICU) controls which event signals are linked to the Nested Vector Interrupt Controller (NVIC), and the Data Transfer Controller (DTC) modules. The ICU also controls non-maskable interrupts.

Table 12.1 lists the ICU specifications, Figure 12.1 shows a block diagram, and Table 12.2 lists the I/O pins.

Table 12.1 ICU specifications

Item		Description
Maskable interrupts	Peripheral function interrupts	• Interrupts from peripheral modules Number of sources: 108
	External pin interrupts	• Interrupt detection on low level*4, falling edge, rising edge, rising and falling edges. One of these detection methods can be set for each source • Digital filter function supported • 8 sources, with interrupts from IRQi (i = 0 to 7) pins.
	Interrupt requests to CPU (NVIC)	• 32 interrupt requests are output to NVIC. • Maskable interrupt sources are classified into 8 groups, and one source can be selected individually from 31 sources that are classified into groups.
	DTC control	• The DTC can be activated using interrupt sources*1 • The method for selecting an interrupt source is the same as that of the interrupt request to NVIC.
Non-maskable interrupts*2	NMI pin interrupt	• Interrupt from the NMI pin • Interrupt detection on falling edge or rising edge • Digital filter function supported
	WDT underflow/refresh error*3	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	IWDT underflow/refresh error*3	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	Low voltage detection 1*3	Voltage monitor 1 interrupt of the voltage monitor 1 circuit (LVD_LVD1)
	Low voltage detection 2*3	Voltage monitor 2 interrupt of the voltage monitor 2 circuit (LVD_LVD2)
	RPEST	Interrupt on SRAM parity error
	CPU stack pointer monitor error	Interrupt on CPU stack pointer monitor
	Oscillation stop detection interrupt*3	Interrupt on detecting that the main oscillation has stopped
	Bus slave MPU error	Interrupt on bus slave MPU error
	Bus master MPU error	Interrupt on bus master MPU error
Low power modes		• Sleep mode: return is initiated by non-maskable interrupts or any other interrupt source • Software Standby mode: return is initiated by non-maskable interrupts. Interrupt can be selected in the WUPEN register. • Snooze mode: return is initiated by non-maskable interrupts. Interrupt can be selected in the SELSR0 and WUPEN registers. See section 12.2.7. SELSR0 : SYS Event Link Setting Register and section 12.2.8. WUPEN : Wakeup Interrupt Enable Register.

Note 1. For the DTC activation sources, see Table 12.4.
Note 2. Non-maskable interrupts can be enabled only once after a reset release.
Note 3. These non-maskable interrupts can also be used as maskable interrupts. When used as maskable interrupts, do not change the value of the NMIER register from the reset state. To enable voltage monitor 1 and voltage monitor 2 interrupts, set the LVD1CR1.IRQSEL and LVD2CR1.IRQSEL bits to 1.
Note 4. Low level: interrupt detection is not canceled if you do not clear it after a detection.

Figure 12.1 shows the ICU block diagram.

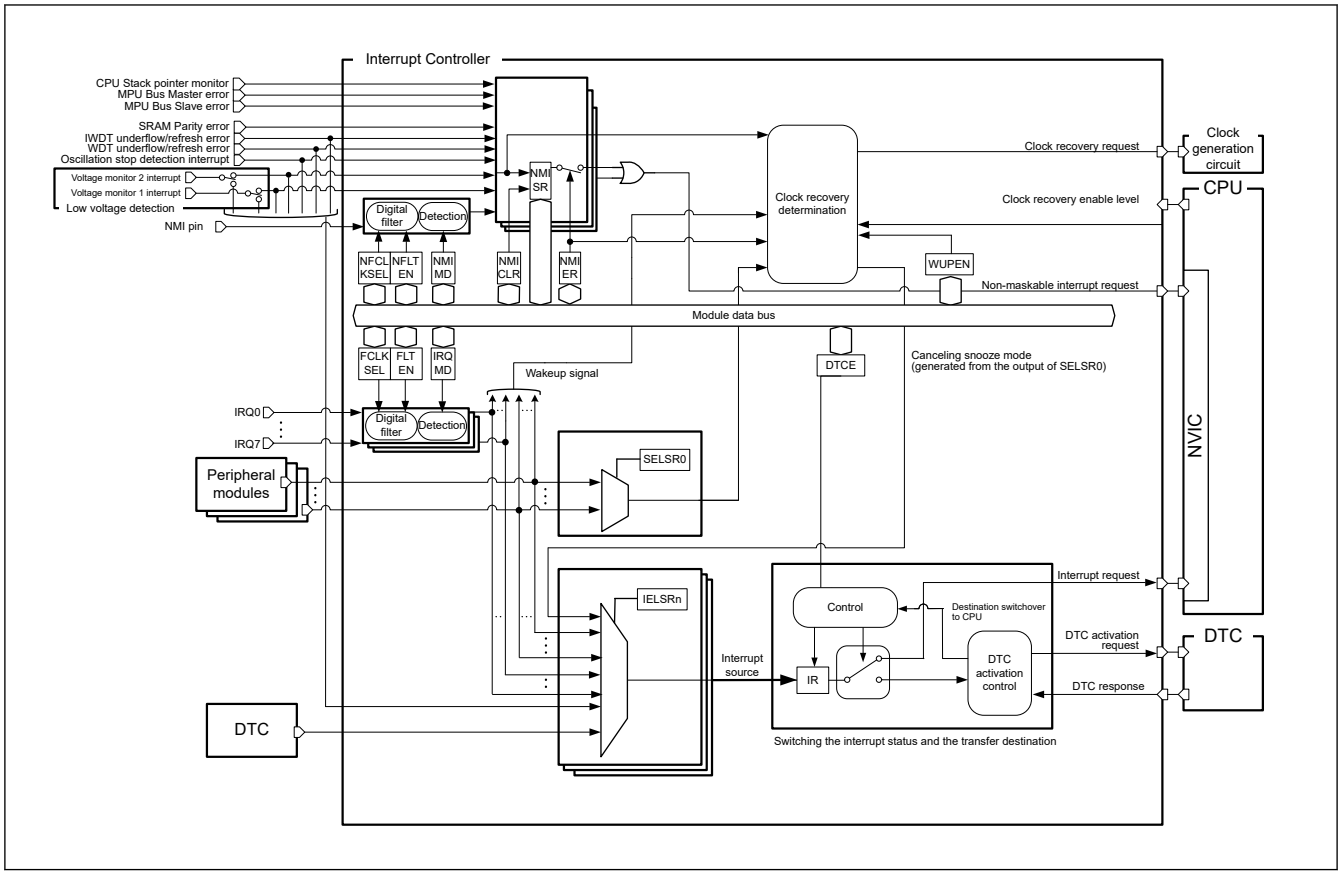


Figure 12.1 ICU block diagram

Table 12.2 lists the ICU input/output pins.

Table 12.2 ICU I/O pins

Pin name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQi (i = 0 to 7)	Input	External interrupt request pins

12.2 Register Descriptions

This chapter does not describe the Arm® NVIC internal registers. For information about these registers, see ARM® Cortex®-M23 Processor Technical Reference Manual (ARM DDI 0550C).

12.2.1 IRQCRi : IRQ Control Register i (i = 0 to 7)

Base address: ICU = 0x4000_6000

Offset address: 0x000 + 0x1 × i

Bit position:	7	6	5	4	3	2	1	0
Bit field:	FLTEN	—	FCLKSEL[1:0]		—	—	IRQMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	IRQMD[1:0]	IRQi Detection Sense Select 0 0: Falling edge 0 1: Rising edge 1 0: Rising and falling edges 1 1: Low level	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
5:4	FCLKSEL[1:0]	IRQi Digital Filter Sampling Clock Select 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	FLTEN	IRQi Digital Filter Enable 0: Digital filter is disabled 1: Digital filter is enabled.	R/W

IRQCRi register changes must satisfy the following conditions:

- For a CPU interrupt or DTC trigger:
  Change the IRQCRi register value before setting the target IELSRn register (n = 0 to 31).
  The register value should be changed only when the value of the target IELSRn register is 0x0000.
- For a wakeup enable signal:
  Change the IRQCRi register setting before setting the target WUPEN.IRQWUPEN[n] (n = 0 to 7). The register value should be changed when the target WUPEN.IRQWUPEN[n] is 0.

IRQMD[1:0] bits (IRQi Detection Sense Select)

The IRQMD[1:0] bits set the detection sensing method for the IRQi external pin interrupt sources. For setting method when using external pin interrupt, see section 12.5.6. External Pin Interrupts.

FCLKSEL[1:0] bits (IRQi Digital Filter Sampling Clock Select)

The FCLKSEL[1:0] bits select the digital filter sampling clock for the IRQi external pin interrupt request pins, selectable to:

- PCLKB (every cycle)
- PCLKB/8 (once every 8 cycles)
- PCLKB/32 (once every 32 cycles)
- PCLKB/64 (once every 64 cycles)

For details of the digital filter, see section 12.5.5. Digital Filter.

FLTEN bit (IRQi Digital Filter Enable)

The FLTEN bit enables the digital filter used for the IRQi external pin interrupt sources. The digital filter is enabled when the IRQCRi.FLTEN bit is 1 and disabled when the IRQCRi.FLTEN bit is 0. The IRQi pin level is sampled at the clock cycle specified in the IRQCRi.FCLKSEL[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes. For details of the digital filter, see section 12.5.5. Digital Filter.

12.2.2 NMISR : Non-Maskable Interrupt Status Register

Base address: ICU = 0x4000_6000

Offset address: 0x140

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SPEST	BUSMST	BUSSST	—	RPEST	NMIST	OSTST	—	—	LVD2ST	LVD1ST	WDTST	IWDTST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTST	IWDT Underflow/Refresh Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
1	WDTST	WDT Underflow/Refresh Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
2	LVD1ST	Voltage Monitor 1 Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
3	LVD2ST	Voltage Monitor 2 Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
5:4	—	These bits are read as 0.	R
6	OSTST	Main Clock Oscillation Stop Detection Interrupt Status Flag 0: Interrupt not requested for main clock oscillation stop 1: Interrupt requested for main clock oscillation stop	R
7	NMIST	NMI Pin Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
8	RPEST	SRAM Parity Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
9	—	This bit is read as 0.	R
10	BUSSST	Bus Slave MPU Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested.	R
11	BUSMST	Bus Master MPU Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
12	SPEST	CPU Stack Pointer Monitor Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
13	—	This bit is read as 0.	R
15:14	—	These bits are read as 0.	R

The NMISR register monitors the status of non-maskable interrupt sources. Writes to the NMISR register are ignored. The setting in the Non-Maskable Interrupt Enable Register (NMIER) does not affect the status flags in this register. Before the end of the non-maskable interrupt handler, check that all of the bits in this register are set to 0 to confirm that no other NMI requests are generated during handler processing.

IWDTST flag (IWDT Underflow/Refresh Error Interrupt Status Flag)

The IWDTST flag indicates an IWDT underflow/refresh error interrupt request. It is read-only and cleared by the NMICLR.IWDTCLR bit.

[Setting condition]

When the IWDT underflow/refresh error interrupt is generated and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.IWDTCLR bit.

WDTST flag (WDT Underflow/Refresh Error Interrupt Status Flag)

The WDTST flag indicates a WDT underflow/refresh error interrupt request. It is read-only and cleared by the NMICLR.WDTCLR bit.

[Setting condition]

When the WDT underflow/refresh error interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.WDTCLR bit.

LVD1ST flag (Voltage Monitor 1 Interrupt Status Flag)

The LVD1ST flag indicates a request for voltage monitor 1 interrupt. It is read-only and cleared by the NMICLR.LVD1CLR bit.

[Setting condition]

When the voltage monitor 1 interrupt is generated and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.LVD1CLR bit.

LVD2ST flag (Voltage Monitor 2 Interrupt Status Flag)

The LVD2ST flag indicates a request for voltage monitor 2 interrupt. It is read-only and cleared by the NMICLR.LVD2CLR bit.

[Setting condition]

When the voltage monitor 2 interrupt is generated and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.LVD2CLR bit.

OSTST flag (Main Clock Oscillation Stop Detection Interrupt Status Flag)

The OSTST flag indicates a main clock oscillation stop detection interrupt request. It is read-only and cleared by the NMICLR.OSTCLR bit.

[Setting condition]

When the main clock oscillation stop detection interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.OSTCLR bit.

NMIST flag (NMI Pin Interrupt Status Flag)

The NMIST flag indicates an NMI pin interrupt request. It is read-only and cleared by the NMICLR.NMICLR bit.

[Setting condition]

When an edge specified by the NMICR.NMIMD bit is input to the NMI pin.

[Clearing condition]

When 1 is written to the NMICLR.NMICLR bit.

RPEST flag (SRAM Parity Error Interrupt Status Flag)

The RPEST flag indicates an SRAM parity error interrupt request.

[Setting condition]

When an interrupt is generated in response to an SRAM parity error.

[Clearing condition]

When 1 is written to the NMICLR.RPECLR bit.

BUSSST flag (Bus Slave MPU Error Interrupt Status Flag)

The BUSSST flag indicates a bus slave error interrupt request.

[Setting condition]

When an interrupt is generated in response to a bus slave error.

[Clearing condition]

When 1 is written to the NMICLR.BUSSCLR bit.

BUSMST flag (Bus Master MPU Error Interrupt Status Flag)

The BUSMST flag indicates a bus master error interrupt request.

[Setting condition]

When an interrupt is generated in response to a bus master error.

[Clearing condition]

When 1 is written to the NMICLR.BUSMCLR bit.

SPEST flag (CPU Stack Pointer Monitor Interrupt Status Flag)

The SPEST flag indicates a CPU stack pointer monitor interrupt request.

[Setting condition]

When an interrupt is generated in response to a CPU stack pointer monitor error.

[Clearing condition]

When 1 is written to the NMICLR.SPECLR bit.

12.2.3 NMIER : Non-Maskable Interrupt Enable Register

Base address: ICU = 0x4000_6000

Offset address: 0x120

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SPEEN	BUSMEN	BUSSEN	—	RPEEN	NMIEN	OSTEN	—	—	LVD2EN	LVD1EN	WDTEN	IWDTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTEN	IWDT Underflow/Refresh Error Interrupt Enable 0: Disabled 1: Enabled.	R/W*1 *2
1	WDTEN	WDT Underflow/Refresh Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
2	LVD1EN	Voltage monitor 1 Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
3	LVD2EN	Voltage monitor 2 Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	OSTEN	Main Clock Oscillation Stop Detection Interrupt Enable 0: Disabled 1: Enabled	R/W*1 *2
7	NMIEN	NMI Pin Interrupt Enable 0: Disabled 1: Enabled	R/W*1
8	RPEEN	SRAM Parity Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1
9	—	This bit is read as 0. The write value should be 0.	R/W
10	BUSSEN	Bus Slave MPU Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1

Bit	Symbol	Function	R/W
11	BUSMEN	Bus Master MPU Error Interrupt Enable 0: Disabled 1: Enabled	R/W*1
12	SPEEN	CPU Stack Pointer Monitor Interrupt Enable 0: Disabled 1: Enabled	R/W*1
13	—	This bit is read as 0. The write value should be 0.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. You can write 1 to this bit only once after reset. Subsequent write accesses are invalid. Writing 0 to this bit is invalid.
Note 2. Do not write 1 to this bit when the source is used as an event signal.

IWDTEN bit (IWDT Underflow/Refresh Error Interrupt Enable)

The IWDTEN bit enables IWDT underflow/refresh error interrupt as an NMI trigger.

WDTEN bit (WDT Underflow/Refresh Error Interrupt Enable)

The WDTEN bit enables WDT underflow/refresh error interrupt as an NMI trigger.

LVD1EN bit (Voltage monitor 1 Interrupt Enable)

The LVD1EN bit enables voltage monitor 1 interrupt as an NMI trigger.

LVD2EN bit (Voltage monitor 2 Interrupt Enable)

The LVD2EN bit enables voltage monitor 2 interrupt as an NMI trigger.

OSTEN bit (Main Clock Oscillation Stop Detection Interrupt Enable)

The OSTEN bit enables main clock oscillation stop detection interrupt as an NMI trigger.

NMIEN bit (NMI Pin Interrupt Enable)

The NMIEN bit enables NMI pin interrupt as an NMI trigger.

RPEEN bit (SRAM Parity Error Interrupt Enable)

The RPEEN bit enables SRAM parity error interrupt as an NMI trigger.

BUSSEN bit (Bus Slave MPU Error Interrupt Enable)

The BUSSEN bit enables bus slave error interrupt as an NMI trigger.

BUSMEN bit (Bus Master MPU Error Interrupt Enable)

The BUSMEN bit enables bus master error interrupt as an NMI trigger.

SPEEN bit (CPU Stack Pointer Monitor Interrupt Enable)

The SPEEN bit enables CPU stack pointer monitor interrupt as an NMI trigger.

12.2.4 NMICLR : Non-Maskable Interrupt Status Clear Register

Base address: ICU = 0x4000_6000

Offset address: 0x130

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SPECLR	BUSMCLR	BUSSCLR	—	RPECLR	NMICLR	OSTCLR	—	—	LVD2CLR	LVD1CLR	WDTCLR	IWDTCLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTCLR	IWDT Underflow/Refresh Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.IWDTST flag	R/W*1
1	WDTCLR	WDT Underflow/Refresh Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.WDTST flag	R/W*1
2	LVD1CLR	Voltage Monitor 1 Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.LVD1ST flag	R/W*1
3	LVD2CLR	Voltage Monitor 2 Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.LVD2ST flag.	R/W*1
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	OSTCLR	Oscillation Stop Detection Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.OSTST flag	R/W*1
7	NMICLR	NMI Pin Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.NMIST flag	R/W*1
8	RPECLR	SRAM Parity Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.RPEST flag	R/W*1
9	—	This bit is read as 0. The write value should be 0.	R/W
10	BUSSCLR	Bus Slave MPU Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.BUSSST flag	R/W*1
11	BUSMCLR	Bus Master MPU Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.BUSMST flag	R/W*1
12	SPECLR	CPU Stack Pointer Monitor Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.SPEST flag	R/W*1
13	—	This bit is read as 0. The write value should be 0.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only write 1 to this bit.

IWDTCLR bit (IWDT Underflow/Refresh Error Interrupt Status Flag Clear)

Writing 1 to the IWDTCLR bit clears the NMISR.IWDTST flag. This bit is read as 0.

WDTCLR bit (WDT Underflow/Refresh Error Interrupt Status Flag Clear)

Writing 1 to the WDTCLR bit clears the NMISR.WDTST flag. This bit is read as 0.

LVD1CLR bit (Voltage Monitor 1 Interrupt Status Flag Clear)

Writing 1 to the LVD1CLR bit clears the NMISR.LVD1ST flag. This bit is read as 0.

LVD2CLR bit (Voltage Monitor 2 Interrupt Status Flag Clear)

Writing 1 to the LVD2CLR bit clears the NMISR.LVD2ST flag. This bit is read as 0.

OSTCLR bit (Oscillation Stop Detection Interrupt Status Flag Clear)

Writing 1 to the OSTCLR bit clears the NMISR.OSTST flag. This bit is read as 0.

NMICLR bit (NMI Pin Interrupt Status Flag Clear)

Writing 1 to the NMICLR bit clears the NMISR.NMIST flag. This bit is read as 0.

RPECLR bit (SRAM Parity Error Interrupt Status Flag Clear)

Writing 1 to the RPECLR bit clears the NMISR.RPEST flag. This bit is read as 0.

BUSSCLR bit (Bus Slave MPU Error Interrupt Status Flag Clear)

Writing 1 to the BUSSCLR bit clears the NMISR.BUSSST flag. This bit is read as 0.

BUSMCLR bit (Bus Master MPU Error Interrupt Status Flag Clear)

Writing 1 to the BUSMCLR bit clears the NMISR.BUSMST flag. This bit is read as 0.

SPECLR bit (CPU Stack Pointer Monitor Interrupt Status Flag Clear)

Writing 1 to the SPECLR bit clears the NMISR.SPEST flag. This bit is read as 0.

12.2.5 NMICR : NMI Pin Interrupt Control Register

Base address: ICU = 0x4000_6000

Offset address: 0x100

Bit position:	7	6	5	4	3	2	1	0
Bit field:	NFLTEN	—	NFCLKSEL[1:0]		—	—	—	NMIMD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NMIMD	NMI Detection Set 0: Falling edge 1: Rising edge	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
5:4	NFCLKSEL[1:0]	NMI Digital Filter Sampling Clock Select 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	NFLTEN	NMI Digital Filter Enable 0: Disabled. 1: Enabled.	R/W

Change the NMICR register settings before enabling NMI pin interrupts, that is, before setting NMIER.NMIEN to 1.

NMIMD bit (NMI Detection Set)

The NMIMD bit selects the detection sensing method for the NMI pin interrupts.

NFCLKSEL[1:0] bits (NMI Digital Filter Sampling Clock Select)

The NFCLKSEL[1:0] bits select the digital filter sampling clock for the NMI pin interrupts, selectable to:

- PCLKB (every cycle)
- PCLKB/8 (once every 8 cycles)
- PCLKB/32 (once every 32 cycles)
- PCLKB/64 (once every 64 cycles)

For details of the digital filter, see section 12.5.5. Digital Filter.

NFLTEN bit (NMI Digital Filter Enable)

The NFLTEN bit enables the digital filter used for NMI pin interrupts. The filter is enabled when NFLTEN is 1, and disabled when NFLTEN is 0. The NMI pin level is sampled at the clock cycle specified in NFCLKSEL[1:0]. When the

sampled level matches three times, the output level from the digital filter changes. For details of the digital filter, see section 12.5.5. Digital Filter.

12.2.6 IELSRn : ICU Event Link Setting Register n (n = 0 to 31)

Base address: ICU = 0x4000_6000

Offset address: 0x300 + 0x4 × n

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

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

Bit	Symbol	Function	R/W
5:0	IELS[5:0]	ICU Event Link Select 0x00: Disable interrupts to the associated NVIC or DTC module Others: Event signal number to be linked. For details, see section 12.3.3. ICU and DTC Event Number.	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
16	IR	Interrupt Status Flag 0: No interrupt request generated. 1: An interrupt request is generated.	R/W*1
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	DTCE	DTC Activation Enable 0: DTC activation is disabled. 1: DTC activation is enabled.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: This register requires halfword or word access.
Note 1. Writing 1 to the IR flag is prohibited.

The IELSRn register selects the IRQi source used by the NVIC. For details, see Table 12.3. IELSRn corresponds to the NVIC IRQ input source number, where n = 0 to 31.

IELS[5:0] bits (ICU Event Link Select)

The IELS[5:0] bits link an event signal to the associated NVIC or DTC module. Event options are classified into 8 groups (groups 0 to 7). For details, see Table 12.3 and Table 12.4.

IR flag (Interrupt Status Flag)

The IR status flag indicates an individual interrupt request from the event specified in IELS[5:0].

[Setting condition]

When an interrupt request is received from the associated peripheral module or IRQi pin.

[Clearing condition]

When 0 is written to the IR flag. DTCE must be set to 0 before writing 0 to the IR flag.

To clear the IR flag:

1. Negate the input interrupt signal.
2. Read access the peripheral once and wait for 2 clock cycles of the target module clock.
3. Clear the IR flag by writing 0.

DTCE bit (DTC Activation Enable)

When the DTCE bit is set to 1, the associated event is selected as the source for DTC activation.

[Setting condition]

- When 1 is written to the DTCE bit.

[Clearing condition]

- When the specified number of transfers is complete. For chain transfers, when the specified number of transfers for the last chain transfer is complete.
- When 0 is written to the DTCE bit.

12.2.7 SELSR0 : SYS Event Link Setting Register

Base address: ICU = 0x4000_6000

Offset address: 0x200

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

Bit	Symbol	Function	R/W
7:0	SELS[7:0]	SYS Event Link Select 0x00: Disable event output to the associated low-power mode module Others: Event signal number to be linked. For details, see Table 12.4.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

The SELSR0 register selects the events that wake up the CPU from Snooze mode. You can use only the events listed in Table 12.4 checked as “Canceling Snooze mode”. When ICU_SNZCANCEL is selected in the IELSRn.IELS[5:0] bits, an interrupt is generated that cancels snooze mode.

12.2.8 WUPEN : Wakeup Interrupt Enable Register

Base address: ICU = 0x4000_6000

Offset address: 0x1A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	IIC0WUPEN	AGT1CBWUPEN	AGT1CAWUPEN	AGT1UDWUPEN	—	—	—	—	—	ACMPHS0WUPEN	—	—	LVD2WUPEN	LVD1WUPEN	KEYWUPEN	IWDTWUPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
7:0	IRQWUPEN[7:0]	IRQ Interrupt Software Standby/Snooze mode Returns Enable 0: Software Standby/Snooze mode returns by IRQn interrupt disabled 1: Software Standby/Snooze mode returns by IRQn interrupt enabled	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	IWDTWUPEN	IWDT Interrupt Software Standby/Snooze mode Returns Enable 0: Software Standby/Snooze mode returns by IWDT interrupt disabled 1: Software Standby/Snooze mode returns by IWDT interrupt enabled	R/W

Bit	Symbol	Function	R/W
17	KEYWUPEN	Key Interrupt Software Standby/Snooze mode Returns Enable 0: Software Standby/Snooze mode returns by KEY interrupt disabled 1: Software Standby/Snooze mode returns by KEY interrupt enabled	R/W
18	LVD1WUPEN	LVD1 Interrupt Software Standby/Snooze mode Returns Enable 0: Software Standby/Snooze mode returns by LVD1 interrupt disabled 1: Software Standby/Snooze mode returns by LVD1 interrupt enabled	R/W
19	LVD2WUPEN	LVD2 Interrupt Software Standby/Snooze mode Returns Enable 0: Software Standby/Snooze mode returns by LVD2 interrupt disabled 1: Software Standby/Snooze mode returns by LVD2 interrupt enabled	R/W
21:20	—	These bits are read as 0. The write value should be 0.	R/W
22	ACMPHS0WUPEN	ACMPHS0 Interrupt Software Standby/Snooze mode Returns Enable 0: Software Standby/Snooze mode returns by ACMPHS0 interrupt disabled 1: Software Standby/Snooze mode returns by ACMPHS0 interrupt enabled	R/W
27:23	—	These bits are read as 0. The write value should be 0.	R/W
28	AGT1UDWUPEN	AGT1 Underflow Interrupt Software Standby/Snooze mode Returns Enable 0: Software Standby/Snooze mode returns by AGT1 underflow interrupt disabled 1: Software Standby/Snooze mode returns by AGT1 underflow	R/W
29	AGT1CAWUPEN	AGT1 Compare Match A Interrupt Software Standby/Snooze mode Returns Enable 0: Software Standby/Snooze mode returns by AGT1 compare match A interrupt disabled. 1: Software Standby/Snooze mode returns by AGT1 compare match A interrupt enabled.	R/W
30	AGT1CBWUPEN	AGT1 Compare Match B Interrupt Software Standby/Snooze mode Returns Enable 0: Software Standby/Snooze mode returns by AGT1 compare match B interrupt disabled. 1: Software Standby/Snooze mode returns by AGT1 compare match B interrupt enabled.	R/W
31	IIC0WUPEN	IIC0 Address Match Interrupt Software Standby/Snooze mode Returns Enable 0: Software Standby/Snooze mode returns by IIC0 address match interrupt disabled 1: Software Standby/Snooze mode returns by IIC0 address match interrupt enabled	R/W

The bits in this register control whether the associated interrupt can wake up the CPU from Software Standby/Snooze mode.

IRQWUPEN[7:0] bits (IRQ Interrupt Software Standby/Snooze mode Returns Enable)

The IRQWUPEN[7:0] bits enable the use of IRQn interrupts to cancel Software Standby/Snooze mode.

IWDTWUPEN bit (IWDT Interrupt Software Standby/Snooze mode Returns Enable)

The IWDTWUPEN bit enables the use of IWDT interrupts to cancel Software Standby/Snooze mode.

KEYWUPEN bit (Key Interrupt Software Standby/Snooze mode Returns Enable)

The KEYWUPEN bit enables the use of key interrupts to cancel Software Standby/Snooze mode.

LVD1WUPEN bit (LVD1 Interrupt Software Standby/Snooze mode Returns Enable)

The LVD1WUPEN bit enables the use of LVD1 interrupts to cancel Software Standby/Snooze mode.

LVD2WUPEN bit (LVD2 Interrupt Software Standby/Snooze mode Returns Enable)

The LVD2WUPEN bit enables the use of LVD2 interrupts to cancel Software Standby/Snooze mode.

ACMPHS0WUPEN bit (ACMPHS0 Interrupt Software Standby/Snooze mode Returns Enable)

The ACMPHS0WUPEN bit enables the use of ACMPHS0 interrupts to cancel Software Standby/Snooze mode.

AGT1UDWUPEN bit (AGT1 Underflow Interrupt Software Standby/Snooze mode Returns Enable)

The AGT1UDWUPEN bit enables the use of the AGT1 underflow interrupts to cancel Software Standby/Snooze mode.

AGT1CAWUPEN bit (AGT1 Compare Match A Interrupt Software Standby/Snooze mode Returns Enable)

The AGT1CAWUPEN bit enables the use of AGT1 compare match A interrupts to cancel Software Standby/Snooze mode.

AGT1CBWUPEN bit (AGT1 Compare Match B Interrupt Software Standby/Snooze mode Returns Enable)

The AGT1CBWUPEN bit enables the use of AGT1 compare match B interrupts to cancel Software Standby/Snooze mode.

IIC0WUPEN bit (IIC0 Address Match Interrupt Software Standby/Snooze mode Returns Enable)

The IIC0WUPEN bit enables the use of IIC0 interrupts to cancel Software Standby/Snooze mode.

12.2.9 IELEN : ICU Event Enable Register

Base address: ICU = 0x4000_6000

Offset address: 0x1C0

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	IELEN	Parts Asynchronous Interrupts Enable 0: Disabled 1: Enabled	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

The bits in this register control whether the associated interrupt can be used.

IELEN bit (Parts Asynchronous Interrupts Enable)

The IELEN bit enables the use of parts asynchronous interrupts as follows:

- IIC0_WUI, AGT1_AGTCMBI, AGT1_AGTCMAI, AGT1_AGTI, ACMPHS0, LVD_LVD1, LVD_LVD2, KEY_INTKR, IWDT_NMIUNDF, PORT_IRQ0 to PORT_IRQ7

12.3 Vector Table

The ICU detects maskable and non-maskable interrupts. Interrupt priorities are set up in the Arm NVIC. For information about these registers, see section 12.9. Reference.

12.3.1 Interrupt Vector Table

Table 12.3 describes the interrupt vector table. The interrupt vector addresses conform to the NVIC specifications.

Table 12.3 Interrupt vector table (1 of 2)

Exception number	IRQ number	Vector offset	Source	Description
0	—	0x000	Arm	Initial stack pointer
1	—	0x004	Arm	Initial program counter (reset vector)
2	—	0x008	Arm	Non-Maskable Interrupt (NMI)
3	—	0x00C	Arm	Hard Fault
4	—	0x010	Arm	Reserved
5	—	0x014	Arm	Reserved
6	—	0x018	Arm	Reserved
7	—	0x01C	Arm	Reserved
8	—	0x020	Arm	Reserved
9	—	0x024	Arm	Reserved

Table 12.3 Interrupt vector table (2 of 2)

Exception number	IRQ number	Vector offset	Source	Description
10	—	0x028	Arm	Reserved
11	—	0x02C	Arm	Supervisor Call (SVCall)
12	—	0x030	Arm	Reserved
13	—	0x034	Arm	Reserved
14	—	0x038	Arm	Pendable request for system service (PendableSrvReq)
15	—	0x03C	Arm	System Tick Timer (SysTick)
16	0	0x040	ICU.IELSR0	Event selected in the ICU.IELSR0 register
17	1	0x044	ICU.IELSR1	Event selected in the ICU.IELSR1 register
18	2	0x048	ICU.IELSR2	Event selected in the ICU.IELSR2 register
19	3	0x04C	ICU.IELSR3	Event selected in the ICU.IELSR3 register
20	4	0x050	ICU.IELSR4	Event selected in the ICU.IELSR4 register
21	5	0x054	ICU.IELSR5	Event selected in the ICU.IELSR5 register
22	6	0x058	ICU.IELSR6	Event selected in the ICU.IELSR6 register
23	7	0x05C	ICU.IELSR7	Event selected in the ICU.IELSR7 register
24	8	0x060	ICU.IELSR8	Event selected in the ICU.IELSR8 register
25	9	0x064	ICU.IELSR9	Event selected in the ICU.IELSR9 register
26	10	0x068	ICU.IELSR10	Event selected in the ICU.IELSR10 register
27	11	0x06C	ICU.IELSR11	Event selected in the ICU.IELSR11 register
28	12	0x070	ICU.IELSR12	Event selected in the ICU.IELSR12 register
29	13	0x074	ICU.IELSR13	Event selected in the ICU.IELSR13 register
30	14	0x078	ICU.IELSR14	Event selected in the ICU.IELSR14 register
31	15	0x07C	ICU.IELSR15	Event selected in the ICU.IELSR15 register
32	16	0x080	ICU.IELSR16	Event selected in the ICU.IELSR16 register
33	17	0x084	ICU.IELSR17	Event selected in the ICU.IELSR17 register
34	18	0x088	ICU.IELSR18	Event selected in the ICU.IELSR18 register
35	19	0x08C	ICU.IELSR19	Event selected in the ICU.IELSR19 register
36	20	0x090	ICU.IELSR20	Event selected in the ICU.IELSR20 register
37	21	0x094	ICU.IELSR21	Event selected in the ICU.IELSR21 register
38	22	0x098	ICU.IELSR22	Event selected in the ICU.IELSR22 register
39	23	0x09C	ICU.IELSR23	Event selected in the ICU.IELSR23 register
40	24	0x0A0	ICU.IELSR24	Event selected in the ICU.IELSR24 register
41	25	0x0A4	ICU.IELSR25	Event selected in the ICU.IELSR25 register
42	26	0x0A8	ICU.IELSR26	Event selected in the ICU.IELSR26 register
43	27	0x0AC	ICU.IELSR27	Event selected in the ICU.IELSR27 register
44	28	0x0B0	ICU.IELSR28	Event selected in the ICU.IELSR28 register
45	29	0x0B4	ICU.IELSR29	Event selected in the ICU.IELSR29 register
46	30	0x0B8	ICU.IELSR30	Event selected in the ICU.IELSR30 register
47	31	0x0BC	ICU.IELSR31	Event selected in the ICU.IELSR31 register

12.3.2 Event Number

The following table lists heading details for Table 12.4, which describes each event number.

Heading	Description
Interrupt request source	Name of the source generating the interrupt request
Name	Name of the interrupt
Connect to NVIC	" ✓" indicates the interrupt can be used as a CPU interrupt
Invoke DTC	" ✓" indicates the interrupt can be used to request DTC activation
Canceling Snooze mode	" ✓" indicates the interrupt can be used to request a return from Snooze mode
Canceling Software Standby mode	" ✓" indicates the interrupt can be used to request a return from Software Standby mode

Table 12.4 Event table (1 of 4)

Event number*5	Interrupt request source	Name	IELSRn		Canceling Snooze	Canceling Software Standby
			Connect to NVIC	Invoke DTC		
0x01	Port	PORT_IRQ0	✓	✓	✓	✓
0x02		PORT_IRQ1	✓	✓	✓	✓
0x03		PORT_IRQ2	✓	✓	✓	✓
0x04		PORT_IRQ3	✓	✓	✓	✓
0x05		PORT_IRQ4	✓	✓	✓	✓
0x06		PORT_IRQ5	✓	✓	✓	✓
0x07		PORT_IRQ6	✓	✓	✓	✓
0x08		PORT_IRQ7	✓	✓	✓	✓
0x09	DTC	DTC_COMPLETE	✓	—	✓*4	—
0x0B	ICU	ICU_SNZCANCEL	✓	—	✓	—
0x0C	FLASH	FCU_FRDYI	✓	—	—	—
0x0D	LVD	LVD_LVD1	✓	—	✓	✓
0x0E		LVD_LVD2	✓	—	✓	✓
0x0F	MOSC	MOSC_STOP	✓	—	—	—
0x10	Low power mode	SYSTEM_SNZREQ	—	✓	—	—
0x11	AGT0	AGT0_AGTI	✓	✓	—	—
0x12		AGT0_AGTCMAI	✓	✓	—	—
0x13		AGT0_AGTCMBI	✓	✓	—	—
0x14	AGT1	AGT1_AGTI	✓	✓	✓	✓
0x15		AGT1_AGTCMAI	✓	✓	✓	✓
0x16		AGT1_AGTCMBI	✓	✓	✓	✓
0x17	IWDT	IWDT_NMIUNDF	✓	—	✓	✓
0x18	WDT	WDT_NMIUNDF	✓	—	—	—
0x1C	ADC12	ADC120_ADI	✓	✓	—	—
0x1D		ADC120_GBADI	✓	✓	—	—
0x1E		ADC120_CMPAI	✓	—	—	—
0x1F		ADC120_CMPBI	✓	—	—	—
0x20		ADC120_WCMPM	—	✓	✓*4	—
0x21		ADC120_WCMPUM	—	✓	✓*4	—
0x22	ACMPHS	ACMP_HS0	✓	—	✓	✓
0x23		ACMP_HS1	✓	—	—	—

Table 12.4 Event table (2 of 4)

Event number[*5]	Interrupt request source	Name	IELSRn		Canceling Snooze	Canceling Software Standby
			Connect to NVIC	Invoke DTC		
0x27	IIC0	IIC0_RXI	✓	✓	—	—
0x28		IIC0_TXI	✓	✓	—	—
0x29		IIC0_TEI	✓	—	—	—
0x2A		IIC0_EEI	✓	—	—	—
0x2B		IIC0_WUI	✓	—	✓	✓
0x33	KINT	KEY_INTKR	✓	—	✓[*1]	✓[*1]
0x34	DOC	DOC_DOPCI	✓	—	✓[*4]	—
0x35	CAC	CAC_FERRI	✓	—	—	—
0x36		CAC_MENDI	✓	—	—	—
0x37		CAC_OVFI	✓	—	—	—
0x3D	I/O Ports	IOPORT_GROUP1	✓	✓[*2]	—	—
0x3E		IOPORT_GROUP2	✓	✓[*2]	—	—
0x3F	ELC	ELC_SWEVT0	✓[*3]	✓	—	—
0x40		ELC_SWEVT1	✓[*3]	✓	—	—
0x41	POEG	POEG_GROUPA	✓	—	—	—
0x42		POEG_GROUPB	✓	—	—	—
0x46	GPT160	GPT0_CCMPA	✓	✓	—	—
0x47		GPT0_CCMPB	✓	✓	—	—
0x48		GPT0_CMPC	✓	✓	—	—
0x49		GPT0_CMPD	✓	✓	—	—
0x4A		GPT0_CMPE	✓	✓	—	—
0x4B		GPT0_CMPF	✓	✓	—	—
0x4C		GPT0_OVF	✓	✓	—	—
0x4D		GPT0_UDF	✓	✓	—	—
0x4E		GPT0_PC	✓	✓	—	—
0x4F		GPT0_ADTRGA	✓	✓	—	—
0x50		GPT0_ADTRGB	✓	✓	—	—
0x51	GPT161	GPT1_CCMPA	✓	✓	—	—
0x52		GPT1_CCMPB	✓	✓	—	—
0x53		GPT1_CMPC	✓	✓	—	—
0x54		GPT1_CMPD	✓	✓	—	—
0x55		GPT1_CMPE	✓	✓	—	—
0x56		GPT1_CMPF	✓	✓	—	—
0x57		GPT1_OVF	✓	✓	—	—
0x58		GPT1_UDF	✓	✓	—	—
0x59		GPT1_PC	✓	✓	—	—
0x5A		GPT1_ADTRGA	✓	✓	—	—
0x5B		GPT1_ADTRGB	✓	✓	—	—

Table 12.4 Event table (3 of 4)

Event number[*5]	Interrupt request source	Name	IELSRn		Canceling Snooze	Canceling Software Standby
			Connect to NVIC	Invoke DTC		
0x5C	GPT162	GPT2_CCMPA	✓	✓	—	—
0x5D		GPT2_CCMPB	✓	✓	—	—
0x5E		GPT2_CMPC	✓	✓	—	—
0x5F		GPT2_CMPD	✓	✓	—	—
0x60		GPT2_CMPE	✓	✓	—	—
0x61		GPT2_CMPF	✓	✓	—	—
0x62		GPT2_OVF	✓	✓	—	—
0x63		GPT2_UDF	✓	✓	—	—
0x64		GPT2_PC	✓	✓	—	—
0x65		GPT2_ADTRGA	✓	✓	—	—
0x66		GPT2_ADTRGB	✓	✓	—	—
0x70	GPT	GPT_UVWEDGE	✓	—	—	—
0x71	SCI0	SCI0_RXI	✓	✓	—	—
0x72		SCI0_TXI	✓	✓	—	—
0x73		SCI0_TEI	✓	—	—	—
0x74		SCI0_ERI	✓	—	—	—
0x75		SCI0_AM	✓	—	✓[*4]	—
0x76		SCI0_RXI_OR_ERI	—	—	✓[*4]	—
0x77	SCI1	SCI1_RXI	✓	✓	—	—
0x78		SCI1_TXI	✓	✓	—	—
0x79		SCI1_TEI	✓	—	—	—
0x7A		SCI1_ERI	✓	—	—	—
0x7B		SCI1_AM	✓	—	—	—
0x7C	SCI9	SCI9_RXI	✓	✓	—	—
0x7D		SCI9_TXI	✓	✓	—	—
0x7E		SCI9_TEI	✓	—	—	—
0x7F		SCI9_ERI	✓	—	—	—
0x80		SCI9_AM	✓	—	—	—
0x81	SPI0	SPI0_SPRI	✓	✓	—	—
0x82		SPI0_SPTI	✓	✓	—	—
0x83		SPI0_SPII	✓	—	—	—
0x84		SPI0_SPEI	✓	—	—	—
0x85		SPI0_SPTEND	✓	—	—	—
0x8E	SCI2	SCI2_RXI	✓	✓	—	—
0x8F		SCI2_TXI	✓	✓	—	—
0x90		SCI2_TEI	✓	—	—	—
0x91		SCI2_ERI	✓	—	—	—
0x92		SCI2_AM	✓	—	—	—

Table 12.4 Event table (4 of 4)

Event number*5	Interrupt request source	Name	IELSRn		Canceling Snooze	Canceling Software Standby
			Connect to NVIC	Invoke DTC		
0x98	GPT163	GPT3_CCMPA	✓	✓	—	—
0x99		GPT3_CCMPB	✓	✓	—	—
0x9A		GPT3_CMPC	✓	✓	—	—
0x9B		GPT3_CMPD	✓	✓	—	—
0x9C		GPT3_CMPE	✓	✓	—	—
0x9D		GPT3_CMPF	✓	✓	—	—
0x9E		GPT3_OVF	✓	✓	—	—
0x9F		GPT3_UDF	✓	✓	—	—
0xA0		GPT3_PC	✓	✓	—	—
0xA1		GPT3_ADTRGA	✓	✓	—	—
0xA2		GPT3_ADTRGB	✓	✓	—	—

Note 1. Only supported when KRCTL.KRMD is 1.
Note 2. Only the first edge detection is valid.
Note 3. Only interrupts after DTC transfer are supported.
Note 4. Using SELSR0.
Note 5. Event number is active only in Canceling Snooze mode and Canceling Software Standby mode. The setting of CPU and DTC interrupts, see Table 12.7.

12.3.3 ICU and DTC Event Number

Table 12.5, Table 12.6 indicates the IELSRn.IELS[5:0] set values on the CPU interrupt or on DTC activation request. Table 12.7 indicates register set values of each event selection.

Table 12.5 ICU input link select (1) (1 of 2)

IELS[5:0]	group0 (interrupt ch IELSR0/8/16/24)	group1 (interrupt ch IELSR1/9/17/25)	group2 (interrupt ch IELSR2/10/18/26)	group3 (interrupt ch IELSR3/11/19/27)
0x00	Interrupt disable	Interrupt disable	Interrupt disable	Interrupt disable
0x01	PORT_IRQ0	PORT_IRQ1	PORT_IRQ2	PORT_IRQ3
0x02	DTC_COMPLETE	LVD_LVD2	FCU_FRDYI	SYSTEM_SNZREQ
0x03	ICU_SNZCANCEL	AGT1_AGTCMAI	AGT1_AGTCMBI	IWDT_NMIUNDF
0x04	LVD_LVD1	Setting prohibited	Setting prohibited	Setting prohibited
0x05	AGT1_AGTI	ADC120_GBADI	ADC120_CMPAI	ADC120_CMPBI
0x06	WDT_NMIUNDF	ADC120_WCMPUM	IIC0_TEI	IIC0_EEI
0x07	ADC120_ADI	ACMP_HS1	Setting prohibited	Setting prohibited
0x08	ADC120_WCMPM	IIC0_TXI	CAC_MENDI	CAC_OVFI
0x09	ACMP_HS0	Setting prohibited	Setting prohibited	Setting prohibited
0x0A	IIC0_RXI	DOC_DOPCI	ELC_SWEVT0	ELC_SWEVT1
0x0B	IIC0_WUI	CAC_FERRI	POEG_GROUPA	POEG_GROUPB
0x0C	Setting prohibited	Setting prohibited	GPT0_CMPC	GPT0_CMPD
0x0D	Setting prohibited	GPT0_CCMPB	GPT2_CMPC	GPT2_CMPD
0x0E	GPT0_CCMPA	GPT0_UDF	GPT2_OVF	GPT2_UDF
0x0F	GPT0_OVF	GPT2_CCMPB	SCI0_TEI	SCI0_ERI
0x10	GPT2_CCMPA	SCI0_TXI	SPI0_SPII	SPI0_SPEI
0x11	GPT_UVWEDGE	SPI0_SPTI	SPI0_SPTEND	AGT0_AGTI

Table 12.5 ICU input link select (1) (2 of 2)

IELS[5:0]	group0 (interrupt ch IELSR0/8/16/24)	group1 (interrupt ch IELSR1/9/17/25)	group2 (interrupt ch IELSR2/10/18/26)	group3 (interrupt ch IELSR3/11/19/27)
0x12	SCI0_RXI	Setting prohibited	Setting prohibited	GPT1_CMPD
0x13	SCI0_AM	AGT0_AGTCMBI	IOPORT_GROUP2	Setting prohibited
0x14	SPI0_SPRI	Setting prohibited	GPT1_CMPC	Setting prohibited
0x15	Setting prohibited	IOPORT_GROUP1	Setting prohibited	Setting prohibited
0x16	AGT0_AGTCMAI	GPT1_CCMPB	Setting prohibited	Setting prohibited
0x17	Setting prohibited	GPT3_UDF	Setting prohibited	Setting prohibited
0x18	KEY_INTKR	Setting prohibited	Setting prohibited	Setting prohibited
0x19	GPT1_CCMPA	Setting prohibited	Setting prohibited	SCI1_ERI
0x1A	GPT3_OVF	Setting prohibited	Setting prohibited	Setting prohibited
0x1B	Setting prohibited	SCI1_TXI	SCI1_TEI	SCI9_AM
0x1C	Setting prohibited	SCI2_AM	Setting prohibited	Setting prohibited
0x1D	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x1E	SCI1_RXI	Setting prohibited	Setting prohibited	Setting prohibited
0x1F	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x20	GPT0_ADTRGA	GPT0_ADTRGB	GPT0_CMPE	GPT0_CMPF
0x21	GPT1_ADTRGA	GPT1_ADTRGB	GPT0_PC	Setting prohibited
0x22	GPT2_CMPE	GPT2_CMPF	GPT1_PC	Setting prohibited
0x23	GPT2_PC	Setting prohibited	GPT2_ADTRGA	GPT2_ADTRGB
0x24	Setting prohibited	Setting prohibited	GPT3_CMPE	GPT3_CMPF

Table 12.6 ICU input link select (2) (1 of 2)

IELS[5:0]	group4 (interrupt ch IELSR4/12/20/28)	group5 (interrupt ch IELSR5/13/21/29)	group6 (interrupt ch IELSR6/14/22/30)	group7 (interrupt ch IELSR 7/15/23/31)
0x00	Interrupt disable	Interrupt disable	Interrupt disable	Interrupt disable
0x01	PORT_IRQ0	PORT_IRQ1	PORT_IRQ2	PORT_IRQ3
0x02	DTC_COMPLETE	LVD_LVD2	FCU_FRDYI	SYSTEM_SNZREQ
0x03	ICU_SNZCANCEL	AGT1_AGTCMAI	AGT1_AGTCMBI	IWDT_NMIUNDF
0x04	LVD_LVD1	Setting prohibited	Setting prohibited	Setting prohibited
0x05	AGT1_AGTI	ADC120_GBADI	ADC120_CMPAI	ADC120_CMPBI
0x06	WDT_NMIUNDF	ADC120_WCMPUM	IIC0_TEI	IIC0_EEI
0x07	ADC120_ADI	ACMP_HS1	Setting prohibited	Setting prohibited
0x08	ADC120_WCMPM	IIC0_TXI	CAC_MENDI	CAC_OVFI
0x09	ACMP_HS0	Setting prohibited	Setting prohibited	Setting prohibited
0x0A	IIC0_RXI	DOC_DOPCI	ELC_SWEVT0	ELC_SWEVT1
0x0B	IIC0_WUI	CAC_FERRI	POEG_GROUPA	POEG_GROUPB
0x0C	Setting prohibited	Setting prohibited	GPT0_CMPC	GPT0_CMPD
0x0D	Setting prohibited	GPT0_CCMPB	GPT2_CMPC	GPT2_CMPD
0x0E	GPT0_CCMPA	GPT0_UDF	GPT2_OVF	GPT2_UDF
0x0F	GPT0_OVF	GPT2_CCMPB	SCI0_TEI	SCI0_ERI
0x10	GPT2_CCMPA	SCI0_TXI	SPI0_SPII	SPI0_SPEI
0x11	GPT_UVWEDGE	SPI0_SPTI	SPI0_SPTEND	PORT_IRQ7
0x12	SCI0_RXI	Setting prohibited	Setting prohibited	GPT3_CMPD

Table 12.6 ICU input link select (2) (2 of 2)

IELS[5:0]	group4 (interrupt ch IELSR4/12/20/28)	group5 (interrupt ch IELSR5/13/21/29)	group6 (interrupt ch IELSR6/14/22/30)	group7 (interrupt ch IELSR 7/15/23/31)
0x13	SCI0_AM	PORT_IRQ5	PORT_IRQ6	Setting prohibited
0x14	SPI0_SPRI	Setting prohibited	MOSC_STOP	Setting prohibited
0x15	Setting prohibited	GPT1_UDF	GPT3_CMPC	Setting prohibited
0x16	PORT_IRQ4	GPT3_CCMPB	Setting prohibited	Setting prohibited
0x17	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x18	GPT1_OVF	Setting prohibited	Setting prohibited	Setting prohibited
0x19	GPT3_CCMPA	Setting prohibited	Setting prohibited	SCI2_ERI
0x1A	Setting prohibited	SCI1_AM	Setting prohibited	SCI9_ERI
0x1B	Setting prohibited	SCI2_TXI	Setting prohibited	Setting prohibited
0x1C	Setting prohibited	SCI9_TXI	SCI2_TEI	Setting prohibited
0x1D	SCI2_RXI	Setting prohibited	Setting prohibited	Setting prohibited
0x1E	SCI9_RXI	Setting prohibited	SCI9_TEI	Setting prohibited
0x1F	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0x20	GPT0_ADTRGA	GPT0_ADTRGB	GPT0_CMPE	GPT0_CMPF
0x21	GPT3_ADTRGA	GPT3_ADTRGB	GPT0_PC	Setting prohibited
0x22	GPT2_CMPE	GPT2_CMPF	GPT1_CMPE	GPT1_CMPF
0x23	GPT2_PC	Setting prohibited	GPT2_ADTRGA	GPT2_ADTRGB
0x24	Setting prohibited	Setting prohibited	GPT3_PC	Setting prohibited

Table 12.7 Register setting for event (1 of 4)

Name	IELSRn.IELS[5:0]							
	Group 0 (n = 0/8/16/24)	Group 1 (n = 1/9/17/25)	Group 2 (n = 2/10/18/26)	Group 3 (n = 3/11/19/27)	Group 4 (n = 4/12/20/28)	Group 5 (n = 5/13/21/29)	Group 6 (n = 6/14/22/30)	Group 7 (n = 7/15/23/31)
PORT_IRQ0	0x01	—	—	—	0x01	—	—	—
PORT_IRQ1	—	0x01	—	—	—	0x01	—	—
PORT_IRQ2	—	—	0x01	—	—	—	0x01	—
PORT_IRQ3	—	—	—	0x01	—	—	—	0x01
PORT_IRQ4	—	—	—	—	0x16	—	—	—
PORT_IRQ5	—	—	—	—	—	0x13	—	—
PORT_IRQ6	—	—	—	—	—	—	0x13	—
PORT_IRQ7	—	—	—	—	—	—	—	0x11
DTC_COMPLETE	0x02*1	—	—	—	0x02*1	—	—	—
ICU_SNZCANCEL	0x03*1	—	—	—	0x03*1	—	—	—
FCU_FRDYI	—	—	0x02*1	—	—	—	0x02*1	—
LVD_LVD1	0x04*1	—	—	—	0x04*1	—	—	—
LVD_LVD2	—	0x02*1	—	—	—	0x02*1	—	—
MOSC_STOP	—	—	—	—	—	—	0x14*1	—
SYSTEM_SNZREQ	—	—	—	0x02*2	—	—	—	0x02*2
AGT0_AGTI	—	—	—	0x11	—	—	—	—
AGT0_AGTCMAI	0x16	—	—	—	—	—	—	—
AGT0_AGTCMBI	—	0x13	—	—	—	—	—	—

Table 12.7 Register setting for event (2 of 4)

Name	IELSRn.IELS[5:0]							
	Group 0 (n = 0/8/16/24)	Group 1 (n = 1/9/17/25)	Group 2 (n = 2/10/18/26)	Group 3 (n = 3/11/19/27)	Group 4 (n = 4/12/20/28)	Group 5 (n = 5/13/21/29)	Group 6 (n = 6/14/22/30)	Group 7 (n = 7/15/23/31)
AGT1_AGTI	0x05	—	—	—	0x05	—	—	—
AGT1_AGTCMAI	—	0x03	—	—	—	0x03	—	—
AGT1_AGTCMBI	—	—	0x03	—	—	—	0x03	—
IWDT_NMIUNDF	—	—	—	0x03*1	—	—	—	0x03*1
WDT_NMIUNDF	0x06*1	—	—	—	0x06*1	—	—	—
ADC120_ADI	0x07	—	—	—	0x07	—	—	—
ADC120_GBADI	—	0x05	—	—	—	0x05	—	—
ADC120_CMPAI	—	—	0x05*1	—	—	—	0x05*1	—
ADC120_CMPBI	—	—	—	0x05*1	—	—	—	0x05*1
ADC120_WCMPM	0x08*2	—	—	—	0x08*2	—	—	—
ADC120_WCMPUM	—	0x06*2	—	—	—	0x06*2	—	—
ACMP_HS0	0x09*1	—	—	—	0x09*1	—	—	—
ACMP_HS1	—	0x07*1	—	—	—	0x07*1	—	—
IIC0_RXI	0x0A	—	—	—	0x0A	—	—	—
IIC0_TXI	—	0x08	—	—	—	0x08	—	—
IIC0_TEI	—	—	0x06*1	—	—	—	0x06*1	—
IIC0_EEI	—	—	—	0x06*1	—	—	—	0x06*1
IIC0_WUI	0x0B*1	—	—	—	0x0B*1	—	—	—
KEY_INTKR	0x18*1	—	—	—	—	—	—	—
DOC_DOPCI	—	0x0A*1	—	—	—	0x0A*1	—	—
CAC_FERRI	—	0x0B*1	—	—	—	0x0B*1	—	—
CAC_MENDI	—	—	0x08*1	—	—	—	0x08*1	—
CAC_OVFI	—	—	—	0x08*1	—	—	—	0x08*1
IOPORT_GROUP1	—	0x15	—	—	—	—	—	—
IOPORT_GROUP2	—	—	0x13	—	—	—	—	—
ELC_SWEVT0	—	—	0x0A	—	—	—	0x0A	—
ELC_SWEVT1	—	—	—	0x0A	—	—	—	0x0A
POEG_GROUPA	—	—	0x0B*1	—	—	—	0x0B*1	—
POEG_GROUPB	—	—	—	0x0B*1	—	—	—	0x0B*1
GPT0_CCMPA	0x0E	—	—	—	0x0E	—	—	—
GPT0_CCMPB	—	0x0D	—	—	—	0x0D	—	—
GPT0_CMPC	—	—	0x0C	—	—	—	0x0C	—
GPT0_CMPD	—	—	—	0x0C	—	—	—	0x0C
GPT0_CMPE	—	—	0x20	—	—	—	0x20	—
GPT0_CMPF	—	—	—	0x20	—	—	—	0x20
GPT0_OVF	0x0F	—	—	—	0x0F	—	—	—
GPT0_UDF	—	0x0E	—	—	—	0x0E	—	—
GPT0_PC	—	—	0x21	—	—	—	0x21	—
GPT0_ADTRGA	0x20	—	—	—	0x20	—	—	—

Table 12.7 Register setting for event (3 of 4)

Name	IELSRn.IELS[5:0]							
	Group 0 (n = 0/8/16/24)	Group 1 (n = 1/9/17/25)	Group 2 (n = 2/10/18/26)	Group 3 (n = 3/11/19/27)	Group 4 (n = 4/12/20/28)	Group 5 (n = 5/13/21/29)	Group 6 (n = 6/14/22/30)	Group 7 (n = 7/15/23/31)
GPT0_ADTRGB	—	0x20	—	—	—	0x20	—	—
GPT1_CCMPA	0x19	—	—	—	—	—	—	—
GPT1_CCMPB	—	0x16	—	—	—	—	—	—
GPT1_CMPC	—	—	0x14	—	—	—	—	—
GPT1_CMPD	—	—	—	0x12	—	—	—	—
GPT1_CMPE	—	—	—	—	—	—	0x22	—
GPT1_CMPF	—	—	—	—	—	—	—	0x22
GPT1_OVF	—	—	—	—	0x18	—	—	—
GPT1_UDF	—	—	—	—	—	0x15	—	—
GPT1_PC	—	—	0x22	—	—	—	—	—
GPT1_ADTRGA	0x21	—	—	—	—	—	—	—
GPT1_ADTRGB	—	0x21	—	—	—	—	—	—
GPT2_CCMPA	0x10	—	—	—	0x10	—	—	—
GPT2_CCMPB	—	0x0F	—	—	—	0x0F	—	—
GPT2_CMPC	—	—	0x0D	—	—	—	0x0D	—
GPT2_CMPD	—	—	—	0x0D	—	—	—	0x0D
GPT2_CMPE	0x22	—	—	—	0x22	—	—	—
GPT2_CMPF	—	0x22	—	—	—	0x22	—	—
GPT2_OVF	—	—	0x0E	—	—	—	0x0E	—
GPT2_UDF	—	—	—	0x0E	—	—	—	0x0E
GPT2_PC	0x23	—	—	—	0x23	—	—	—
GPT2_ADTRGA	—	—	0x23	—	—	—	0x23	—
GPT2_ADTRGB	—	—	—	0x23	—	—	—	0x23
GPT3_CCMPA	—	—	—	—	0x19	—	—	—
GPT3_CCMPB	—	—	—	—	—	0x16	—	—
GPT3_CMPC	—	—	—	—	—	—	0x15	—
GPT3_CMPD	—	—	—	—	—	—	—	0x12
GPT3_CMPE	—	—	0x24	—	—	—	—	—
GPT3_CMPF	—	—	—	0x24	—	—	—	—
GPT3_OVF	0x1A	—	—	—	—	—	—	—
GPT3_UDF	—	0x17	—	—	—	—	—	—
GPT3_PC	—	—	—	—	—	—	0x24	—
GPT3_ADTRGA	—	—	—	—	0x21	—	—	—
GPT3_ADTRGB	—	—	—	—	—	0x21	—	—
GPT_UVWEDGE	0x11*1	—	—	—	0x11*1	—	—	—
SCI0_RXI	0x12	—	—	—	0x12	—	—	—
SCI0_TXI	—	0x10	—	—	—	0x10	—	—
SCI0_TEI	—	—	0x0F*1	—	—	—	0x0F*1	—
SCI0_ERI	—	—	—	0x0F*1	—	—	—	0x0F*1
SCI0_AM	0x13*1	—	—	—	0x13*1	—	—	—

Table 12.7 Register setting for event (4 of 4)

Name	IELSRn.IELS[5:0]							
	Group 0 (n = 0/8/16/24)	Group 1 (n = 1/9/17/25)	Group 2 (n = 2/10/18/26)	Group 3 (n = 3/11/19/27)	Group 4 (n = 4/12/20/28)	Group 5 (n = 5/13/21/29)	Group 6 (n = 6/14/22/30)	Group 7 (n = 7/15/23/31)
SCI1_RXI	0x1E	—	—	—	—	—	—	—
SCI1_TXI	—	0x1B	—	—	—	—	—	—
SCI1_TEI	—	—	0x1B*1	—	—	—	—	—
SCI1_ERI	—	—	—	0x19*1	—	—	—	—
SCI1_AM	—	—	—	—	—	0x1A*1	—	—
SCI2_RXI	—	—	—	—	0x1D	—	—	—
SCI2_TXI	—	—	—	—	—	0x1B	—	—
SCI2_TEI	—	—	—	—	—	—	0x1C*1	—
SCI2_ERI	—	—	—	—	—	—	—	0x19*1
SCI2_AM	—	0x1C*1	—	—	—	—	—	—
SCI9_RXI	—	—	—	—	0x1E	—	—	—
SCI9_TXI	—	—	—	—	—	0x1C	—	—
SCI9_TEI	—	—	—	—	—	—	0x1E*1	—
SCI9_ERI	—	—	—	—	—	—	—	0x1A*1
SCI9_AM	—	—	—	0x1B*1	—	—	—	—
SPI0_SPRI	0x14	—	—	—	0x14	—	—	—
SPI0_SPTI	—	0x11	—	—	—	0x11	—	—
SPI0_SPII	—	—	0x10*1	—	—	—	0x10*1	—
SPI0_SPEI	—	—	—	0x10*1	—	—	—	0x10*1
SPI0_SPTEND	—	—	0x11*1	—	—	—	0x11*1	—

Note 1. Use for CPU interrupt only.
Note 2. Use for DTC interrupt only.

12.4 Interrupt Operation

The ICU performs the following functions:

- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations such as CPU interrupt, DTC activation.

12.4.1 Detecting Interrupts

The ICU selects an event source input from a peripheral function interrupt or an external pin interrupt with IELSRn.IELS[5:0].

The accepted interrupt source sets the IELSRn.IR to 1 and sends an interrupt request to the NVIC.

External pin interrupt requests are detected by either:

- Edges (falling edge, rising edge, or rising and falling edges)
- Level (low level) of the interrupt signal.

Set the IRQCRi.IRQMD[1:0] bits to select the detection mode for the IRQi pins. For interrupt sources associated with peripheral modules, see Table 12.3 and Table 12.4. Events must be accepted by the NVIC before an interrupt occurs and is accepted by the CPU.

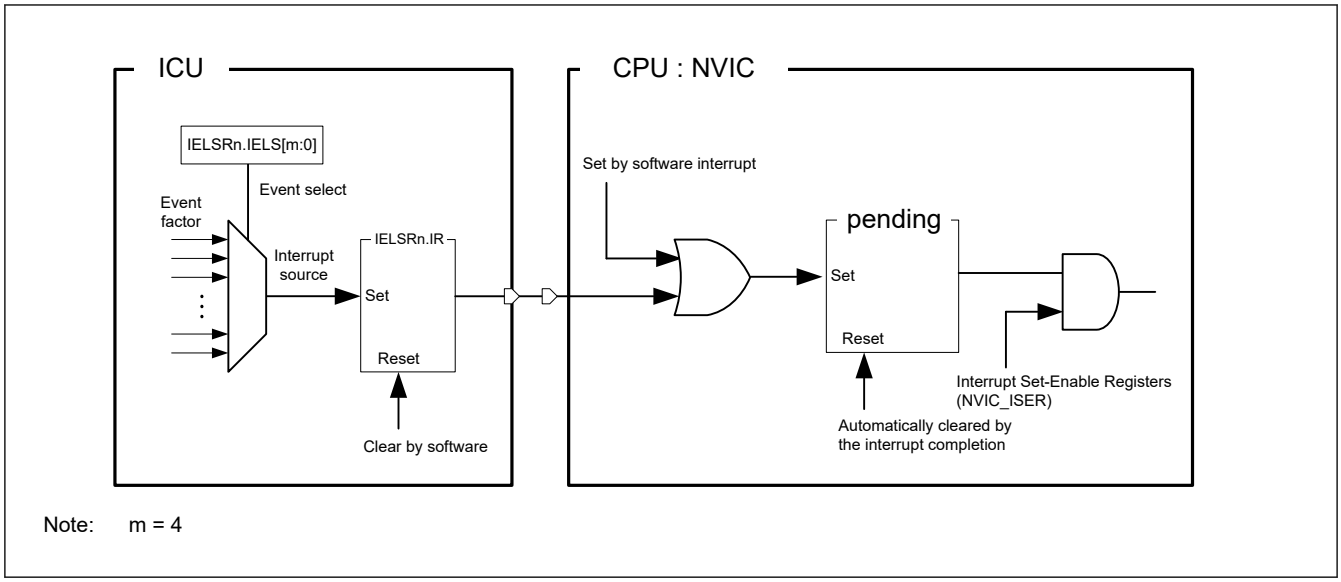


Figure 12.2 Interrupt path of the ICU and CPU (NVIC)

12.5 Interrupt setting procedure

12.5.1 Enabling Interrupt Requests

The procedure for enabling an interrupt request is as follows:

1. Set the Interrupt Set-Enable register (NVIC_ISER).
2. Set the IELSRn.IELS[5:0] bits as the interrupt source.
3. Specify the operation settings for the event source, such as snooze mode cancellation (SELSR0.SELS[7:0]), software standby mode cancellation (WUPEN register setting).

12.5.2 Disabling Interrupt Requests

The procedure to disable the interrupt request is as follows:

1. Disable the operation settings for the event source, such as snooze mode cancellation (SELSR0.SELS[7:0]), software standby mode cancellation (WUPEN register setting).
2. Clear the interrupt source setting (IELSRn.IELS[5:0] = 0x00).
3. Clear the interrupt status flag (IELSRn.IR = 0).
4. Clear the interrupt Clear-Enable register (NVIC_ICER) and interrupt Clear-Pending register (NVIC_ICPR).

12.5.3 Polling for interrupts

The procedure for polling for interrupt requests is as follows:

1. Set the Interrupt Clear-Enable register (NVIC_ICER).
2. Set the IELSRn.IELS[5:0] bits as the interrupt source.
3. Specify the operation settings for the event source, such as snooze mode cancellation (SELSR0.SELS[7:0]), software standby mode cancellation (WUPEN register setting).
4. Poll the interrupt Set-Pending register (NVIC_ISPR).

12.5.4 Selecting Interrupt Request Destinations

The available destinations are fixed for each interrupt, as described in Table 12.3, Table 12.4 and Table 12.6.

The interrupt output destination, CPU, or DTC can be independently selected for each interrupt source.

Use an interrupt request destination setting that is indicated by a "✓" in the event list.

If the DTC is selected as the destination for requests from an IRQi pin, you must set the IRQCRi.IRQMD[1:0] bits for that interrupt to select edge detection.

12.5.4.1 CPU interrupt request

When IELSRn.DTCE = 0, the event specified in the IELSRn register is output to the NVIC. Set the IELSRn.IELS[5:0] bits to the target event and set the IELSRn.DTCE bit to 0.

12.5.4.2 DTC Activation

When IELSRn.DTCE = 1, the event specified in the IELSRn register is output to the DTC. Use the following procedure:

1. Set the IELSRn.IELS[5:0] bits to the target event and set the IELSRn.DTCE bit to 1.
2. Set the DTC Module Start bit (DTCST.DTCST) to 1.

Table 12.8 shows operation when the DTC is the interrupt request destination.

Table 12.8 Operation when DTC becomes interrupt request destination

Interrupt request destination	DISEL*1	Remaining transfer operations	Operation per request	IR*2	Interrupt request destination after transfer
DTC*3	1	≠ 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	CPU (IELSRn.DTCE bit is automatically cleared)
	0	≠ 0	DTC transfer	Cleared at the start of DTC data transfer after reading DTC transfer data	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	CPU (IELSRn.DTCE bit is automatically cleared)

Note 1. DTC.MRB.DISEL bit controls the interrupt generates timing from DTC to CPU.
Note 2. When the IELSRn.IR flag is 1, an interrupt request (DTC activation request) that occurs again is ignored.
Note 3. For chain transfers, DTC transfer continues until the last chain transfer ends. The DISEL bit state and the remaining transfer count determine whether a CPU interrupt occurs, the IELSRn.IR flag clear timing, and the interrupt request destination after transfer. See Table 15.2 in section 15, Data Transfer Controller (DTC).

12.5.5 Digital Filter

A digital filter function is provided for the external interrupt request pins IRQi, (i = 0 to 7) and the NMI pin interrupt. It samples input signals on the filter PCLKB sampling clock and removes any signal with a pulse width less than 3 sampling cycles.

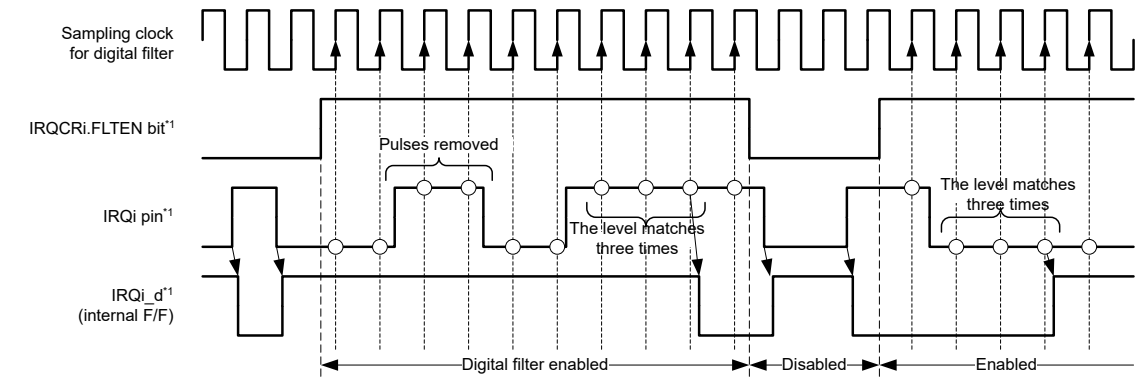
To use the digital filter for an IRQi pin:

1. Set the sampling clock cycle to PCLKB, PCLKB/8, PCLKB/32, or PCLKB/64 in the IRQCRi.FCLKSEL[1:0] bits (i = 0 to 7).
2. Set the IRQCRi.FLTEN bit (i = 0 to 7) to 1 (digital filter enabled).

To use the digital filter for an NMI pin:

1. Set the sampling clock cycle to PCLKB, PCLKB/8, PCLKB/32, or PCLKB/64 in the NMICR.NFCLKSEL[1:0] bits.
2. Set the NMICR.NFLTEN bit to 1 (digital filter enabled).

Figure 12.3 shows an example of digital filter operation.



Note 1. i = 0 to 7

Figure 12.3 Digital filter operation example

Before entering Software Standby mode, disable the digital filters by clearing the IRQCRi.FLTEN and NMICR.NFLTEN bits. The ICU clock stops in Software Standby mode.

On exiting Software Standby mode, the circuit detects the edge by comparing the state before standby to the state after standby release. If the input changes during Software Standby mode, an incorrect edge might be detected. The digital filters can be enabled again after exiting Software Standby mode.

During Software Standby mode, the digital filter is forcibly turned off by Hardware. After returning to normal mode from Software Standby mode, follow the values of IRQCRi.FLTEN (i = 0 to 15) and NMICR.NFLTEN. Once the filter is disabled, the event information sampled up to that point will be lost.

12.5.6 External Pin Interrupts

To use external pin interrupts:

1. Configure I/O ports settings.
2. Clear the IRQCRi.FLTEN bit (i = 0 to 7) to 0 (digital filter disabled).
3. Set the IRQMD[1:0] bits of the given IRQCRi register (i = 0 to 7) to select the senses of detection.
4. Set the FCLKSEL[1:0] bits, and the FLTEN bit of the IRQCRi register.
5. Select the IRQ pin as follows:
   - If the IRQ pin is to be used for CPU interrupt requests, set the IELSRn.IELS[5:0] bits and the IELSRn.DTCE bit to 0.
   - If the IRQ pin is to be used for DTC activation, set the IELSRn.IELS[5:0] bits and the IELSRn.DTCE bit to 1.

12.6 Non-Maskable Interrupt Operation

The following sources can trigger a non-maskable interrupt:

- NMI pin interrupt
- Oscillation stop detection interrupt
- WDT underflow/refresh error interrupt
- IWDT underflow/refresh error interrupt
- Voltage monitor 1 interrupt
- Voltage monitor 2 interrupt

- SRAM parity error interrupt
- Bus master MPU error interrupt
- Bus slave MPU error interrupt
- CPU stack pointer monitor interrupt

Non-maskable interrupts can only be used with the CPU, not to activate the DTC. Non-maskable interrupts take precedence over all other interrupts. The non-maskable interrupt states can be verified in the Non-Maskable Interrupt Status Register (NMISR). Confirm that all bits in the NMISR are 0 before returning from the NMI handler.

Non-maskable interrupts are disabled by default. To use non-maskable interrupts:

1. Clear the NMICR.NFLTEN bit to 0 (digital filter disabled).
2. Set the NMIMD bit, NFCLKSEL[1:0] bits, and NFLTEN bit of NMICR register.
3. Write 1 to the NMICLR.NMICLR bit to clear the NMISR.NMIST flag to 0.
4. Enable the non-maskable interrupt by writing 1 to the associated bit in the Non-Maskable Interrupt Enable Register (NMIER).

After 1 is written to the NMIER register, subsequent write access to the NMIEN bit in NMIER is ignored. An NMI cannot be disabled when enabled, except by a reset.

12.7 Return from Low Power Modes

Table 12.4 lists the interrupt sources that can be used to exit Sleep or Software Standby mode. For more information, see section 10, Low Power Modes.

12.7.1 Return from Sleep Mode

To return from Sleep mode in response to an interrupt:

non-maskable interrupt

- Use the NMIER register to enable the target interrupt request.

maskable interrupt

- Select the CPU as the interrupt request destination.
- Enable the interrupt in the NVIC.

12.7.2 Return from Software Standby Mode

The ICU returns from Software Standby mode using a non-maskable interrupt or a maskable interrupt. For maskable interrupt of canceling source, see Table 12.4.

To return from Software Standby mode:

1. Select the interrupt source that enables return from Software Standby:
   - For non-maskable interrupts, use the NMIER register to enable the target interrupt request
   - For maskable interrupts, use the WUPEN register to enable the target interrupt request.
2. Select the CPU as the interrupt request destination
3. Enable the interrupt in the NVIC.

Interrupt requests through the IRQn pins that do not satisfy these conditions are not detected while the clock is stopped in Software Standby mode.

12.7.3 Return from Snooze Mode

The ICU can return to Normal mode from Snooze mode using the interrupts provided for this mode.

To return to Normal mode from Snooze mode:

1. Set the number of the required interrupt request in SELSR0.SELS[7:0]

2. Set the value 0x03 (ICU_SNZCANCEL) in IELSRn.IELS[5:0] (n = one of following numbers: 0, 4, 8, 12, 16, 20, 24, and 28).
3. Select the CPU as the interrupt request destination.
4. Enable the interrupt in the NVIC.

Note: In Snooze mode, a clock is supplied to the ICU. If an event selected in IELSRn is detected, the CPU acknowledges the interrupt after returning to Normal mode from Software Standby mode.

12.8 Using the WFI Instruction with Non-Maskable Interrupts

Whenever a WFI instruction is executed, confirm that all status flags in the NMISR register are 0.

12.9 Reference

- ARM® Cortex®-M23 Processor Technical Reference Manual (ARM DDI 0550C)

13. Buses

13.1 Overview

Table 13.1 lists the bus specifications, Figure 13.1 shows the bus configuration, and Table 13.2 lists the addresses assigned for each bus.

Table 13.1 Bus specifications

Bus type		Description
Main bus	System bus (CPU)	• Connected to CPU • Connected to on-chip memory and internal peripheral bus
	DMA bus	• Connected to DTC • Connected to on-chip memory and internal peripheral bus
Slave interface	Memory bus 1	Connected to code flash memory
	Memory bus 4	Connected to SRAM0
	Internal peripheral bus 1	Connected to system control related to peripheral modules
	Internal peripheral bus 3	• Connected to peripheral modules (CAC, ELC, I/O Ports, POEG, WDT, IWDT, IIC, ADC12, DOC, SCI, SPI, and CRC) • Connected to peripheral modules (KINT, AGTW,GPT,ACMPHS)
	Internal peripheral bus 9	Connected to flash memory (in P/E (Programming/Erasure)), data flash memory and TSN

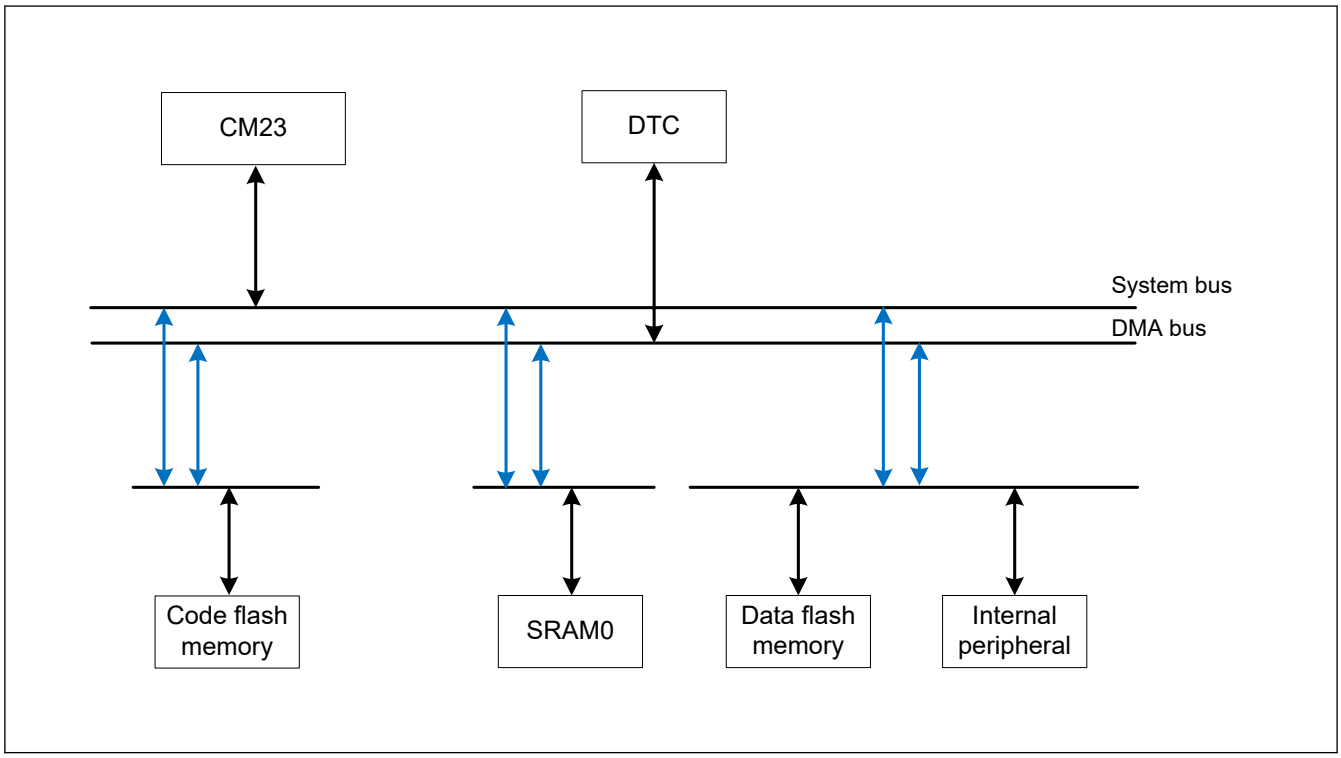


Figure 13.1 Bus configuration

Table 13.2 Addresses assigned for each bus (1 of 2)

Address	Bus	Area
0x0000_0000 to 0x01FF_FFFF	Memory bus 1	Code flash memory
0x2000_0000 to 0x2000_7FFF	Memory bus 4	SRAM0
0x4000_0000 to 0x4001_8FFF	Internal peripheral bus 1	Peripheral I/O registers
0x4001_9000 to 0x4001_9FFF	Memory bus 4	MTB I/O registers

Table 13.2 Addresses assigned for each bus (2 of 2)

Address	Bus	Area
0x4001_A000 to 0x4001_FFFF	Internal peripheral bus 1	Peripheral I/O registers
0x4004_0000 to 0x400B_FFFF	Internal peripheral bus 3	
0x4010_0000 to 0x407F_FFFF	Internal peripheral bus 9	Flash memory (in P/E), data flash memory and TSN

13.2 Description of Buses

13.2.1 Main Buses

The main buses consist of the system bus and DMA bus. The system bus and DMA bus are connected to the following:

- Code flash memory
- SRAM0
- Data flash memory
- Internal peripheral bus

The system bus is used for instruction and data accesses to the CPU.

Different master and slave transfer combinations can proceed simultaneously. In addition, requests for bus access from masters other than the DTC are not accepted during reads of transfer control information for the DTC.

13.2.2 Slave Interface

For connections from the main bus to the slave interfaces, see the slave interfaces in section 13.1. Overview.

Bus access from the system bus and DMA bus is arbitrated and has the following fixed priority order:

DMA bus > system bus

Different master and slave transfer combinations can proceed simultaneously.

13.2.3 Parallel Operations

Parallel operations are possible when different bus masters request access to different slave modules. Figure 13.2 shows an example of parallel operations. In this example, the CPU uses the instruction and operand buses for simultaneous access to the flash and SRAM, respectively. Additionally, the DTC simultaneously uses the DMA bus for access to a peripheral bus during access to the flash and SRAM by the CPU.

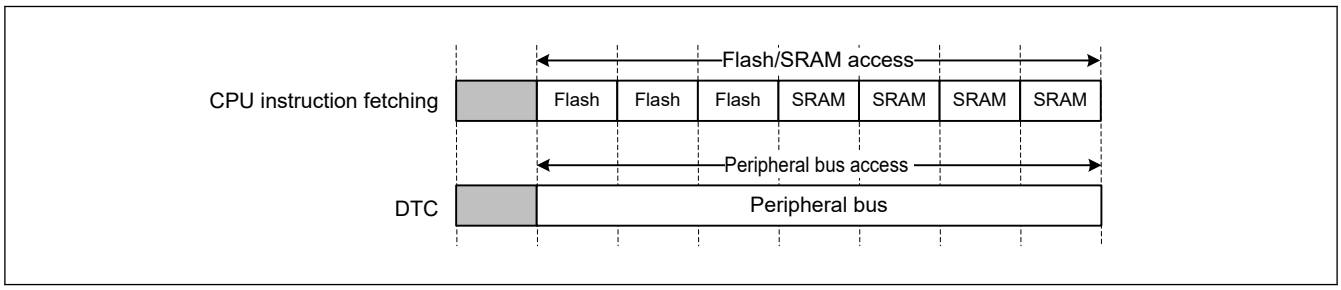


Figure 13.2 Example of parallel operations

13.2.4 Restriction on Endianness

Memory space must be little-endian to execute code on the Cortex®-M23 core.

13.2.5 Restriction on Exclusive Access

The main bus does not support exclusive transfers and there is no global monitor in the MCU.

The main bus always deasserts the HEXOKAY signal (a signal in the AHB-Lite protocol) to the CPU. This means that a store exclusive instruction such as STREX instruction always gets a failed status. When an exclusive write operation is performed by the CPU, the main bus always writes the data successfully.

13.3 Register Descriptions

13.3.1 BUSMCNTx : Master Bus Control Register x (x = SYS, DMA)

Base address: BUS = 0x4000_3000

Offset address: 0x1008 (BUSMCNTSYS)
0x100C (BUSMCNTDMA)

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

Bit	Symbol	Function	R/W
14:0	—	These bits are read as 0. The write value should be 0.	R/W
15	IERES	Ignore Error Responses 0: A bus error is reported. 1: A bus error is not reported.	R/W

Note: Changing reserved bits from the initial value of 0 is prohibited. Operation during the change is not guaranteed.

IERES bit (Ignore Error Responses)

The IERES bit specifies the enable or disable of an error response of the AHB-Lite protocol.

Table 13.3 lists the registers associated with each bus type.

Table 13.3 Associations between bus types and registers

Bus type	Master Bus Control Register	Bus Error Address Register	Bus Error Status Register
System bus (CPU)	BUSMCNTSYS	BUS3ERRADD	BUS3ERRSTAT
DMA bus	BUSMCNTDMA	BUS4ERRADD	BUS4ERRSTAT

13.3.2 BUSnERRADD : Bus Error Address Register n (n = 3, 4)

Base address: BUS = 0x4000_3000

Offset address: 0x1820 (n = 3)
0x1830 (n = 4)

Bit position:	31	0
Bit field:	BERAD[31:0]	
Value after reset:	x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x	

Bit	Symbol	Function	R/W
31:0	BERAD[31:0]	Bus Error Address When a bus error occurs, these bits store the error address.	R

Note: BUSnERRADD is only cleared by resets other than MPU-related resets. For more information, see section 5, Resets, and section 14, Memory Protection Unit (MPU).

Table 13.3 lists the registers associated with each bus type.

BERAD[31:0] bits (Bus Error Address)

The BERAD[31:0] bits store the accessed address when a bus error occurred. For more information, see the description of the ERRSTAT flag in section 13.3.3. BUSnERRSTAT : Bus Error Status Register n (n = 3, 4) and section 13.4. Bus Error Monitoring Section.

The value of the BUSnERRADD.BERAD[31:0] bits (n = 3, 4) is valid only when the BUSnERRSTAT.ERRSTAT flag (n = 3, 4) is set to 1.

13.3.3 BUSnERRSTAT : Bus Error Status Register n (n = 3, 4)

Base address: BUS = 0x4000_3000

Offset address: 0x1824 (n = 3)
0x1834 (n = 4)

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

Bit	Symbol	Function	R/W
0	ACCSTAT	Error Access Status flag Access status when the error occurred: 0: Read access 1: Write access	R
6:1	—	These bits are read as 0.	R
7	ERRSTAT	Bus Error Status flag 0: No bus error occurred. 1: Bus error occurred.	R

Note: BUSnERRSTAT is only cleared by resets other than MPU-related resets. For more information, see section 5, Resets, and section 14, Memory Protection Unit (MPU).

Table 13.3 lists the registers associated with each bus type.

ACCSTAT flag (Error Access Status flag)

The ACCSTAT flag indicates the access status, write or read access, when a bus error occurs. For more information, see the description of the BUSnERRSTAT.ERRSTAT flag and section 13.4. Bus Error Monitoring Section.

The value is valid only when the BUSnERRSTAT.ERRSTAT flag (n = 3, 4) is set to 1.

ERRSTAT flag (Bus Error Status flag)

The ERRSTAT flag indicates whether a bus error occurred. When a bus error occurs, the access address and status of write or read access are stored. The BUSnERRSTAT.ERRSTAT flag (n = 3, 4) is set to 1.

For more information on bus errors, see section 13.4. Bus Error Monitoring Section and section 14, Memory Protection Unit (MPU).

13.4 Bus Error Monitoring Section

The monitoring system monitors each individual area, and whenever it detects an error, it returns the error to the requesting master IP using the AHB-Lite error response protocol.

13.4.1 Error Type that Occurs by Bus

Three types of errors can occur on each bus:

- Illegal address access
- Bus master MPU error
- Bus slave MPU error

section 13.4.3. Conditions for issuing illegal Address Access Errors lists the address ranges where access leads to illegal address access errors. The reserved area in the slave does not trigger an illegal address access error.

For more information on bus master MPU and bus slave MPU, see section 14, Memory Protection Unit (MPU).

13.4.2 Operation when a Bus Error Occurs

When a bus error occurs, operation is not guaranteed and the error is returned to the requesting master IP. The bus error information occurred in each master is stored in the BUSnERRADD and BUSnERRSTAT registers. These registers must be cleared by reset only. For more information, see section 13.3.2. BUSnERRADD : Bus Error Address Register n (n = 3, 4) and section 13.3.3. BUSnERRSTAT : Bus Error Status Register n (n = 3, 4).

Note: DTC does not receive bus errors. If the DTC accesses the bus, the transfer continues.

13.4.3 Conditions for issuing illegal Address Access Errors

Table 13.4 lists the address spaces for each bus that issue illegal address access errors.

Table 13.4 Conditions leading to illegal address access errors

Address	Slave bus name	Main buses	
		System bus(CPU)	DMA bus
0x0000_0000 to 0x01FF_FFFF	Memory bus 1	—	—
0x0200_0000 to 0x1FFF_FFFF	Reserved	E	E
0x2000_0000 to 0x2000_7FFF	Memory bus 4	—	—
0x2000_8000 to 0x3FFF_FFFF	Reserved	E	E
0x4000_0000 to 0x4001_8FFF	Internal peripheral bus 1	—	—
0x4001_9000 to 0x4001_9FFF	Memory bus 4	—	—
0x4001_A000 to 0x4001_FFFF	Internal peripheral bus 1	—	—
0x4002_0000 to 0x4003_FFFF	Reserved	E	E
0x4004_0000 to 0x400B_FFFF	Internal peripheral bus 3	—	—
0x400C_0000 to 0x400D_FFFF	Reserved	—	—
0x400E_0000 to 0x400F_FFFF	Reserved	E	E
0x4010_0000 to 0x407F_FFFF	Internal peripheral bus 9	—	—
0x4080_0000 to 0xDFFF_FFFF	Reserved	E	E
0xE000_0000 to 0xFFFF_FFFF	System for Cortex®-M23	—	E

Note: E indicates the path where an illegal address access error occurs.
— indicates the path where an illegal address access error does not occur.
Note: The bus module detects an access error resulting from access to reserved area, for example if no area is assigned for the slave.
0x0200_0000 to 0x1FFF_FFFF: Access error detection.
0x0000_0000 to 0x01FF_FFFF: Memory bus 1 no access error detection.

13.5 References

1. *ARM®v8-M Architecture Reference Manual* (ARM DDI0553B.a)
2. *ARM® Cortex®-M23 Processor User Guide* (ARM DUI0963B)
3. *ARM® AMBA® 5 AHB-Lite Protocol Specification* (ARM IHI0033B.b)

14. Memory Protection Unit (MPU)

14.1 Overview

The MCU has four Memory Protection Units (MPUs) and a CPU stack pointer monitor function are provided.

Table 14.1 lists the MPU specifications, and Table 14.2 shows the behavior on detection of each MPU error.

Table 14.1 MPU specifications

Classification	Module/Function	Specifications
Illegal memory access	Illegal address access	• Arm CPU has a default memory map. If the CPU makes an illegal address access, a Hard Fault occurs • The Arm MPU can change a default memory map.
	CPU stack pointer monitor	2 regions: • Main Stack Pointer (MSP) • Process Stack Pointer (PSP).
Memory protection	Arm MPU	Memory protection function for the CPU: • 8 MPU regions with sub regions and background region.
	Bus master MPU	Memory protection function for each bus master except for the CPU: • Bus master MPU group A: 4 regions
	Bus slave MPU	Memory protection function for each bus slave
Security	Security MPU	Protect accesses from Non-secure programs to the following secure regions: • 2 regions (PC) • 4 regions (code flash, SRAM, 2 secure functions).

Table 14.2 Behavior on MPU error detection

MPU type	Notice method	Bus access at error detection	Storing of error access information
CPU stack pointer monitor	Reset or non-maskable interrupt	Don't care	Not stored
Arm MPU	Hard fault	• Does not correctly have write access • Does not correctly have read access.	Not stored
Bus master MPU	Reset or non-maskable interrupt	• Write access to the protection region • Read access to the protection region.	Stored
Bus slave MPU	Reset or non-maskable interrupt	• Write access ignored • Read access is read as 0.	Stored
Security MPU	Not notified	• Does not correctly have write access • Does not correctly have read access.	Not stored

For information on error access for the Arm MPU, see section 14.8. References. For information on error access for other MPUs, see section 13.3. Register Descriptions and section 13.4. Bus Error Monitoring Section in section 13, Buses.

14.2 CPU Stack Pointer Monitor

The CPU stack pointer monitor detects underflows and overflows of the stack pointer. Because the Arm CPU has two stack pointers, a Main Stack Pointer (MSP) and a Process Stack Pointer (PSP), it supports two CPU stack pointer monitors. If a stack pointer underflow or overflow is detected, the CPU stack pointer monitor generates a reset or a non-maskable interrupt. The CPU stack pointer monitor is enabled by setting the Stack Pointer Monitor Enable bits in the Stack Pointer Monitor Access Control Register (MSPMPUCTL, PSPMPUCTL) to 1.

Table 14.3 lists the specifications of the CPU stack pointer monitor. Figure 14.1 shows a CPU stack pointer monitor block diagram and Figure 14.2 shows the CPU stack pointer monitor register setting flow.

Table 14.3 CPU stack pointer monitor specifications (1 of 2)

Parameter	Description
Region to be monitored	SRAM region

Table 14.3 CPU stack pointer monitor specifications (2 of 2)

Parameter	Description
Number of regions	2 regions: ● Main Stack Pointer (MSP) ● Process Stack Pointer (PSP).
Address specification for individual regions	Specifying start and end addresses for individual regions
Stack pointer monitor enable or disable setting for individual regions	Enabling or disabling stack pointer monitor for individual regions
Operation on error detection	Reset or non-maskable interrupts
Register protection	Prevents illegal writing to the CPU stack pointer monitor register

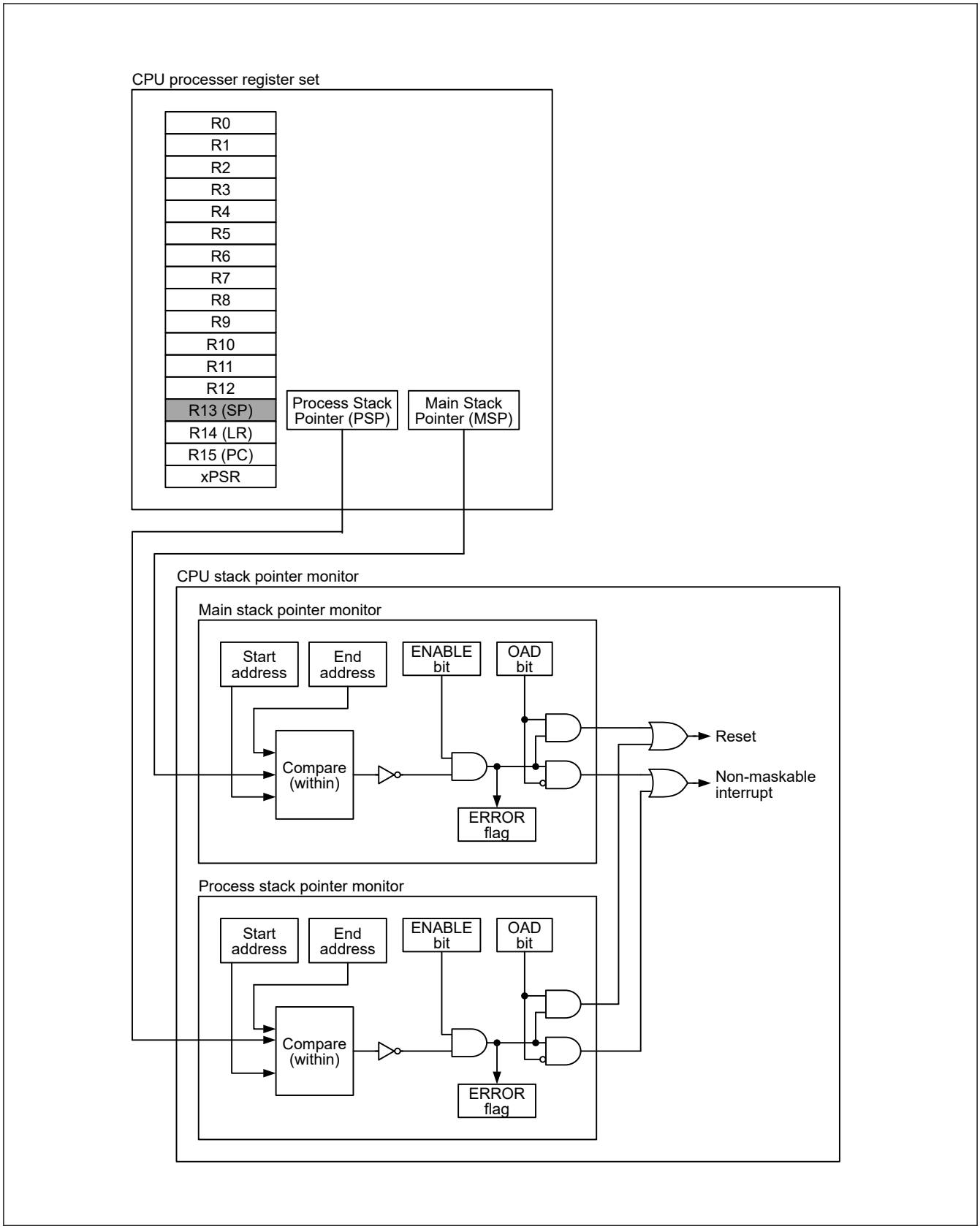


Figure 14.1 CPU stack pointer monitor block diagram

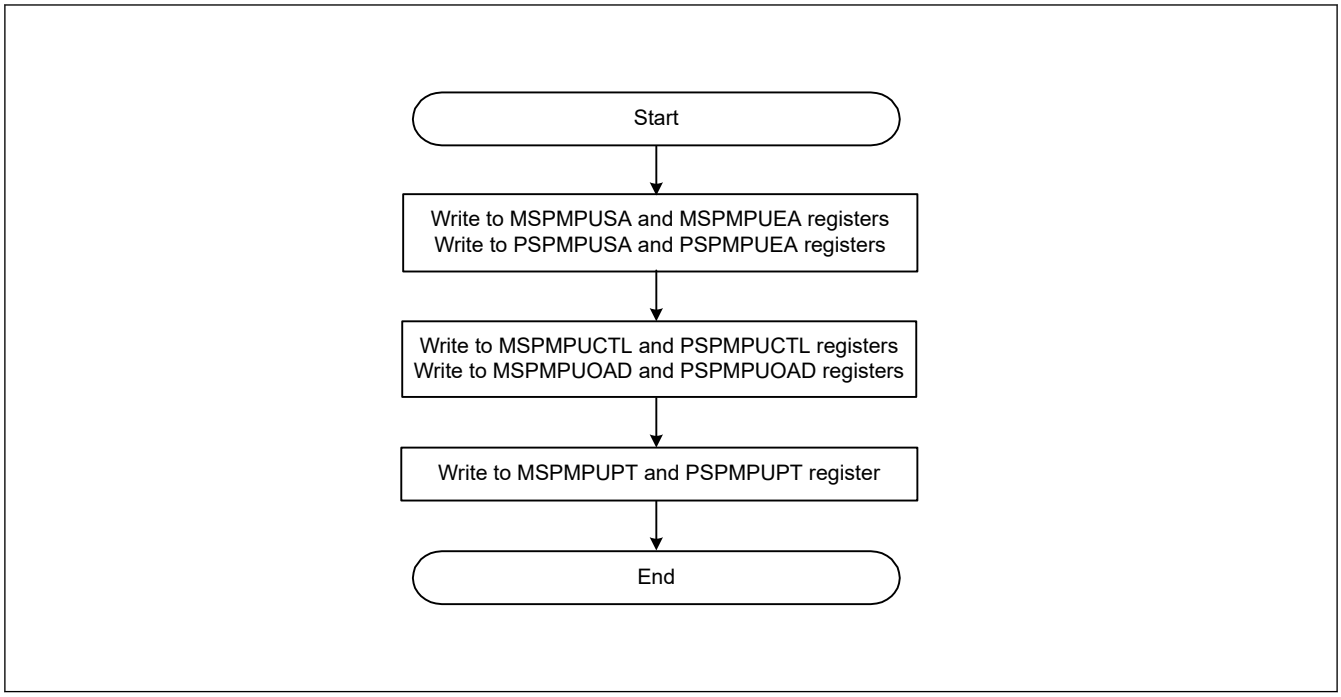


Figure 14.2 CPU stack pointer monitor register setting flow

14.2.1 Protecting the Registers

Registers related to the CPU stack pointer monitor can be protected with the associated PROTECT bit. For details, see section 14.2.3.7. MSPMPUPT, PSPMPUPT : Stack Pointer Monitor Protection Register.

14.2.2 Overflow and Underflow Errors

The CPU stack pointer monitor generates an error if an overflow or underflow error is detected. Set the OAD bit to select whether the error is reported as a non-maskable interrupt or reset.

The non-maskable interrupt status is indicated in ICU.NMISR.SPEST. For details, see section 12, Interrupt Controller Unit (ICU). The reset status is indicated in SYSTEM.RSTSR1.SPERF. For details, see section 5, Resets.

When ICU.NMISR.SPEST indicates that a CPU stack pointer monitor interrupt occurred, check the ERROR flags in MSPMPUCTL and PSPMPUCTL registers to determine whether it is a main stack pointer monitor error or a process stack pointer monitor error.

A non-maskable interrupt is generated continuously while the stack pointer overflows or underflows. To clear the error, set the stack pointer within the specified region and then clear the non-maskable interrupt flag by setting the ICU.NMICLR.SPECLR bit to 1. Write 0 to clear the ERROR flags in the MSPMPUCTL and PSPMPUCTL registers.

14.2.3 Register Descriptions

Note: Bus access must be stopped before writing to MPU registers.

14.2.3.1 MSPMPUSA : Main Stack Pointer (MSP) Monitor Start Address Register

Base address: RMPU = 0x4000_0000

Offset address: 0xD08

Bit position:	31																												0
Bit field:	MSPMPUSA[31:0]																												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	MSPMPUSA[31:0]	Region Start Address Address where the region starts, for use in region determination. The lower 2 bits should be 0. The value range is from 0x1FF0_0000 to 0x200F_FFFC, excluding reserved areas.	R/W

The MSPMPUSA and MSPMPUEA registers specify the CPU stack region of SRAM (0x1FF0_0000 to 0x200F_FFFF, excluding reserved areas). For SRAM area to be covered, see section 4.1. Address Space.

14.2.3.2 MSPMPUEA : Main Stack Pointer (MSP) Monitor End Address Register

Base address: RMPU = 0x4000_0000

Offset address: 0xD0C

Bit position:	31																												0
Bit field:	MSPMPUEA[31:0]																												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	MSPMPUEA[31:0]	Region End Address Address where the region ends, for use in region determination. The lower 2 bits should be 1. The value range is from 0x1FF0_0003 to 0x200F_FFFF, excluding reserved areas. For SRAM area to be covered, see section 4.1. Address Space.	R/W

14.2.3.3 PSPMPUSA : Process Stack Pointer (PSP) Monitor Start Address Register

Base address: RMPU = 0x4000_0000

Offset address: 0xD18

Bit position:	31																												0
Bit field:	PSPMPUSA[31:0]																												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	PSPMPUSA[31:0]	Region Start Address Address where the region starts, for use in region determination. The lower 2 bits should be 0. The value range is from 0x1FF0_0000 to 0x200F_FFFC, excluding reserved areas.	R/W

The PSPMPUSA and PSPMPUEA registers specify the CPU stack region of SRAM (0x1FF0_0000 to 0x200F_FFFF, excluding reserved areas). For SRAM area to be covered, see section 4.1. Address Space.

14.2.3.4 PSPMPUEA : Process Stack Pointer (PSP) Monitor End Address Register

Base address: RMPU = 0x4000_0000

Offset address: 0xD1C

Bit position:	31																												0
Bit field:	PSPMPUEA[31:0]																												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
31:0	PSPMPUEA[31:0]	Region End Address Address where the region ends, for use in region determination. The lower 2 bits should be 1. The value range is from 0x1FF0_0003 to 0x200F_FFFF, excluding reserved areas. For SRAM area to be covered, see section 4.1. Address Space.	R/W

14.2.3.5 MSPMPUOAD, PSPMPUOAD : Stack Pointer Monitor Operation After Detection Register

Base address: RMPU = 0x4000_0000

Offset address: 0xD00 (MSPMPUOAD)
0xD10 (PSPMPUOAD)

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

Bit	Symbol	Function	R/W
0	OAD	Operation after Detection 0: Non-maskable interrupt 1: Reset	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writes to the OAD bit	R/W*1

Note 1. Write data is not retained.

OAD bit (Operation after Detection)

The OAD bit selects either a reset or a non-maskable interrupt when a stack pointer underflow or overflow is detected by the CPU stack pointer monitor.

The main and the process stack pointer monitors each uses an OAD bit to determine which signal is generated when a stack pointer underflow or overflow is detected. Write 0xA5 in KEY[7:0] bits in halfword access simultaneously when setting the OAD bit.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the OAD bit. When writing to the OAD bit, simultaneously write 0xA5 to KEY[7:0] bits. When values other than 0xA5 are written to the KEY[7:0] bits, the OAD bit is not updated. The KEY[7:0] bits are always read as 0x00.

14.2.3.6 MSPMPUCTL, PSPMPUCTL : Stack Pointer Monitor Access Control Register

Base address: RMPU = 0x4000_0000

Offset address: 0xD04 (MSPMPUCTL)
0xD14 (PSPMPUCTL)

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

Bit	Symbol	Function	R/W
0	ENABLE	Stack Pointer Monitor Enable 0: Stack pointer monitor is disabled 1: Stack pointer monitor is enabled	R/W

Bit	Symbol	Function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	ERROR	Stack Pointer Monitor Error Flag 0: Stack pointer has not overflowed or underflowed 1: Stack pointer has overflowed or underflowed	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value depends on the reset generation sources.

ENABLE bit (Stack Pointer Monitor Enable)

The ENABLE bit enables or disables the stack pointer monitor function, independently set for the main stack pointer monitor and the process stack pointer monitor.

When the MSPMPUCTL.ENABLE bit is set to 1, the following registers are available:

- MSPMPUSA
- MSPMPUEA
- MSPMPUOAD.

When the PSPMPUCTL.ENABLE bit is set to 1, the following registers are available:

- PSPMPUSA
- PSPMPUEA
- PSPMPUOAD.

ERROR flag (Stack Pointer Monitor Error Flag)

The ERROR flag indicates the status of the stack pointer monitor. Each stack pointer monitor has an independent ERROR flag.

[Setting condition]

- Overflow or underflow of the stack pointer.

[Clearing condition]

- 0 is written to this flag
- A reset other than the bus master MPU error reset, bus slave MPU error reset, and stack pointer error reset. (For the details of reset factors, see section 5, Resets.)

Note: Only 0 can be written to the ERROR flag.

14.2.3.7 MSPMPUPT, PSPMPUPT : Stack Pointer Monitor Protection Register

Base address: RMPU = 0x4000_0000

Offset address: 0xD06 (MSPMPUPT)
0xD16 (PSPMPUPT)

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

Bit	Symbol	Function	R/W
0	PROTECT	Protection of Register 0: Stack pointer monitor register writes are permitted. 1: Stack pointer monitor register writes are protected. Reads are permitted	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writes to the PROTECT bit	R/W*1

Note 1. Write data is not retained.

PROTECT bit (Protection of Register)

The PROTECT bit enables or disables writes to the associated registers to be protected, independently set for the main stack pointer monitor and the process stack pointer monitor.

MSPMPUPT.PROTECT controls the following main stack pointer protection registers:

- MSPMPUCTL
- MSPMPUSA
- MSPMPUEA.

PSPMPUPT.PROTECT controls the following process stack pointer protection registers:

- PSPMPUCTL
- PSPMPUSA
- PSPMPUEA.

When writing to the PROTECT bit, simultaneously write 0xA5 to the KEY[7:0] bits, using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the PROTECT bit simultaneously. When writing to the PROTECT bit, write 0xA5 to KEY[7:0] bits. When values other than 0xA5 are written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0.

14.3 Arm MPU

The Arm MPU monitors the addresses accessed by the CPU across the entire address space (0x0000_0000 to 0xFFFF_FFFF) and provides support for:

- 8 protected regions
- Setting access permissions to protected region (Read, Write, Execution)
- Export of memory attributes to the system.

Arm MPU mismatches and permission violations invoke the programmable-priority MemManage fault (Hard Fault) handler. For details, see section 14.8. References.

14.4 Bus Master MPU

The bus master MPU monitors the addresses accessed by the bus masters in the entire address space (0x0000_0000 to 0xFFFF_FFFF). The access control information, consisting of read and write permissions, can be independently set for up to 4 regions. The bus master MPU monitors access to each region based on these settings.

If access to a protected region is detected, the bus master MPU generates an internal reset or a non-maskable interrupt. For information on error access, see section 13.3. Register Descriptions and section 13.4. Bus Error Monitoring Section in section 13, Buses.

The access control information for each area consists of protected/not-protected to read or write.

Table 14.4 lists the specifications of the bus master MPU, and Figure 14.3 shows a block diagram.

Table 14.4 Bus master MPU specifications (1 of 2)

Parameter	Description
Protected master groups	• Bus master MPU group A: DMA bus
Protected regions	0x0000_0000 to 0xFFFF_FFFF
Number of regions	• Bus master MPU group A: 4 regions
Address specification for individual regions	• Specifying start and end address for individual regions
Enable or disable setting for memory protection in individual regions	• Enabling or disabling setting for the associated region

Table 14.4 Bus master MPU specifications (2 of 2)

Parameter	Description
Access-control settings for individual regions	• Permission for read and write
Operation on error detection	• Reset or non-maskable interrupts
Register protection	• Protecting registers from illegal writes

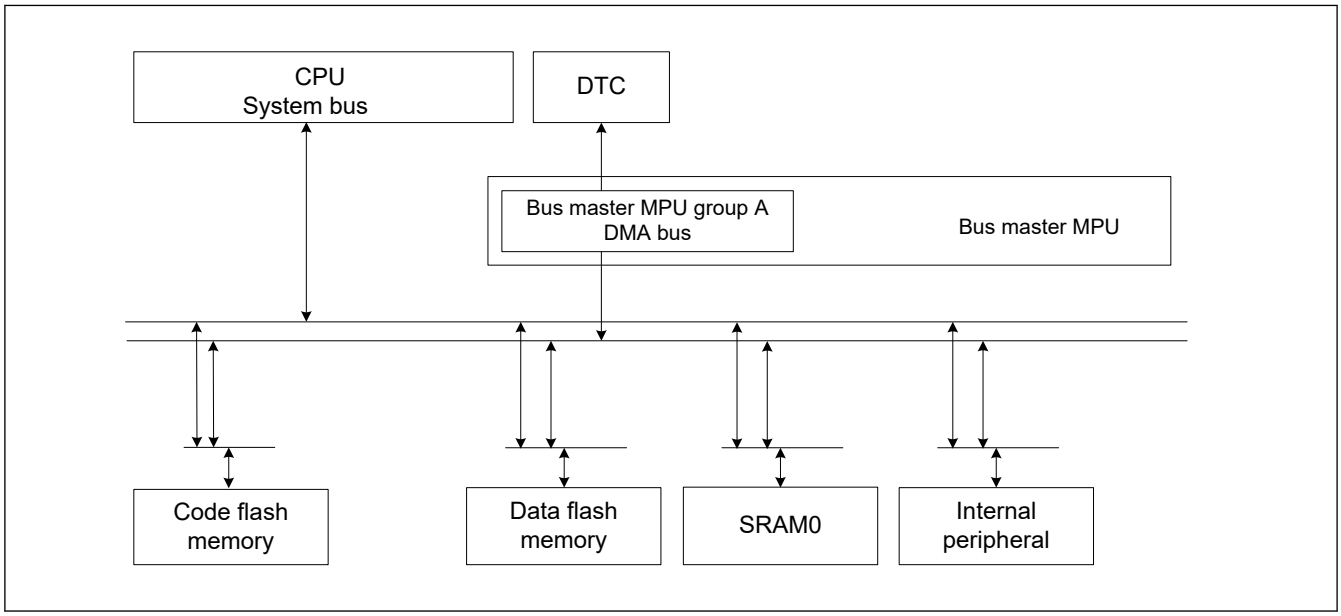


Figure 14.3 MPU bus master block diagram

Figure 14.4 shows the MPU bus master MPU group A.

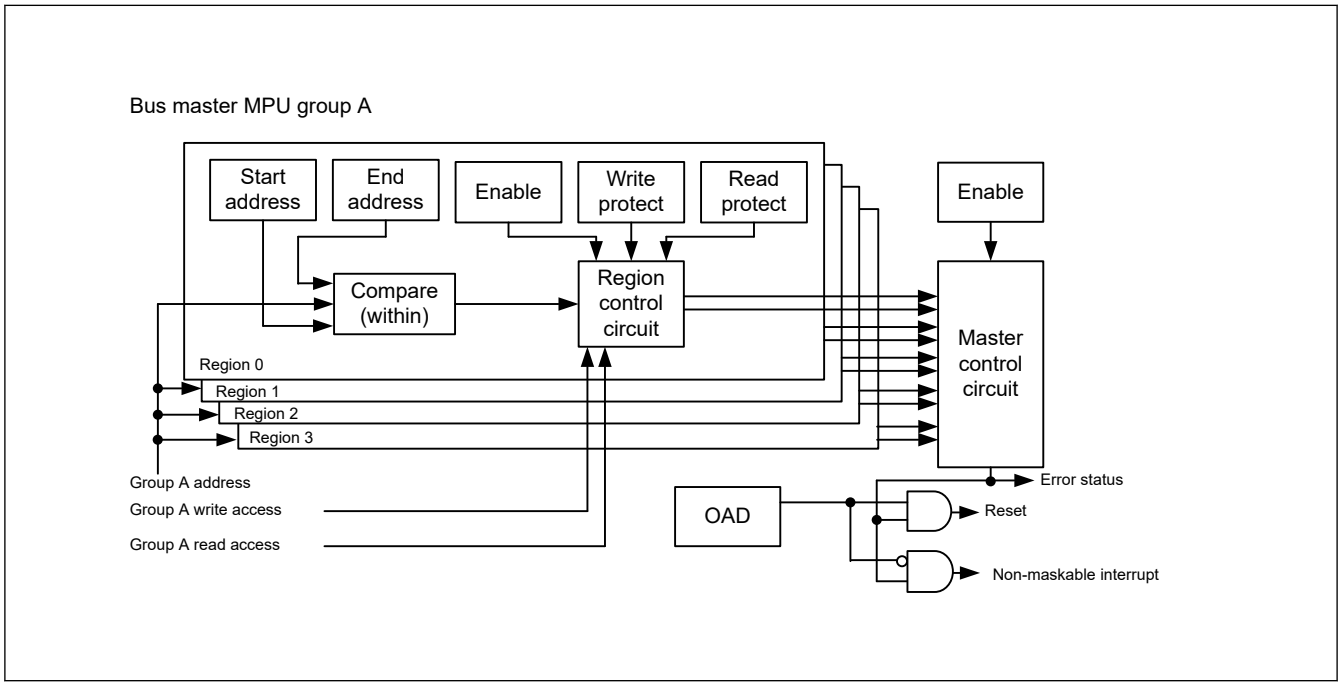


Figure 14.4 MPU bus master MPU group A

14.4.1 Register Descriptions

Note: Bus access must be stopped before writing to MPU registers.

14.4.1.1 MMPUSAn : Group A Region n Start Address Register (n = 0 to 3)

Base address: RMPU = 0x4000_0000

Offset address: 0x204 + (0x010 × n)

Bit position:	31																										1	0
Bit field:	MMPUSA[31:0]																											
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
31:0	MMPUSA[31:0]	Region Start Address Address where the region starts, for use in region determination. The lower 2 bits should be 0.	R/W

14.4.1.2 MMPUEAn : Group A Region n End Address Register (n = 0 to 3)

Base address: RMPU = 0x4000_0000

Offset address: 0x208 + 0x010 × n

Bit position:	31																										1	0
Bit field:	MMPUEA[31:0]																											
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1

Bit	Symbol	Function	R/W
31:0	MMPUEA[31:0]	Region End Address Address where the region ends, for use in region determination. The lower 2 bits should be 1.	R/W

14.4.1.3 MMPUACAn : Group A Region n Access Control Register (n = 0 to 3)

Base address: RMPU = 0x4000_0000

Offset address: 0x200 + 0x010 × n

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

Bit	Symbol	Function	R/W
0	ENABLE	Region Enable 0: Group A region n disabled 1: Group A region n enabled	R/W
1	RP	Read Protection 0: Read permission 1: Read protection	R/W
2	WP	Write Protection 0: Write permission 1: Write protection	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

The ENABLE, RP, and WP bits are individually configurable for each group A region n.

ENABLE bit (Region Enable)

The ENABLE bit enables or disables group A region n.

When the ENABLE bit is set to 1, the RP bit and the WP bit can be set to permit or protect access to the region that is set in MMPUSAn and MMPUEAn. When the ENABLE bit is set to 0, no region is specified for group A region n access.

RP bit (Read Protection)

The RP bit enables or disables read protection for group A region n. The RP bit is available when the ENABLE bit is set to 1.

WP bit (Write Protection)

The WP bit enables or disables write protection for group A region n. The WP bit is available when the ENABLE bit is set to 1.

Table 14.5 Function of region control circuit

MMPUACAn.ENABLE	MMPUACAn.RP	MMPUACAn.WP	Access	Region	Output of group A region n
0	—	—	Read	—	Outside of region
			Write		Outside of region
1	0	0	Read	Inside	Permitted region
				Outside	Outside of region
			Write	Inside	Permitted region
				Outside	Outside of region
	0	1	Read	Inside	Permitted region
				Outside	Outside of region
			Write	Inside	Protected region
				Outside	Outside of region
	1	0	Read	Inside	Protected region
				Outside	Outside of region
			Write	Inside	Permitted region
				Outside	Outside of region
	1	1	Read	Inside	Protected region
				Outside	Outside of region
			Write	Inside	Protected region
				Outside	Outside of region

Note: n = 0 to 3

Table 14.6 Function of master control circuit

MMPUCTLA.ENABLE	Output of group A region 0 unit	Output of group A region 1 unit	Output of group A region 2 to 3 unit	Function of group A
1	Protected region	Don’t care	Don’t care	Generate error
1	Don’t care	Protected region	Don’t care	Generate error
1	Don’t care	Don’t care	Protected region	Generate error
1	Outside of region	Outside of region	Outside of region	Generate error
Other case				No error

A master MPU error occurs on the following conditions:

- MMPUCTLA.ENABLE = 1, and output of one or more region n units is to a protected region
- MMPUCTLA.ENABLE = 1, and output of all region n units is outside of region.

Other cases are handled as permitted regions.

14.4.1.4 MMPUCTLA : Bus Master MPU Control Register

Base address: RMPU = 0x4000_0000

Offset address: 0x000

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

Bit	Symbol	Function	R/W
0	ENABLE	Master Group Enable 0: Master group A disabled 1: Master group A enabled	R/W
1	OAD	Operation After Detection 0: Non-maskable interrupt 1: Reset	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writes to the OAD and ENABLE bits	R/W*1

Note 1. Write data is not retained.

ENABLE bit (Master Group Enable)

The ENABLE bit enables or disables the bus master MPU function of master group A.

When this bit is set to 1, MMPUACAn is available. When this bit is set to 0, MMPUACAn is unavailable, including permission for all regions. When writing to the ENABLE bit simultaneously, write 0xA5 to the KEY[7:0] bits using halfword access.

OAD bit (Operation After Detection)

The OAD bit generates either a reset or non-maskable interrupt when access to the protected region is detected by the bus master MPU. When writing to the OAD bit simultaneously, write 0xA5 to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the ENABLE and OAD bits. When writing to the ENABLE and OAD bits simultaneously, write 0xA5 to the KEY[7:0] bits. When other values are written to the KEY[7:0] bits, the ENABLE and the OAD bits are not updated. The KEY[7:0] bits are always read as 0x00.

14.4.1.5 MMPUPTA : Group A Protection of Register

Base address: RMPU = 0x4000_0000

Offset address: 0x102

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

Bit	Symbol	Function	R/W
0	PROTECT	Protection of Register 0: All bus master MPU group A register writes are permitted. 1: All bus master MPU group A register writes are protected. Reads are permitted.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writes to the PROTECT bit	R/W*1

Note 1. Write data is not retained.

PROTECT bit (Protection of Register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

MMPUPTA.PROTECT controls the bus master MPU group A protection registers. The following registers are protected by MMPUPTA.PROTECT:

- MMPUSAn
- MMPUEAn
- MMPUACAn
- MMPUCTLA.

When the PROTECT bit is set simultaneously, write 0xA5 to the KEY[7:0] bits using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the PROTECT bit. When writing to the PROTECT bit simultaneously, write 0xA5 to the KEY[7:0] bits. When other values are written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

14.4.2 Operation

14.4.2.1 Memory protection

The bus master MPU monitors memory access using control settings made individually for the access control regions. If access to a protected region is detected, the bus master MPU generates a memory protection error.

The bus master MPU can be set for up to 4 protected regions. Protected regions include those with overlapping permitted and protected regions, and those with two overlapping permitted regions.

The bus master MPU has group A. The memory protection function checks the address of the bus for the master group and all master group accesses are protected. The bus master MPU sets the permission for all of the regions after reset. Setting MMPUCTLA.ENABLE to 1 protects all of the regions. Each region sets up a permitted region within the protected region. If access to the protected region is detected, the bus master MPU generates an error.

Figure 14.5 shows the use case of a bus master MPU.

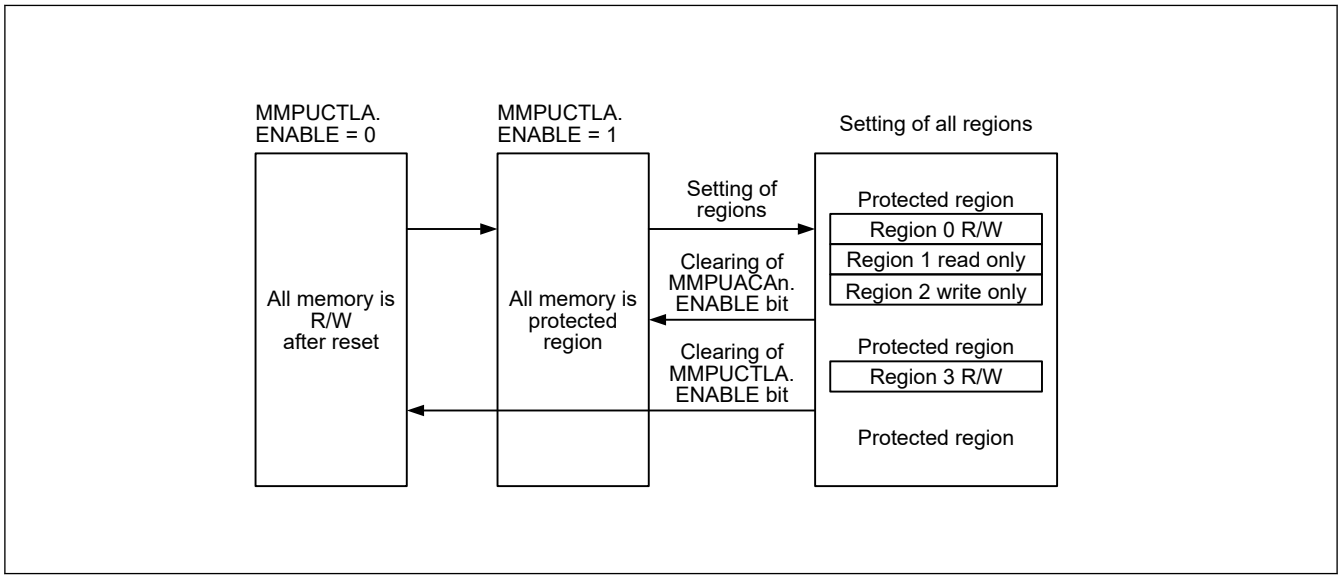


Figure 14.5 Use case of bus master MPU

Figure 14.6 shows the access permission or protection by the overlapping bus master MPU regions.

Access control for the overlapping regions is as follows:

- The region is handled as a protected region when output of one or more region units is a protected region
- The region is handled as a protected region when output of all region units is outside of the regions

- Other cases are handled as permitted regions.

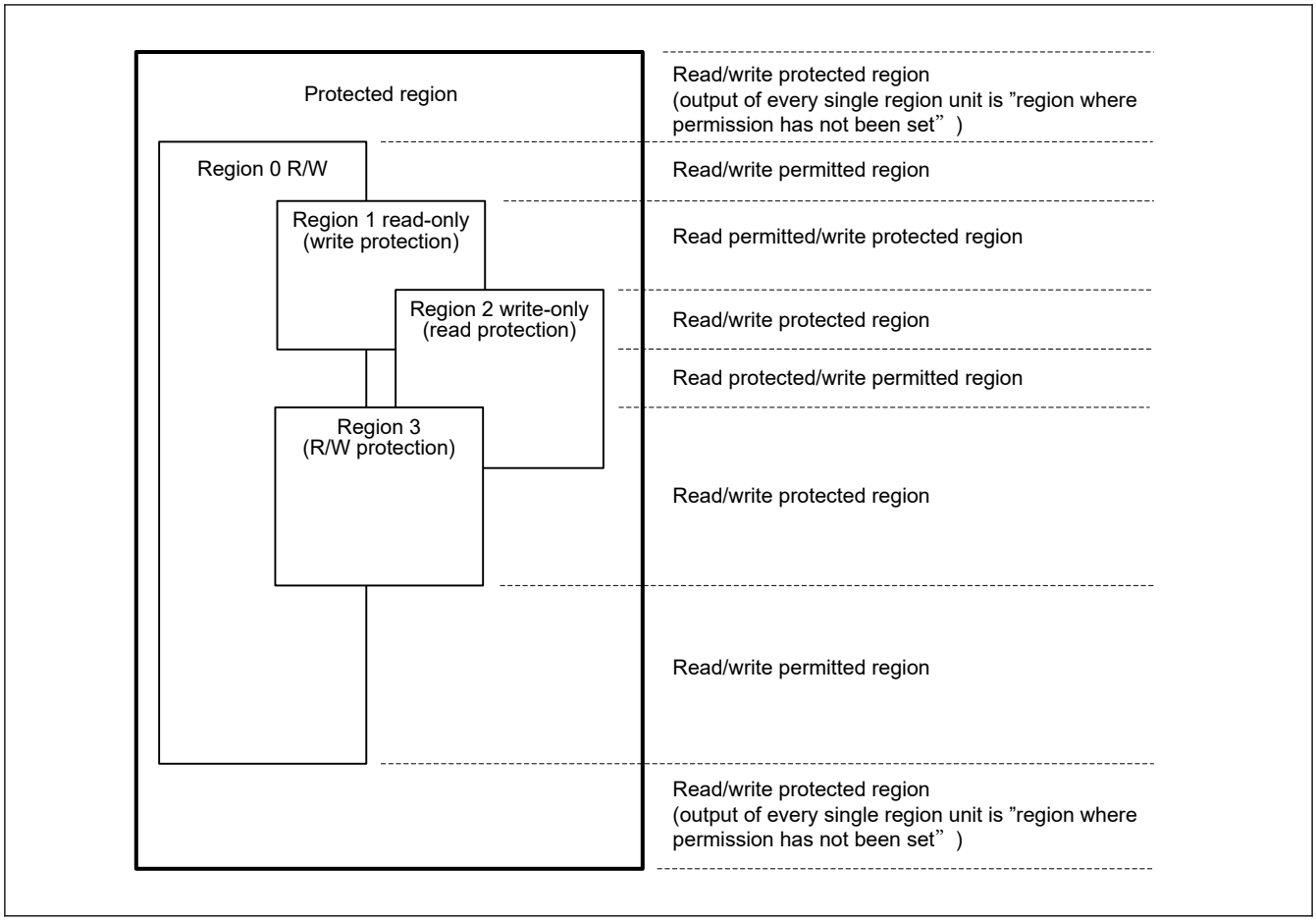


Figure 14.6 Access permission or protection by overlap of the bus master MPU regions

Figure 14.7 shows the register setting flow after reset. During this register setting, stop all bus masters except the CPU.

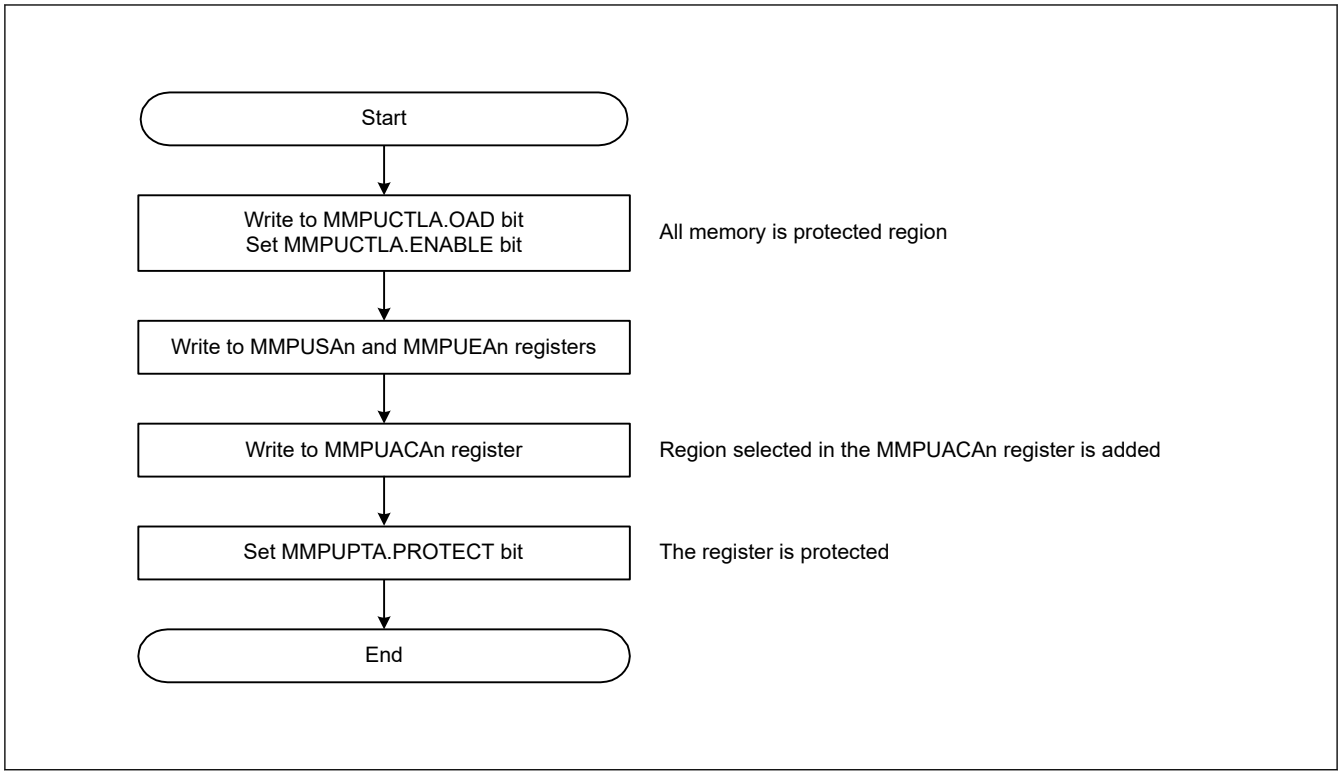


Figure 14.7 Register setting flow of bus master MPU after reset

Figure 14.8 shows the register setting flow for adding regions. During this register setting, stop all masters except the CPU.

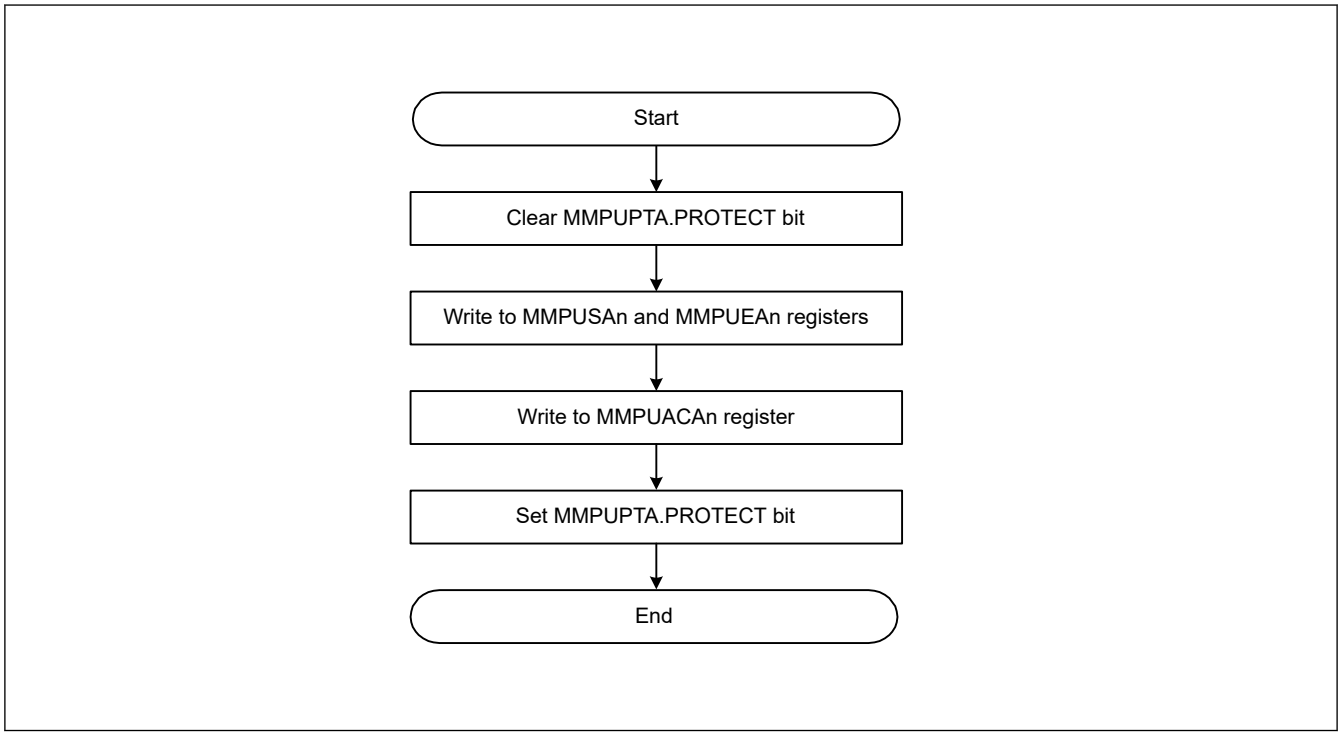


Figure 14.8 Register setting flow for region addition

14.4.2.2 Protecting the registers

To protect the registers related to the bus master MPU, set the PROTECT bit in the MMPUPTA register.

14.4.2.3 Memory protection error

If access to a protected region is detected, the bus master MPU generates an error. Set the OAD bit to select whether the error is reported as a non-maskable interrupt or a reset.

The non-maskable interrupt status is indicated in ICU.NMISR.BUSMST. For details, see section 12, Interrupt Controller Unit (ICU). The reset status is indicated in SYSTEM.RSTSR1.BUSMRF. For details, see section 5, Resets.

14.5 Bus Slave MPU

The bus slave MPU monitors access to the bus slave functions, such as flash memory or SRAM. The bus slave function can be accessed from two bus masters, the CPU, and the bus master MPU group A. The bus slave MPU has a separate protection register for each of the two bus masters, with individual access protection control. If access to a protected region is detected, the bus slave MPU generates a reset or a non-maskable interrupt, and store the bus error status, error access status, and bus error address in the I/O Registers. For details, see section 13.3. Register Descriptions and section 13.4. Bus Error Monitoring Section in section 13, Buses. The supported access control information for the individual regions consists of permission to read and write.

Table 14.7 lists the specifications of the bus slave MPU and Figure 14.9 shows a Bus slave MPU block diagram.

Table 14.7 Specifications of bus slave MPU

Specifications	Description	
Protected bus master	Bus master MPU group A:	DMA bus and System bus (CPU)
Protected bus slave function	Memory bus 1:	Code flash memory
	Memory bus 4:	SRAM0
	Internal peripheral bus 1:	Connected to peripheral modules related system control
	Internal peripheral bus 3:	Connected to peripheral modules (CAC, ELC, I/O Ports, POEG, WDT, IWDT, IIC, ADC12, DOC, GPT, SCI, SPI, CRC, KINT, AGTW, ACMPHS, and MSTP)
	Internal peripheral bus 9:	Connected to flash memory(in P/E), data flash and TSN.
Access-control information settings for individual regions	Permission to read and write	
Operation on error detection	Reset or non-maskable interrupt	
Protection of register	Register can be protected from illegal writes	

The bus slave MPU is located on each bus slave side and controls the permission or protection of access from each bus master to each bus slave.

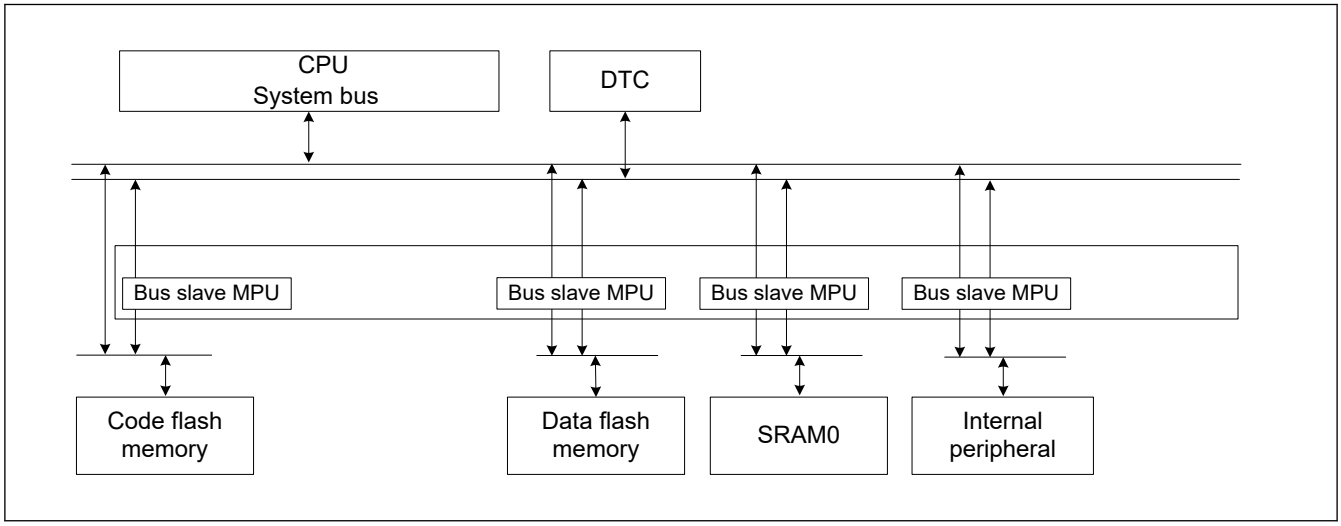


Figure 14.9 Bus slave MPU block diagram

14.5.1 Register Descriptions

Note: Bus access must be stopped before writing to MPU registers.

14.5.1.1 SMPUMBIU : Access Control Register for Memory Bus 1

Base address: RMPU = 0x4000_0000

Offset address: 0xC10

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

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	RPGRPA	Master MPU Group A Read Protection 0: Memory protection read for master MPU group A disabled 1: Memory protection read for master MPU group A enabled	R/W
3	WPGRPA	Master MPU Group A Write Protection 0: Memory protection write for master MPU group A disabled 1: Memory protection write for master MPU group A enabled	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

RPGRPA bit (Master MPU Group A Read Protection)

The RPGRPA bit enables or disables memory protection for master MPU group A reads on memory bus 1.

WPGRPA bit (Master MPU Group A Write Protection)

The WPGRPA bit enables or disables memory protection for master MPU group A writes on memory bus 1.

14.5.1.2 SMPUSRAM0 : Access Control Register for Memory Bus 4

Base address: RMPU = 0x4000_0000

Offset address: 0xC18

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

Bit	Symbol	Function	R/W
0	RPCPU	CPU Read Protection 0: Memory protection for CPU read disabled 1: Memory protection for CPU read enabled	R/W
1	WPCPU	CPU Write Protection 0: Memory protection for CPU write disabled 1: Memory protection for CPU write enabled	R/W
2	RPGRPA	Master MPU Group A Read Protection 0: Memory protection for master MPU group A read disabled 1: Memory protection for master MPU group A read enabled	R/W
3	WPGRPA	Master MPU Group A Write Protection 0: Memory protection for master MPU group A write disabled 1: Memory protection for master MPU group A write enabled	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for CPU reads on memory bus 4.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for CPU writes on memory bus 4.

RPGRPA bit (Master MPU Group A Read Protection)

The RPGRPA bit enables or disables memory protection for master MPU group A reads on memory bus 4.

WPGRPA bit (Master MPU Group A Write Protection)

The WPGRPA bit enables or disables memory protection for master MPU group A writes on memory bus 4.

14.5.1.3 SMPUP0BIU : Access Control Register for Internal Peripheral Bus 1

Base address: RMPU = 0x4000_0000

Offset address: 0xC20

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

Bit	Symbol	Function	R/W
0	RPCPU	CPU Read Protection 0: Memory protection for CPU read disabled 1: Memory protection for CPU read enabled	R/W
1	WPCPU	CPU Write Protection 0: Memory protection for CPU write disabled 1: Memory protection for CPU write enabled	R/W
2	RPGRPA	Master MPU Group A Read Protection 0: Memory protection for master MPU group A read disabled 1: Memory protection for master MPU group A read enabled	R/W
3	WPGRPA	Master MPU Group A Write Protection 0: Memory protection for master MPU group A write disabled 1: Memory protection for master MPU group A write enabled	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for CPU reads on internal peripheral bus 1.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for CPU writes on internal peripheral bus 1.

RPGRPA bit (Master MPU Group A Read Protection)

The RPGRPA bit enables or disables memory protection for master MPU group A reads on internal peripheral bus 1.

WPGRPA bit (Master MPU Group A Write Protection)

The WPGRPA bit enables or disables memory protection for master MPU group A writes on internal peripheral bus 1.

14.5.1.4 SMPUP2BIU : Access Control Register for Internal Peripheral Bus 3

Base address: RMPU = 0x4000_0000

Offset address: 0xC24

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

Bit	Symbol	Function	R/W
0	RPCPU	CPU Read Protection 0: Memory protection for CPU read disabled 1: Memory protection for CPU read enabled	R/W
1	WPCPU	CPU Write Protection 0: Memory protection for CPU write disabled 1: Memory protection for CPU write enabled	R/W
2	RPGRPA	Master MPU Group A Read Protection 0: Memory protection for master MPU group A read disabled 1: Memory protection for master MPU group A read enabled	R/W
3	WPGRPA	Master MPU Group A Write Protection 0: Memory protection for master MPU group A write disabled 1: Memory protection for master MPU group A write enabled	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for CPU reads on internal peripheral bus 3.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for CPU writes on internal peripheral bus 3.

RPGRPA bit (Master MPU Group A Read Protection)

The RPGRPA bit enables or disables memory protection for master MPU group A reads on internal peripheral bus 3.

WPGRPA bit (Master MPU Group A Write Protection)

The WPGRPA bit enables or disables memory protection for master MPU group A writes on internal peripheral bus 3.

14.5.1.5 SMPUFBIU : Access Control Register for Internal Peripheral Bus 9

Base address: RMPU = 0x4000_0000

Offset address: 0xC14

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

Bit	Symbol	Function	R/W
0	RPCPU	CPU Read Protection 0: Memory protection for CPU read disabled 1: Memory protection for CPU read enabled	R/W
1	WPCPU	CPU Write Protection 0: Memory protection for CPU write disabled 1: Memory protection for CPU write enabled	R/W

Bit	Symbol	Function	R/W
2	RPGRPA	Master MPU Group A Read Protection 0: Memory protection for master MPU group A read disabled 1: Memory protection for master MPU group A read enabled	R/W
3	WPGRPA	Master MPU Group A Write Protection 0: Memory protection for master MPU group A write disabled 1: Memory protection for master MPU group A write enabled	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

RPCPU bit (CPU Read Protection)

The RPCPU bit enables or disables memory protection for CPU reads on internal peripheral bus 9.

WPCPU bit (CPU Write Protection)

The WPCPU bit enables or disables memory protection for CPU writes on internal peripheral bus 9.

RPGRPA bit (Master MPU Group A Read Protection)

The RPGRPA bit enables or disables memory protection for master MPU group A reads on internal peripheral bus 9.

WPGRPA bit (Master MPU Group A Write Protection)

The WPGRPA bit enables or disables memory protection for master MPU group A writes on internal peripheral bus 9.

14.5.1.6 SMPUCTL : Slave MPU Control Register

Base address: RMPU = 0x4000_0000

Offset address: 0xC00

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

Bit	Symbol	Function	R/W
0	OAD	Operation After Detection 0: Non-maskable interrupt 1: Reset	R/W
1	PROTECT	Protection of Register 0: All bus slave register writes are permitted 1: All bus slave register writes are protected. Reads are permitted	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writes to the OAD and PROTECT bits	R/W*1

Note 1. Write data is not retained.

OAD bit (Operation After Detection)

The OAD bit generates either a reset or non-maskable interrupt when access to the protected region is detected by the bus slave MPU. When the OAD bit is set simultaneously, write 0xA5 to the KEY[7:0] bits using halfword access.

PROTECT bit (Protection of Register)

The PROTECT bit enables or disables writes to the associated registers to be protected. SMPUCTL.PROTECT controls the following registers:

- SMPUMBIU
- SMPUFBIU
- SMPUSRAM0
- SMPUP0BIU

- SMPUP2BIU
- SMPUP6BIU.

When the PROTECT bit is set, write 0xA5 to the KEY[7:0] bits simultaneously using halfword access.

KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writes to the OAD and PROTECT bits. When writing to the OAD and PROTECT bits simultaneously, write 0xA5 to the KEY[7:0] bits. When other values are written, the OAD and the PROTECT bits are not updated. The KEY[7:0] bits are always read as 0x00.

14.5.2 Functions

14.5.2.1 Memory protection

The bus slave MPU monitoring uses access control information that is set for the individual access control registers, whether or not access by the bus slaves violates the access control settings. If access to the protected region is detected, the bus slave MPU generates a memory protection error.

The bus slave MPU is enabled by writing 1 to the Write Protect (WPCPU or WPGRPA) bit or the Read Protect (RPCPU or RPGRPA) bit in the access control registers (SMPUMBIU, SMPUFBIU, SMPUSRAM0, SMPUP0BIU, SMPUP2BIU, and SMPUP6BIU).

14.5.2.2 Protecting the registers

Registers related to the bus slave MPU can be protected with the PROTECT bit in the SMPUCTL register.

14.5.2.3 Memory protection error

If access to a protected region is detected, the bus slave MPU generates a memory protection error. Set the OAD bit to select whether the error is reported as a non-maskable interrupt or reset.

The non-maskable interrupt status is indicated in ICU.NMISR.BUSSST. For details, see section 12, Interrupt Controller Unit (ICU). The reset status is indicated in SYSTEM.RSTSR1.BUSSRF. For details, see section 5, Resets.

14.6 Security MPU

The MCU incorporates a security MPU with four secure regions that include the code flash, SRAM, and two security functions. The secure regions can be protected from Non-secure program accesses. A Non-secure program cannot access a protected region.

Table 14.8 lists the specifications of the security MPU and Figure 14.10 shows a block diagram.

Table 14.8 Security MPU specifications

Specifications	Description
Secure regions	Code flash, SRAM, two security functions
Protected regions	0x0000_0000 to 0xFFFF_FFFF
Number of regions	Program Counter = 2 regions Data Access = 4 regions
Address specification for individual regions	Setting the address where regions start and end
Enable or disable setting for memory protection in individual regions	Settings enabled or disabled for the associated region

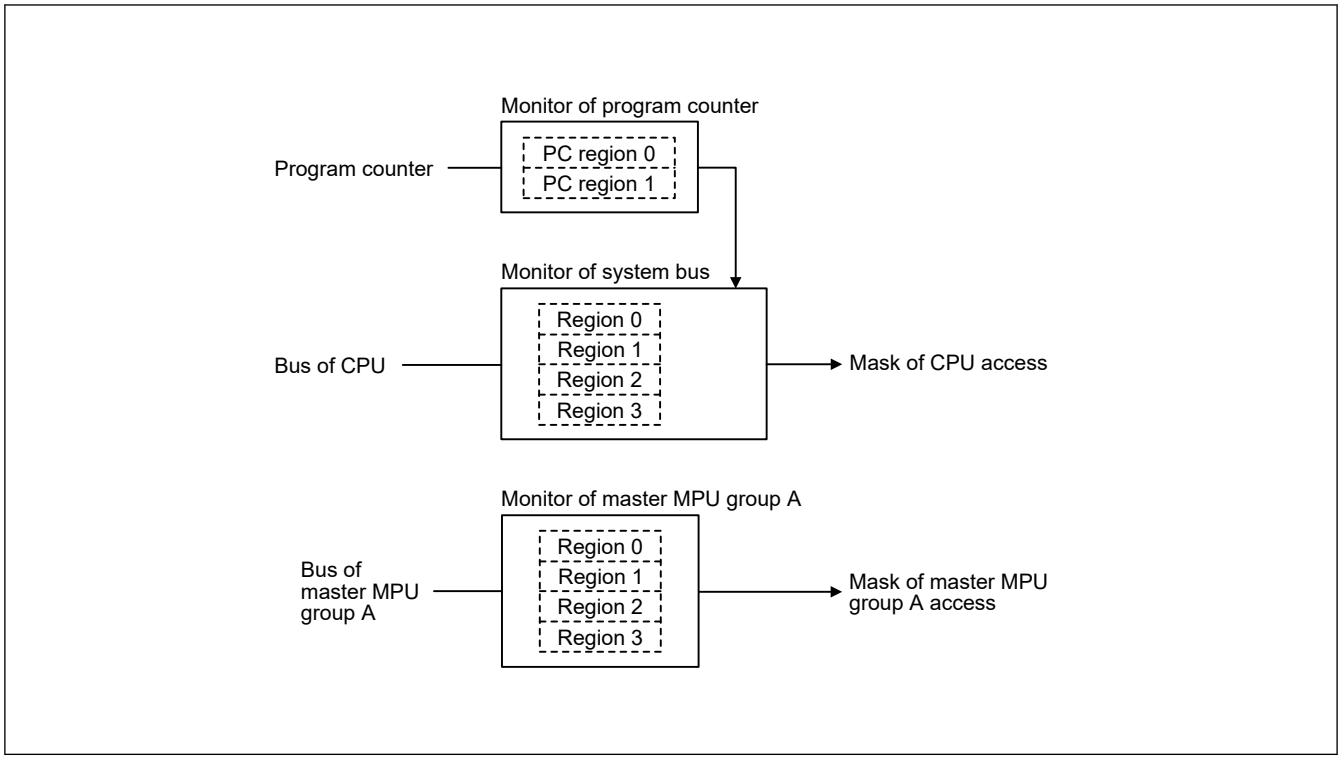


Figure 14.10 Security MPU block diagram

14.6.1 Register Descriptions (Option-Setting Memory)

All security MPU registers are option-setting memory. Option-setting memory refers to a set of registers that are available for selecting the state of the microcontroller after a reset. The option-setting memory is allocated in the code flash.

14.6.1.1 SECMPUPCSn : Security MPU Program Counter Start Address Register n (n = 0, 1)

Address: 0x0000_0408/0x0000_2408*1 (n = 0), 0x0000_0410/0x0000_2410*1 (n = 1)

Bit position:	31	0
Bit field:	SECMPUPCS[31:0]	
Value after reset:	The value set by user	

Note 1. The address of these registers will be changed when the boot swap is set.

Bit	Symbol	Function	R/W
31:0	SECMPUPCS[31:0]	Region Start Address Address where the region starts, for use in region determination. The value range is from 0x0000_0000 to 0x000F_FFFC or 0x1FF0_0000 to 0x200F_FFFC, excluding reserved areas. The lower 2 bits are read as 0. When programming to the code flash, the lower 2 bits write value should be 0.	R/W

The SECMPUPCSn and SECMPUPCEn registers specify the security fetch region of the code flash memory (0x0000_0000 to 0x000F_FFFF, not including the reserved areas) or SRAM (0x1FF0_0000 to 0x200F_FFFF, not including the reserved areas).

The secure program is executed in the memory space defined by the SECMPUPCSn and SECMPUPCEn registers and can access the secure data specified in the SECMPUSm and SECMPUEm registers (m = 0 to 3).

14.6.1.2 SECMPUPCEn : Security MPU Program Counter End Address Register n (n = 0, 1)

Address: 0x0000_040C/0x0000_240C*1 (n = 0), 0x0000_0414/0x0000_2414*1 (n = 1)

Bit position:	31																															0
Bit field:	SECMPUPCE[31:0]																															
Value after reset:	The value set by user																															

Note 1. The address of these registers will be changed when the boot swap is set.

Bit	Symbol	Function	R/W
31:0	SECMPUPCE[31:0]	Region End Address Address where the region ends, for use in region determination. The value range is from 0x0000_0003 to 0x000F_FFFF or 0x1FF0_0003 to 0x200F_FFFF, excluding reserved areas. The lower 2 bits are read as 1. When programming to the code flash, the lower 2 bits write value should be 1.	R/W

14.6.1.3 SECMPUS0 : Security MPU Region 0 Start Address Register

Address: 0x0000_0418/0x0000_2418*1

Bit position:	31							23																								0
Bit field:	—	—	—	—	—	—	—	—	SECMPUS[23:0]																								
Value after reset:	0	0	0	0	0	0	0	0	The value set by user																								

Note 1. The address of these registers will be changed when the boot swap is set.

Bit	Symbol	Function	R/W
23:0	SECMPUS[23:0]	Region Start Address Address where the region starts, for use in region determination. The value range is from 0x0000_0000 to 0x000F_FFFC, excluding reserved areas. The lower 2 bits are read as 0. When programming to the code flash, the lower 2 bits write value should be 0.	R/W
31:24	—	These bits are read as 0. When programming to the code flash, the write value should be 0.	R/W

The SECMPUS0 and SECMPUE0 registers specify the security program and data of the code flash memory (0x0000_0000 to 0x000F_FFFF, not including the reserved areas). The memory space defined in the SECMPUS0 and SECMPUE0 registers can only be accessed from the secure program set up in the SECMPUPCSn and SECMPUPCEn registers (n = 0, 1).

Setting of the vector table area is prohibited.

14.6.1.4 SECMPUE0 : Security MPU Region 0 End Address Register

Address: 0x0000_041C/0x0000_241C*1

Bit position:	31							23																								0
Bit field:	—	—	—	—	—	—	—	—	SECMPUE[23:0]																								
Value after reset:	0	0	0	0	0	0	0	0	The value set by user																								

Note 1. The address of these registers will be changed when the boot swap is set.

Bit	Symbol	Function	R/W
23:0	SECMPUE[23:0]	Region End Address Address where the region ends, for use in region determination. The value range is from 0x0000_0003 to 0x000F_FFFF, excluding reserved areas. The lower 2 bits are read as 1. When programming to the code flash, the lower 2 bits write value should be 1.	R/W
31:24	—	These bits are read as 0. When programming to the code flash, the write value should be 0.	R/W

14.6.1.5 SECMPUS1 : Security MPU Region 1 Start Address Register

Address: 0x0000_0420/0x0000_2420*1

Bit position:	31	0
Bit field:	SECMPUS[31:0]	
Value after reset:	The value set by user	

Note 1. The address of these registers will be changed when the boot swap is set.

Bit	Symbol	Function	R/W
31:0	SECMPUS[31:0]	Region Start Address Address where the region starts, for use in region determination. The value range is from 0x1FF0_0000 to 0x200F_FFFC, excluding reserved areas. The lower 2 bits are read as 0. When programming to the code flash, the lower 2 bits write value should be 0.	R/W

The SECMPUS1 and SECMPUE1 registers specify the security program and data of the SRAM (0x1FF0_0000 to 0x200F_FFFF, excluding reserved areas).

The memory space defined in the SECMPUS1 and SECMPUE1 registers can only be accessed from the secure program set up in the SECMPUPCSn and SECMPUPCEn registers (n = 0, 1).

Setting of the stack area and the vector table are prohibited.

14.6.1.6 SECMPUE1 : Security MPU Region 1 End Address Register

Address: 0x0000_0424/0x0000_2424*1

Bit position:	31	0
Bit field:	SECMPUE[31:0]	
Value after reset:	The value set by user	

Note 1. The address of these registers will be changed when the boot swap is set.

Bit	Symbol	Function	R/W
31:0	SECMPUE[31:0]	Region End Address Address where the region ends, for use in region determination. The value range is from 0x1FF0_0003 to 0x200F_FFFF, excluding reserved areas. The lower 2 bits are read as 1. When programming to the code flash, the lower 2 bits write value should be 1.	R/W

14.6.1.7 SECMPUS2 : Security MPU Region 2 Start Address Register

Address: 0x0000_0428/0x0000_2428*1

Bit position:	31	0
Bit field:	SECMPUS[31:0]	
Value after reset:	The value set by user	

Note 1. The address of these registers will be changed when the boot swap is set.

Bit	Symbol	Function	R/W
31:0	SECMPUS[31:0]	Region Start Address Address where the region starts, for use in region determination. The value range is from 0x4010_0000 to 0x407F_FFFC. The lower 2 bits are read as 0. When programming to the code flash, the lower 2 bits write value should be 0.	R/W

The SECMPUS2 and SECMPUE2 registers specify the secure region of the security function (0x4010_0000 to 0x407F_FFFF). The memory space defined in the SECMPUS2 and SECMPUE2 registers can only be accessed from the secure program set up in the SECMPUPCSn and SECMPUPCEn registers (n = 0, 1).

14.6.1.8 SECMPUE2 : Security MPU Region 2 End Address Register

Address: 0x0000_042C/0x0000_242C*1

Bit position:	31	0
Bit field:	SECMPUE[31:0]	
Value after reset:	The value set by user	

Note 1. The address of these registers will be changed when the boot swap is set.

Bit	Symbol	Function	R/W
31:0	SECMPUE[31:0]	Region End Address Address that determines where the region ends. The value range is from 0x4010_0003 to 0x407F_FFFF. The lower 2 bits are read as 1. When programming to the code flash, the lower 2 bits write value should be 1.	R/W

14.6.1.9 SECMPUS3 : Security MPU Region 3 Start Address Register

Address: 0x0000_0430/0x0000_2430*1

Bit position:	31	0
Bit field:	SECMPUS[31:0]	
Value after reset:	The value set by user	

Note 1. The address of these registers will be changed when the boot swap is set.

Bit	Symbol	Function	R/W
31:0	SECMPUS[31:0]	Region Start Address Address where the region starts, for use in region determination. The value range is from 0x4010_0000 to 0x407F_FFFC. The lower 2 bits are read as 0. When programming to the code flash, the lower 2 bits write value should be 0.	R/W

The SECMPUS3 and SECMPUE3 registers specify the secure region of the security function (0x4010_0000 to 0x407F_FFFF). The memory space defined in the SECMPUS3 and SECMPUE3 registers can only be accessed from the secure program set up in the SECMPUPCSn and SECMPUPCEn registers (n = 0, 1).

14.6.1.10 SECMPUE3 : Security MPU Region 3 End Address Register

Address: 0x0000_0434/0x0000_2434*1

Bit position:	31	0
Bit field:	SECMPUE[31:0]	
Value after reset:	The value set by user	

Note 1. The address of these registers will be changed when the boot swap is set.

Bit	Symbol	Function	R/W
31:0	SECMPUE[31:0]	Region End Address Address where the region ends, for use in region determination. The value range is from 0x4010_0003 to 0x407F_FFFF. The lower 2 bits are read as 1. When programming to the code flash, the lower 2 bits write value should be 1.	R/W

14.6.1.11 SECMPUAC : Security MPU Access Control Register

Address: 0x0000_0438/0x0000_2438*1

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	DISPC1	DISPC0	—	—	—	—	DIS3	DIS2	DIS1	DIS0
Value after reset:	1	1	1	1	1	1	The value set by user		1	1	1	1	The value set by user			

Note 1. The address of these registers will be changed when the boot swap is set.

Bit	Symbol	Function	R/W
0	DIS0	Region 0 Disable 0: Security MPU region 0 enabled 1: Security MPU region 0 disabled	R/W
1	DIS1	Region 1 Disable 0: Security MPU region 1 enabled 1: Security MPU region 1 disabled	R/W
2	DIS2	Region 2 Disable 0: Security MPU region 2 enabled 1: Security MPU region 2 disabled	R/W
3	DIS3	Region 3 Disable 0: Security MPU region 3 enabled 1: Security MPU region 3 disabled	R/W
7:4	—	These bits are read as 1. When programming to the code flash, the write value should be 1.	R/W
8	DISPC0	PC Region 0 Disable 0: Security MPU PC region 0 enabled 1: Security MPU PC region 0 disabled	R/W
9	DISPC1	PC Region 1 Disable 0: Security MPU PC region 1 enabled 1: Security MPU PC region 1 disabled	R/W
15:10	—	These bits are read as 1. When programming to the code flash, the write value should be 1.	R/W
31:16	—	These bits are read as 1. When programming to the code flash, the write value should be 1.	R/W

Note: When flash memory is erased, the settings of the security MPU is disabled.
Note: To enable or disable the security MPU, see section 14.6.2. Memory Protection.

DIS0 bit (Region 0 Disable)

The DIS0 bit enables or disables the security MPU region 0. If security MPU region 0 is enabled, the code flash memory region within the limits set up by SECMPUS0 and SECMPUE0 is secure data.

DIS1 bit (Region 1 Disable)

The DIS1 bit enables or disables the security MPU region 1. If security MPU region 1 is enabled, the SRAM region within the limits set up by SECMPUS1 and SECMPUE1 is secure data.

DIS2 bit (Region 2 Disable)

The DIS2 bit enables or disables the security MPU region 2. If security MPU region 2 is enabled, the data of the security function region within the limits set up by SECMPUS2 and SECMPUE2 is secure data.

DIS3 bit (Region 3 Disable)

The DIS3 bit enables or disables the security MPU region 3. If security MPU region 3 is enabled, the data of the security function region within the limits set up by SECMPUS3 and SECMPUE3 is secure data.

DISPC0 bit (PC Region 0 Disable)

The DISPC0 bit enables or disables the security MPU PC region 0. If security MPU PC region 0 is enabled, the code flash memory or the SRAM region within the limits set up by SECMPUPCS0 and SECMPUPCE0 contains a secure program.

DISPC1 bit (PC Region 1 Disable)

The DISPC1 bit enables or disables the security MPU PC region 1. If security MPU PC region 1 is enabled, the code flash memory or the SRAM region within the limits set up by SECMPUPCS1 and SECMPUPCE1 contains a secure program.

14.6.2 Memory Protection

The security MPU protects the secured regions (the code flash memory, the SRAM, and the security functions) from being accessed by non-secure programs. If access to a protected region is detected, the access becomes invalid.

When the security MPU is enabled, DISPC0 or DISPC1 in the Security MPU Access Control Register (SECMPUAC) and DIS0, DIS1, DIS2, or DIS3 in the Security MPU Access Control Register (SECMPUAC) must be set to 0.

When the security MPU is disabled, all bits in DISPC0, DISPC1, DIS0, DIS1, DIS2, and DIS3 in the Security MPU Access Control Register (SECMPUAC) must be set to 1.

Other settings in the Security MPU Access Control Register (SECMPUAC) are prohibited.

The security MPU provides access protection in the following conditions:

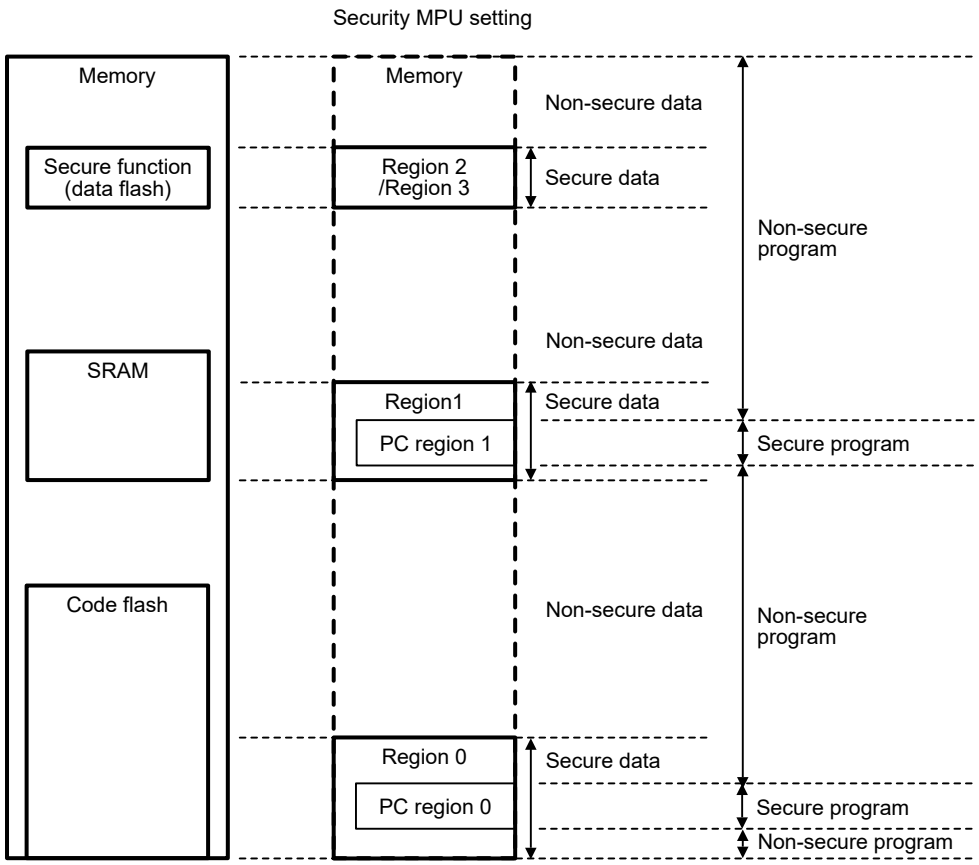
- Secure data is accessed from a non-secure program
- Secure data is accessed from other than the CPU (DTC)
- Secure data is accessed from the debugger.

Secure data is accessible only from a secure program.

Note: Secure program:
Code flash or SRAM region within the limits set up by SECMPUPCS0 and SECMPUPCE0.
Code flash or SRAM region within the limits set up by SECMPUPCS1 and SECMPUPCE1.

Non-secure program:
All regions without the secure program.

Secure data:
Code flash region within the limits set up by SECMPUS0 and SECMPUE0.
SRAM region within the limits set up by SECMPUS1 andSECMPUE1.
Security Function region within the limits set up by SECMPUS2 and SECMPUE2.
Security Function region within the limits set up by SECMPUS3 and SECMPUE3.



Note: Secure program in code flash (PC region 0) can access all data (secure data and non-secure data).
Note: Secure program in SRAM (PC region 1) can access all data (secure data and non-secure data).
Note: Non-secure program (not PC region 0 nor PC region 1) cannot access secure data (region 0, region 1, region 2, and region 3).
Note: Non-secure program (not PC region 0 nor PC region 1) can access non-secure data.

Figure 14.11 Use case of security MPU

14.7 Usage Notes

14.7.1 Notes on the Use of a Debugger

The memory cannot be debugged if the security MPU is enabled. Disable the security MPU when debug a program, OCD debug only valid when SECMPUAC register is 0xFFFF_FFFF.

14.8 References

1. *ARM®v8-M Architecture Reference Manual* (ARM DDI 0553B.a)
2. *ARM® Cortex®-M23 Processor Technical Reference Manual* (ARM DDI 0550C)
3. *ARM® Cortex®-M23 Processor User Guide* (ARM DUI 0963B)

15. Data Transfer Controller (DTC)

15.1 Overview

A Data Transfer Controller (DTC) module is provided for transferring data when activated by an interrupt request.

Table 15.1 lists the DTC specifications and Figure 15.1 shows DTC block diagram.

Table 15.1 DTC specifications

Parameter	Description
Transfer modes	• Normal transfer mode A single activation leads to a single data transfer. • Repeat transfer mode A single activation leads to a single data transfer. The transfer address returns to the start address after the number of data transfers reaches the specified repeat size. The maximum number of repeat transfers is 256 and the maximum data transfer size is 256 × 32 bits (1024 bytes) • Block transfer mode A single activation leads to a transfer of a single block. The maximum block size is 256 × 32 bits = 1024 bytes.
Transfer channel	• Channel transfer can be associated with the interrupt source (transferred by a DTC activation request from the ICU) • Multiple data units can be transferred on a single activation source (chain transfer) • Chain transfers are selectable to either execute when the counter is 0, or always execute.
Transfer space	• 4 GB area from 0x0000_0000 to 0xFFFF_FFFF, excluding reserved areas
Data transfer units	• Single data unit: 1 byte (8 bits), 1 halfword (16 bits), 1 word (32 bits) • Single block size: 1 to 256 data units.
CPU interrupt source	• An interrupt request can be generated to the CPU on a DTC activation interrupt • An interrupt request can be generated to the CPU after a single data transfer • An interrupt request can be generated to the CPU after a data transfer of a specified volume.
Event link function	An event link request is generated after one data transfer (for block, after one block transfer)
Read skip	Read of transfer information can be skipped
Write-back skip	When the transfer source or destination address is specified as fixed, a write-back of transfer information can be skipped
Module-stop function	Module-stop state can be set to reduce power consumption

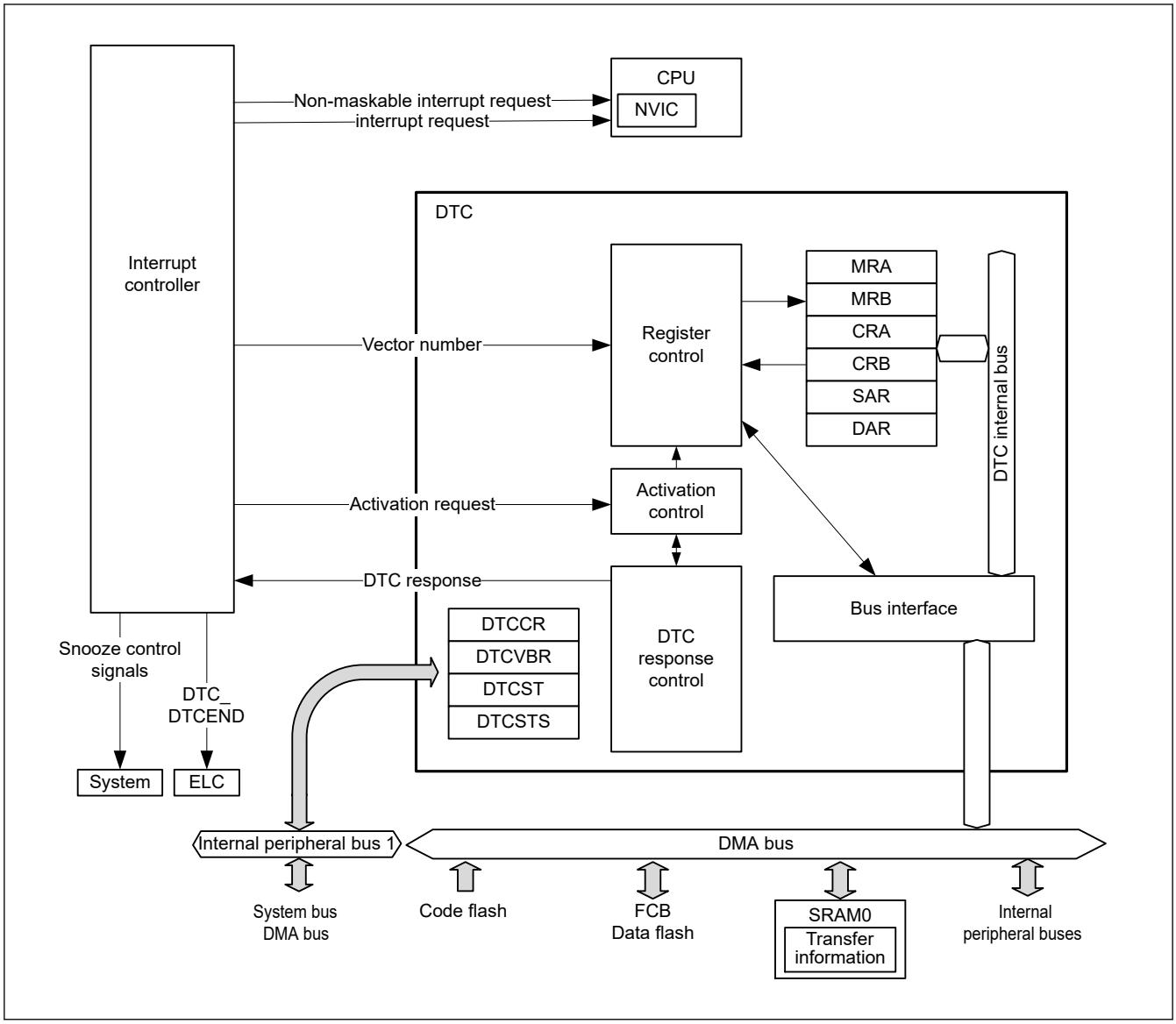


Figure 15.1 DTC block diagram

See section 12.1. Overview in section 12, Interrupt Controller Unit (ICU) for the connections between the DTC and NVIC in the CPU.

15.2 Register Descriptions

MRA, MRB, SAR, DAR, CRA, and CRB are all DTC internal registers that cannot be directly accessed from the CPU. Values to be set in these DTC internal registers are placed in the SRAM area as transfer information. When an activation request is generated, the DTC reads the transfer information from the SRAM area and sets it in its internal registers. After the data transfer ends, the internal register contents are written back to the SRAM area as transfer information.

15.2.1 MRA : DTC Mode Register A

Base address: DTCVBR

Offset address: 0x03 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 15.3.1. Allocating Transfer Information and DTC Vector Table)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MD[1:0]		SZ[1:0]		SM[1:0]		—	—
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
1:0	—	The read values are undefined. The write value should be 0.	—
3:2	SM[1:0]	Transfer Source Address Addressing Mode 0 0: Address in the SAR register is fixed (write-back to SAR is skipped.) 0 1: Address in the SAR register is fixed (write-back to SAR is skipped.) 1 0: SAR value is incremented after data transfer: +1 when SZ[1:0] = 00b +2 when SZ[1:0] = 01b +4 when SZ[1:0] = 10b 1 1: SAR value is decremented after data transfer: -1 when SZ[1:0] = 00b -2 when SZ[1:0] = 01b -4 when SZ[1:0] = 10b	—
5:4	SZ[1:0]	DTC Data Transfer Size 0 0: Byte (8-bit) transfer 0 1: Halfword (16-bit) transfer 1 0: Word (32-bit) transfer 1 1: Setting prohibited	—
7:6	MD[1:0]	DTC Transfer Mode Select 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited	—

The MRA register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0x03) and DTC transfers it automatically to and from the MRA register. See section 15.3.1. Allocating Transfer Information and DTC Vector Table.

15.2.2 MRB : DTC Mode Register B

Base address: DTCVBR

Offset address: 0x02 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 15.3.1. Allocating Transfer Information and DTC Vector Table)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CHNE	CHNS	DISEL	DTS	DM[1:0]		—	—
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
1:0	—	The read values are undefined. The write value should be 0.	—
3:2	DM[1:0]	Transfer Destination Address Addressing Mode 0 0: Address in the DAR register is fixed (write-back to DAR is skipped) 0 1: Address in the DAR register is fixed (write-back to DAR is skipped) 1 0: DAR value is incremented after data transfer: +1 when MRA.SZ[1:0] = 00b +2 when SZ[1:0] = 01b +4 when SZ[1:0] = 10b 1 1: DAR value is decremented after data transfer: -1 when MRA.SZ[1:0] = 00b -2 when SZ[1:0] = 01b -4 when SZ[1:0] = 10b	—
4	DTS	DTC Transfer Mode Select 0: Select transfer destination as repeat or block area. 1: Select transfer source as repeat or block area.	—
5	DISEL	DTC Interrupt Select 0: Generate an interrupt request to the CPU when specified data transfer is complete. 1: Generate an interrupt request to the CPU each time DTC data transfer is performed.	—

Bit	Symbol	Function	R/W
6	CHNS	DTC Chain Transfer Select 0: Chain transfer is continuous. 1: Chain transfer occurs only when the transfer counter changes from 1 to 0 or 1 to CRAH.	—
7	CHNE	DTC Chain Transfer Enable 0: Chain transfer is disabled. 1: Chain transfer is enabled.	—

The MRB register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0x02) and DTC transfers it automatically to and from the MRB register. See section 15.3.1. Allocating Transfer Information and DTC Vector Table.

DM[1:0] bits (Transfer Destination Address Addressing Mode)

The DM[1:0] bits are to fix the address of the DAR register or specify increment / decrement of the DAR register after transfer.

DTS bit (DTC Transfer Mode Select)

The DTS bit specifies whether the transfer source or destination is the repeat or block area in repeat or block transfer mode.

DISEL bit (DTC Interrupt Select)

The DISEL bit specifies the condition for generating an interrupt request to the CPU.

CHNS bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition. When CHNE is 0, the CHNS setting is ignored. For details on the conditions for chain transfer, see Table 15.3.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the activation source flag is not cleared, and an interrupt request to the CPU is not generated.

CHNE bit (DTC Chain Transfer Enable)

The CHNE bit enables chain transfer. The chain transfer condition is selected by the CHNS bit. For details on chain transfer, see section 15.4.6. Chain Transfer.

15.2.3 SAR : DTC Transfer Source Register

Base address: DTCVBR

Offset address: 0x04 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 15.3.1. Allocating Transfer Information and DTC Vector Table)

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

The SAR sets the transfer source start address and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x04) and DTC transfers it automatically to and from the SAR register. See section 15.3.1. Allocating Transfer Information and DTC Vector Table.

Misalignment is prohibited for DTC transfers. Bit[0] must be 0 when MRA.SZ[1:0] = 01b, and bit[1] and bit[0] must be 0 when MRA.SZ[1:0] = 10b.

15.2.4 DAR : DTC Transfer Destination Register

Base address: DTCVBR

Offset address: 0x08 + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 15.3.1. Allocating Transfer Information and DTC Vector Table)

Bit position:	31																												0
Bit field:																													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

The DAR sets the transfer destination start address and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x08) and DTC transfers it automatically to and from the DAR register. See section 15.3.1. Allocating Transfer Information and DTC Vector Table.

Misalignment is prohibited for DTC transfers. Bit[0] must be 0 when MRA.SZ[1:0] = 01b, and bit[1] and bit[0] must be 0 when MRA.SZ[1:0] = 10b.

15.2.5 CRA : DTC Transfer Count Register A

Base address: DTCVBR

Offset address: 0x0E + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 15.3.1. Allocating Transfer Information and DTC Vector Table)

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

Bit	Symbol	Function	R/W
7:0	CRAL	Transfer Counter A Lower Register Specify the transfer count.	—
15:8	CRAH	Transfer Counter A Upper Register Specify the transfer count.	—

Note: The function depends on the transfer mode.
Note: Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

The CRA register consists of 16 bits. CRAL is the lower 8 bits and CRAH is the upper 8 bits. CRA is used in normal mode.

CRAL and CRAH are used in repeat transfer mode and block transfer mode.

The CRA register cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x0E) and DTC transfers it automatically to and from the CRA register. See section 15.3.1. Allocating Transfer Information and DTC Vector Table.

(1) Normal transfer mode (MRA.MD[1:0] = 00b)

In normal transfer mode, CRA functions as a 16-bit transfer counter. The transfer count is 1, 65535, and 65536 when the set value is 0x0001, 0xFFFF, and 0x0000, respectively. The CRA value is decremented (-1) on each data transfer.

(2) Repeat transfer mode (MRA.MD[1:0] = 01b)

In repeat transfer mode, the CRAH register holds the transfer count and the CRAL register functions as an 8-bit transfer counter. The transfer count is 1, 255, and 256 when the set value is 0x01, 0xFF, and 0x00, respectively. The CRAL value is decremented (-1) on each data transfer. When it reaches 0x00, the CRAH value is transferred to CRAL.

(3) Block transfer mode (MRA.MD[1:0] = 10b)

In block transfer mode, the CRAH register holds the block size and the CRAL register functions as an 8-bit block size counter. The transfer count is 1, 255, and 256 when the set value is 0x01, 0xFF, and 0x00, respectively. The CRAL value is decremented (-1) on each data transfer. When it reaches 0x00, the CRAH value is transferred to CRAL.

15.2.6 CRB : DTC Transfer Count Register B

Base address: DTCVBR

Offset address: 0x0C + 0x4 × Vector number
(Inaccessible directly from the CPU. See section 15.3.1. Allocating Transfer Information and DTC Vector Table)

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

The CRB sets the block transfer count for block transfer mode. The transfer count is 1, 65535, and 65536 when the set value is 0x0001, 0xFFFF, and 0x0000, respectively. The CRB value is decremented (-1) when the final data of a single block size is transferred. When normal transfer mode or repeat transfer mode is selected, this register is not used, and the set value is ignored.

The CRB cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x0C) and DTC transfers it automatically to and from the CRB register. See section 15.3.1. Allocating Transfer Information and DTC Vector Table.

15.2.7 DTCCR : DTC Control Register

Base address: DTC = 0x4000_5400

Offset address: 0x00

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

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
4	RRS	DTC Transfer Information Read Skip Enable 0: Transfer information read is not skipped 1: Transfer information read is skipped when vector numbers match	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

RRS bit (DTC Transfer Information Read Skip Enable)

The RRS bit enables skipping of transfer information reads when vector numbers match. The DTC vector number is compared with the vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transfer information. However, when the previous transfer is a chain transfer, the transfer information is read regardless of the RRS bit.

When the transfer counter (CRA register) becomes 0 during the previous normal transfer and when the transfer counter (CRB register) becomes 0 during the previous block transfer, the transfer information is read regardless of the RRS bit value.

15.2.8 DTCVBR : DTC Vector Base Register

Base address: DTC = 0x4000_5400

Offset address: 0x04

Bit position:	31	0
Bit field:		
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	n/a	DTC Vector Base Address Set the DTC vector base address. The lower 10 bits should be 0.	R/W

The DTCVBR sets the base address for calculating the DTC vector table address, which can be set in the range of 0x0000_0000 to 0xFFFF_FFFF (4 GB) in 1-KB units.

15.2.9 DTCST : DTC Module Start Register

Base address: DTC = 0x4000_5400

Offset address: 0x0C

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

Bit	Symbol	Function	R/W
0	DTCST	DTC Module Start 0: DTC module stopped. 1: DTC module started.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

DTCST bit (DTC Module Start)

Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When this bit is set to 0, transfer requests are no longer accepted. If this bit is set to 0 during a data transfer, the accepted transfer request is active until processing completes.

DTCST must be set to 0 before transitioning to one of the following state or mode:

- Module-stop state
- Software Standby mode without Snooze mode transition

For details on these transitions, see section 15.9. Low Power Consumption Function and section 10, Low Power Modes.

15.2.10 DTCSTS : DTC Status Register

Base address: DTC = 0x4000_5400

Offset address: 0x0E

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

Bit	Symbol	Function	R/W
7:0	VECN[7:0]	DTC-Activating Vector Number Monitoring These bits indicate the vector number for the activation source when a DTC transfer is in progress. The value is only valid if a DTC transfer is in progress (ACT flag is 1).	R
14:8	—	These bits are read as 0.	R
15	ACT	DTC Active Flag 0: DTC transfer operation is not in progress. 1: DTC transfer operation is in progress.	R

VECN[7:0] bits (DTC-Activating Vector Number Monitoring)

While transfer by the DTC is in progress, the VECN[7:0] bits indicate the vector number associated with the activation source for the transfer. The value read from the VECN[7:0] bits is valid if the ACT flag is 1, indicating a DTC transfer in progress, and invalid if the ACT flag is 0, indicating no DTC transfer is in progress.

ACT flag (DTC Active Flag)

The ACT flag indicates the state of the DTC transfer operation.

[Setting condition]

- When the DTC is activated by a transfer request.

[Clearing condition]

- When transfer by the DTC, in response to a transfer request, is complete.

15.3 Activation Sources

The DTC is activated by an interrupt request. Setting the ICU.IELSRn.DTCE bit to 1 enables activation of the DTC by the associated interrupt. The selector output *n* number set in ICU.IELSRn is defined as the interrupt vector number, where n = 0 to 31. For an enabled interrupt, the specific DTC interrupt source associated with each interrupt vector number *n* is selected in ICU.IELSRn.IELS[4:0] where n = 0 to 31, as listed in section 12.3.2. Event Number in section 12, Interrupt Controller Unit (ICU). For activation by software, see section 16.2.2. ELSEGRn : Event Link Software Event Generation Register n (n = 0, 1).

The interrupt vector number is equivalent to the DTC vector table number. After the DTC accepted an activation request, it does not accept another activation request until the transfer for that single request is complete, regardless of the priority of the requests. When multiple activation requests are generated during a DTC transfer, the highest priority request is accepted on completion of the transfer. When multiple activation requests are generated while the DTC Module Start bit (DTCST.DTCST) is 0, the DTC accepts the highest priority request when DTCST.DTCST is subsequently set to 1. The smaller interrupt vector number has higher priority.

The DTC performs the following operations at the start of a single data transfer or for a chain transfer, after the last of the consecutive transfers:

- On completion of a specified round of data transfer, the ICU.IELSRn.DTCE bit is set to 0, and an interrupt request is sent to the CPU.
- If the MRB.DISEL bit is 1, an interrupt request is sent to the CPU on completion of a data transfer.
- For other transfers, the ICU.IELSRn.IR flag of the activation source is set to 0 at the start of the data transfer.

15.3.1 Allocating Transfer Information and DTC Vector Table

The DTC reads the start address of the transfer information associated with each activation source from the vector table and reads the transfer information starting at that address.

The vector table must be located so that the lower 10 bits of the base address (start address) are 0. Use the DTC Vector Base Register (DTCVBR) to set the base address of the DTC vector table. Transfer information is allocated in the SRAM area. In the SRAM area, the start address of the transfer information *n* with vector number *n* must be 4n added to the base address in the vector table.

Figure 15.2 shows the relationship between the DTC vector table and transfer information. Figure 15.3 shows the allocation of transfer information in the SRAM area.

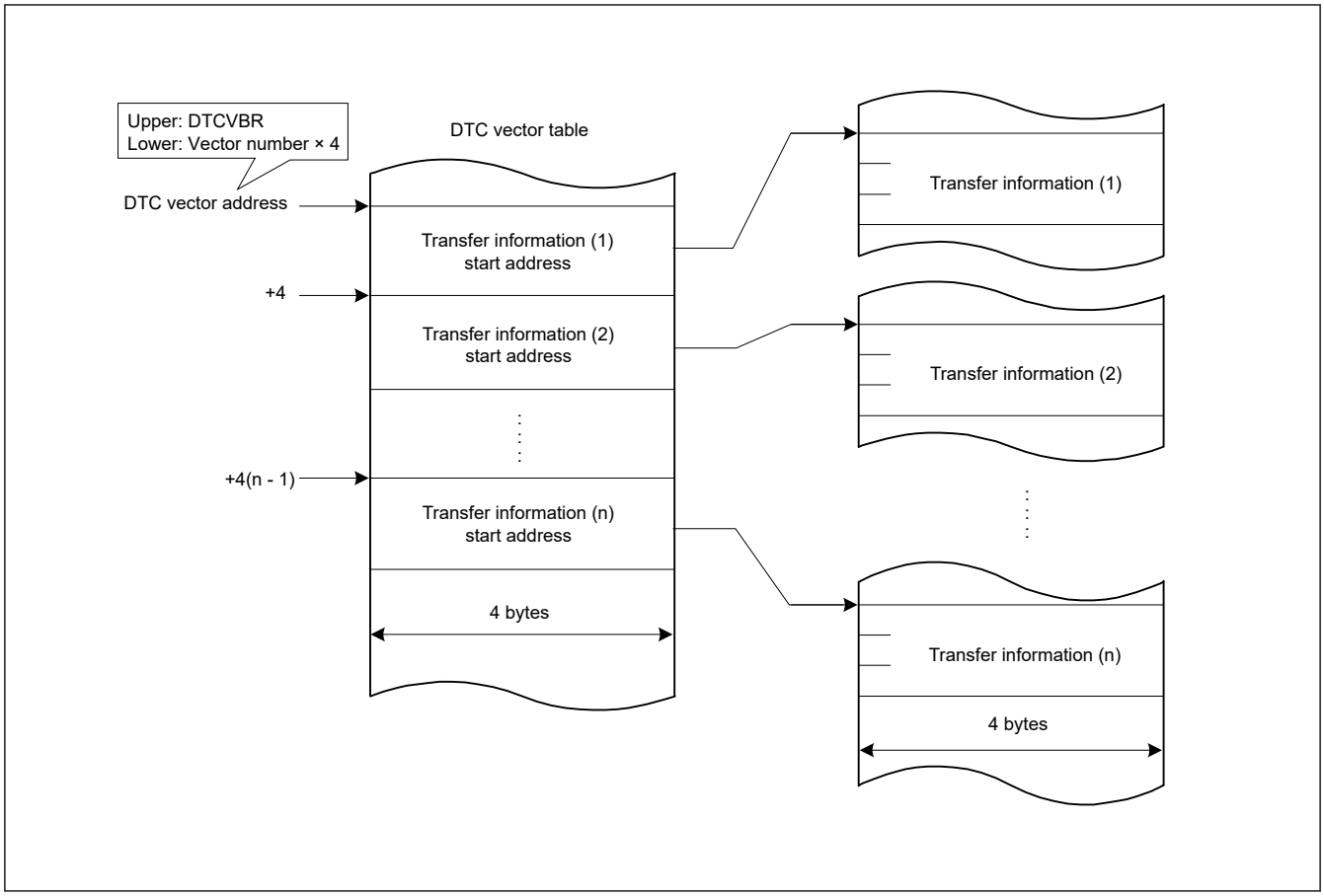


Figure 15.2 DTC vector table and transfer information

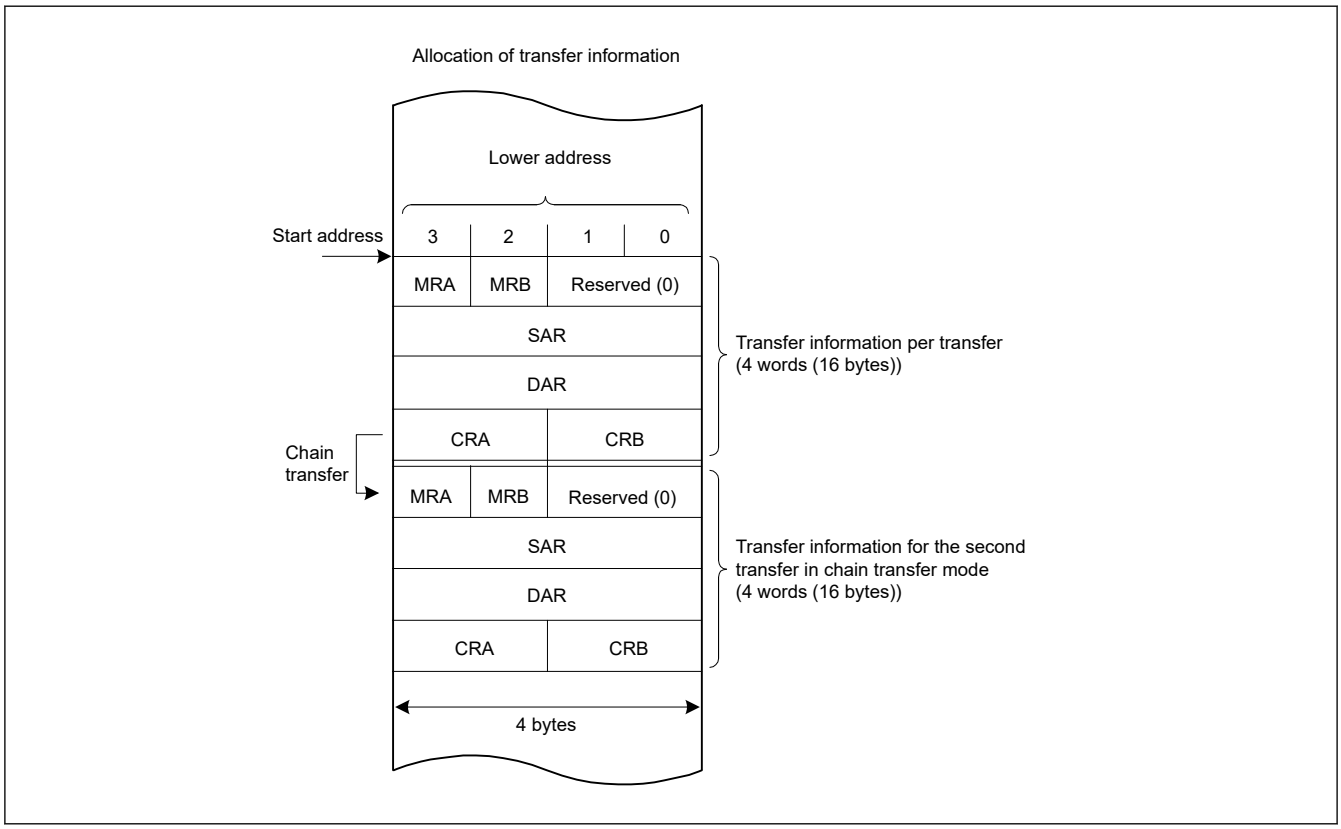


Figure 15.3 Allocation of transfer information in the SRAM area

15.4 Operation

The DTC transfers data according to the transfer information. Storage of the transfer information in the SRAM area is required before a DTC operation. When the DTC is activated, it reads the DTC vector associated with the vector number. The DTC reads the transfer information from the transfer information store address referenced by the DTC vector and transfers the data. After the data transfer, the DTC writes back the transfer information. Storing the transfer information in the SRAM area allows data transfer of any number of channels.

The transfer modes include:

- Normal transfer mode
- Repeat transfer mode
- Block transfer mode.

The DTC specifies a transfer source address in the SAR register and a transfer destination address in the DAR register. The values of these registers are incremented, decremented, or address-fixed independently after the data transfer.

Table 15.2 describes the DTC transfer modes.

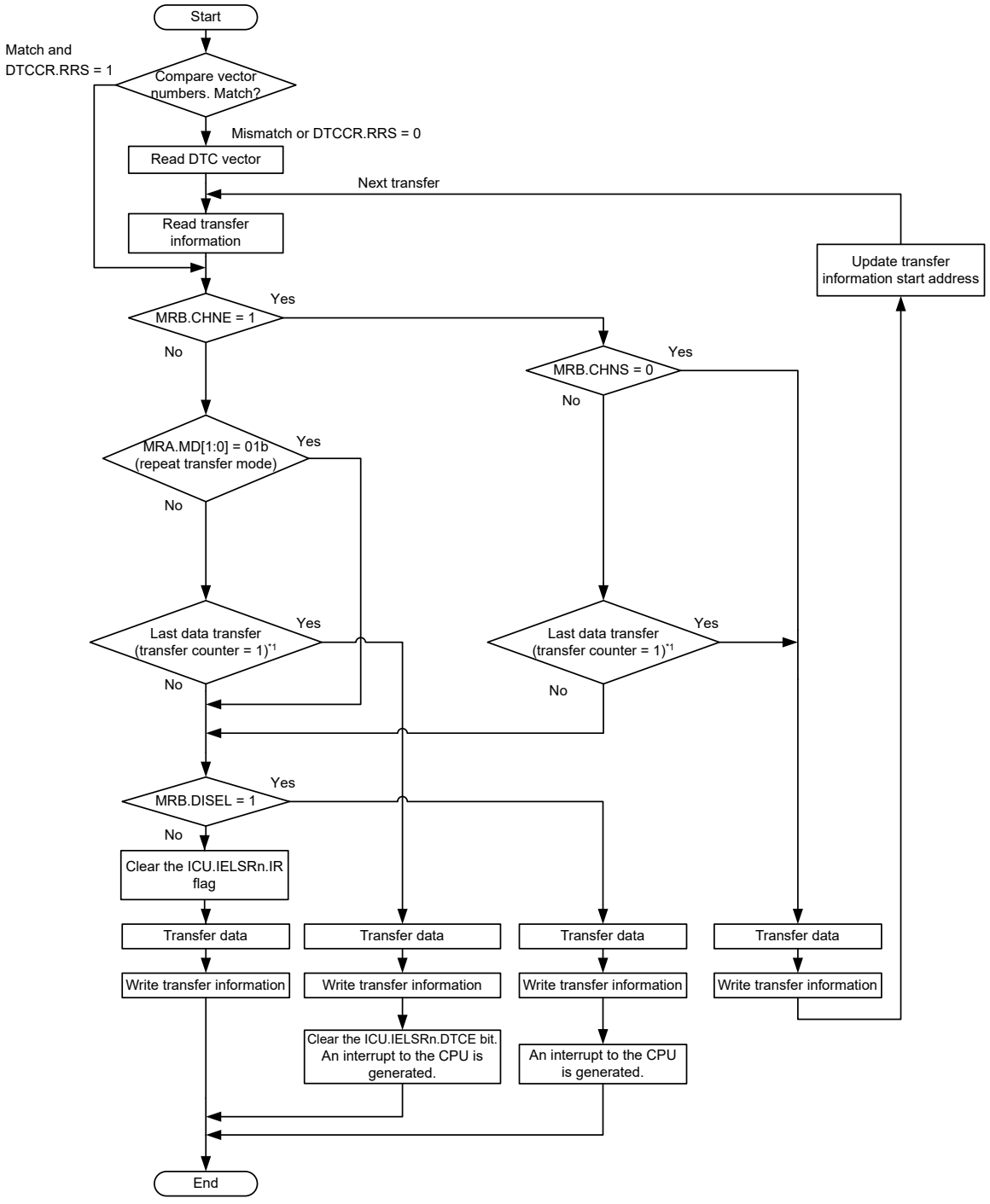
Table 15.2 DTC transfer modes

Transfer mode	Data size transferred on single transfer request	Increment or decrement of memory address	Settable transfer count
Normal transfer mode	1 byte (8 bit), 1 halfword (16 bit), 1 word (32 bit)	Incremented or decremented by 1, 2, or 4 or address-fixed	1 to 65536
Repeat transfer mode*1	1 byte (8 bit), 1 halfword (16 bit), 1 word (32 bit)	Incremented or decremented by 1, 2, or 4 or address-fixed	1 to 256*3
Block transfer mode*2	Block size specified in CRAH (1 to 256 bytes, 1 to 256 halfwords (2 to 512 bytes), or 1 to 256 words (4 to 1024 bytes))	Incremented or decremented by 1, 2, or 4 or address-fixed	1 to 65536

Note 1. Set the transfer source or transfer destination as the repeat area.
Note 2. Set the transfer source or transfer destination as the block area.
Note 3. After a data transfer of the specified count, the initial state is restored and operation restarts.

Setting the MRB.CHNE bit to 1 allows multiple transfers or chain transfer on a single activation source. It also enables a chain transfer when the specified data transfer is complete.

Figure 15.4 shows the operation flow of the DTC. Table 15.3 lists the chain transfer conditions. The combination of control information for the second and subsequent transfers are omitted in this table.



Note 1. Counter value before starting data transfer.

Figure 15.4 DTC operation flow

Table 15.3 Chain transfer conditions

First transfer				Second transfer*3				
CHNE bit	CHNS bit	DISEL bit	Transfer counter*1 *2	CHNE bit	CHNS bit	DISEL bit	Transfer counter*1 *2	DTC transfer
0	—	0	Other than (1 → 0)	—	—	—	—	Ends after the first transfer
0	—	0	(1 → 0)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU
0	—	1	—	—	—	—	—	
1	0	—	—	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	0	Other than (1 → *)	—	—	—	—	Ends after the first transfer
1	1	—	(1 → *)	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	1	Other than (1 → *)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU

Note 1. The transfer counter used depends on the transfer modes as follows:
Normal transfer mode — CRA register
Repeat transfer mode — CRAL register
Block transfer mode — CRB register

Note 2. On completion of a data transfer, the counters operate as follows:
1 → 0 in normal and block transfer modes
1 → CRAH in repeat transfer mode
(1 → *) in the table indicates both of these two operations, depending on the mode.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The conditions for the combination of the second transfer and CHNE = 1 is omitted.

15.4.1 Transfer Information Read Skip Function

Reading of vector addresses and transfer information can be skipped by setting the DTCCR.RRS bit. When a DTC activation request is generated, the current DTC vector number is compared with the DTC vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, the DTC data transfer is performed without reading the vector address and transfer information. However, when the previous transfer is a chain transfer, the vector address and transfer information are read. Additionally, when the transfer counter (CRA register) becomes 0 during the previous normal transfer, and when the transfer counter (CRB register) becomes 0 during the previous block transfer, transfer information is read regardless of the RRS bit. Figure 15.12 shows an example when reading the transfer information is skipped.

To update the vector table and transfer information, set the RRS bit to 0, update the vector table and transfer information, then set the RRS bit to 1. The stored vector number is discarded by setting the RRS bit to 0. The updated DTC vector table and transfer information are read in the next activation process.

15.4.2 Transfer Information Write-Back Skip Function

When the MRA.SM[1:0] bits or the MRB.DM[1:0] bits are set to address fixed, a part of the transfer information is not written back. Table 15.4 lists the transfer information write-back skip conditions and the associated registers. The CRA and CRB registers are written back, and the write-back of the MRA and MRB registers is skipped.

Table 15.4 Transfer information write-back skip conditions and applicable registers

MRA.SM[1:0] bits		MRB.DM[1:0] bits			
b3	b2	b3	b2	SAR register	DAR register
0	0	0	0	Skip	Skip
0	0	0	1		
0	1	0	0		
0	1	0	1		
0	0	1	0	Skip	Write-back
0	0	1	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	Write-back	Skip
1	0	0	1		
1	1	0	0		
1	1	0	1		
1	0	1	0	Write-back	Write-back
1	0	1	1		
1	1	1	0		
1	1	1	1		

15.4.3 Normal Transfer Mode

The normal transfer mode allows a 1-byte (8 bit), 1-halfword (16 bit), 1-word (32 bit) data transfer on a single activation source. The transfer count can be set from 1 to 65536. Transfer source and destination addresses can be independently set to increment, decrement, or fixed. This mode enables an interrupt request to the CPU to be generated at the end of a specified-count transfer.

Table 15.5 lists register functions in normal transfer mode, and Figure 15.5 shows the memory map of normal transfer mode.

Table 15.5 Register functions in normal transfer mode

Register	Description	Value written back by writing transfer information
SAR	Transfer source address	Increment, decrement, or fixed*1
DAR	Transfer destination address	Increment, decrement, fixed*1
CRA	Transfer counter A	CRA - 1
CRB	Transfer counter B	Not updated

Note 1. Write-back operation is skipped in address-fixed mode.

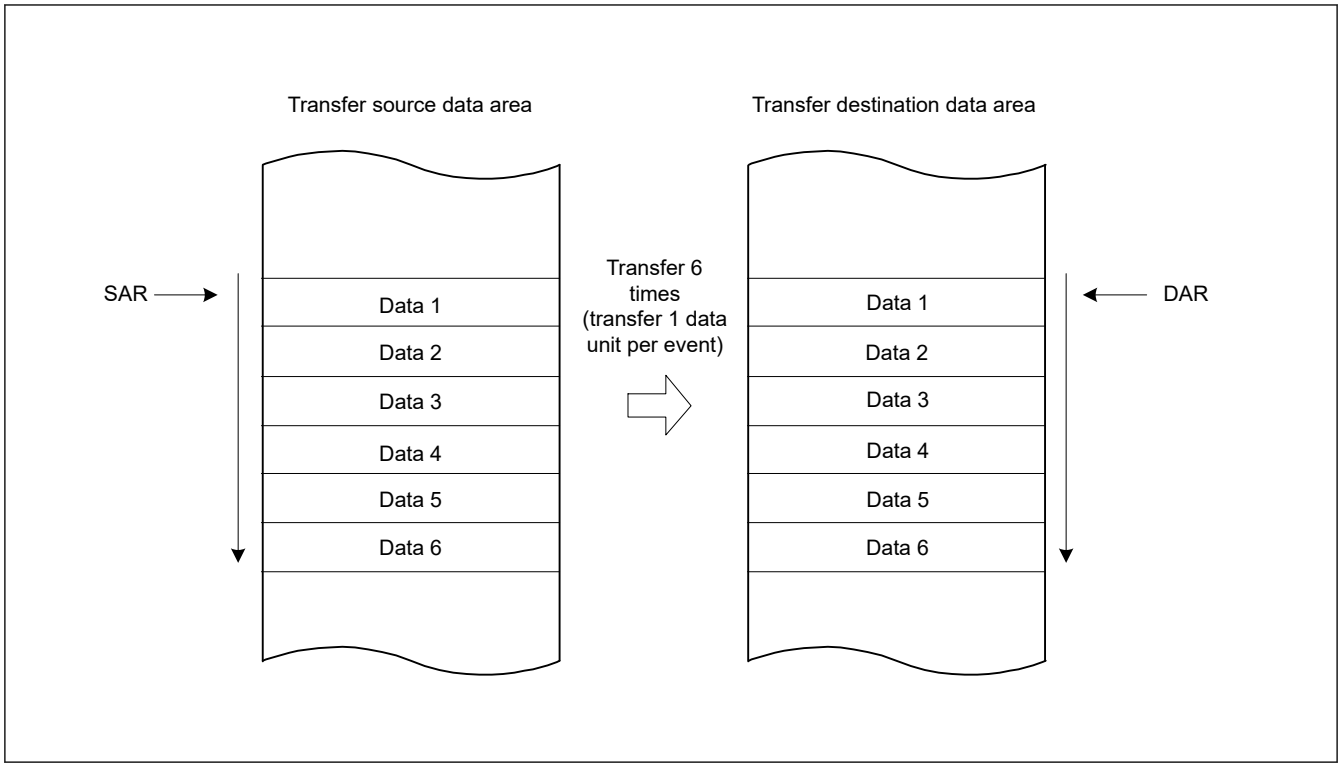


Figure 15.5 Memory map of normal transfer mode (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRA = 0x0006)

15.4.4 Repeat Transfer Mode

The repeat transfer mode allows a 1-byte (8-bit), 1-halfword (16-bit), or 1-word (32-bit) data transfer on a single activation source. Transfer source or transfer destination for the repeat area must be specified in the MRB.DTS bit. The transfer count can be set from 1 to 256. When the specified transfer count is complete, the initial value of the address register specified in the repeat area is restored, the initial value of the transfer counter is restored, and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.

When the transfer counter CRAL decrements to 0x00 in repeat transfer mode, the CRAL value is updated to the value set in the CRAH register. As a result, the transfer counter does not clear to 0x00, which disables interrupt requests to the CPU when the MRB.DISEL bit is set to 0. An interrupt request to the CPU is generated when the specified data transfer completes.

Table 15.6 lists the register functions in repeat transfer mode, and Figure 15.6 shows the memory map of repeat transfer mode.

Table 15.6 Register functions in repeat transfer mode

Register	Description	Value written back by writing transfer information	
		When CRAL is not 1	When CRAL is 1
SAR	Transfer source address	Increment, decrement, fixed*1	• When the MRB.DTS bit is 0 Increment, decrement, or fixed*1 • When the MRB.DTS bit is 1 SAR register initial value
DAR	Transfer destination address	Increment, decrement, or fixed*1	• When the MRB.DTS bit is 0 DAR register initial value • When the MRB.DTS bit is 1 Increment, decrement, or fixed*1
CRAH	Retains transfer counter	CRAH	CRAH
CRAL	Transfer counter A	CRAL - 1	CRAH
CRB	Transfer counter B	Not updated	Not updated

Note 1. Write-back is skipped in address-fixed mode.

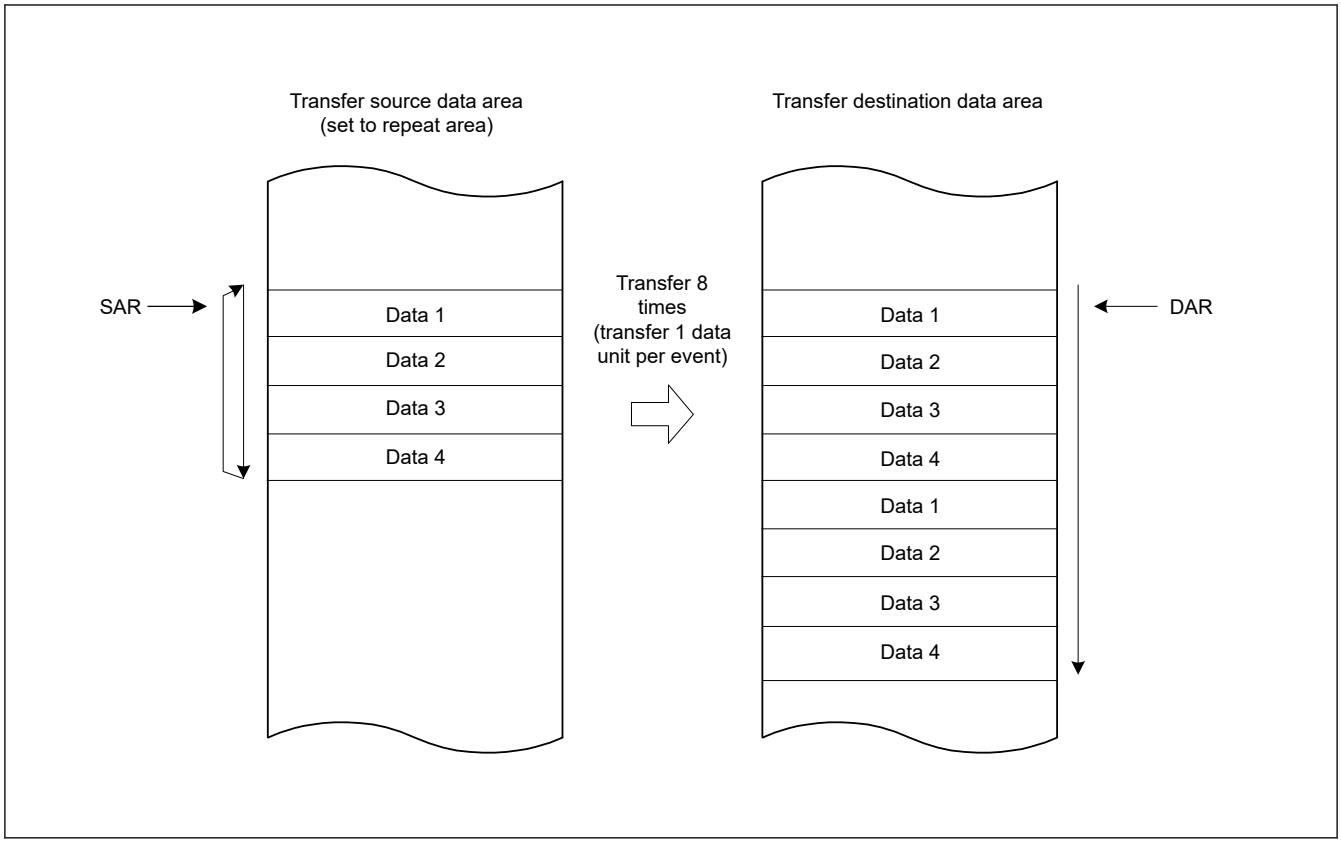


Figure 15.6 Memory map of repeat transfer mode when transfer source is a repeat area (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRAH = 0x04)

15.4.5 Block Transfer Mode

The block transfer mode allows single-block data transfer on a single activation source. Transfer source or transfer destination for the block area must be specified in the MRB.DTS bit. The block size can be set from 1 to 256 bytes, 1 to 256 halfwords (2 to 512 bytes), or 1 to 256 words (4 to 1024 bytes). When transfer of the specified block completes, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS = 1 or the DAR register when the DTS = 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.

The transfer count (block count) can be set from 1 to 65536. This mode enables an interrupt request to the CPU to be generated at the end of the specified-count block transfer.

Table 15.7 lists the register functions in block transfer mode, and Figure 15.7 shows the memory map for block transfer mode.

Table 15.7 Register functions in block transfer mode

Register	Description	Value written back by writing transfer information
SAR	Transfer source address	• When MRB.DTS bit is 0 Increment, decrement, or fixed*1 • When MRB.DTS bit is 1 SAR register initial value.
DAR	Transfer destination address	• When MRB.DTS bit is 0 DAR register initial value • When MRB.DTS bit is 1 Increment, decrement, or fixed*1.
CRAH	Holds block size	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB - 1

Note 1. Write-back is skipped in address-fixed mode.

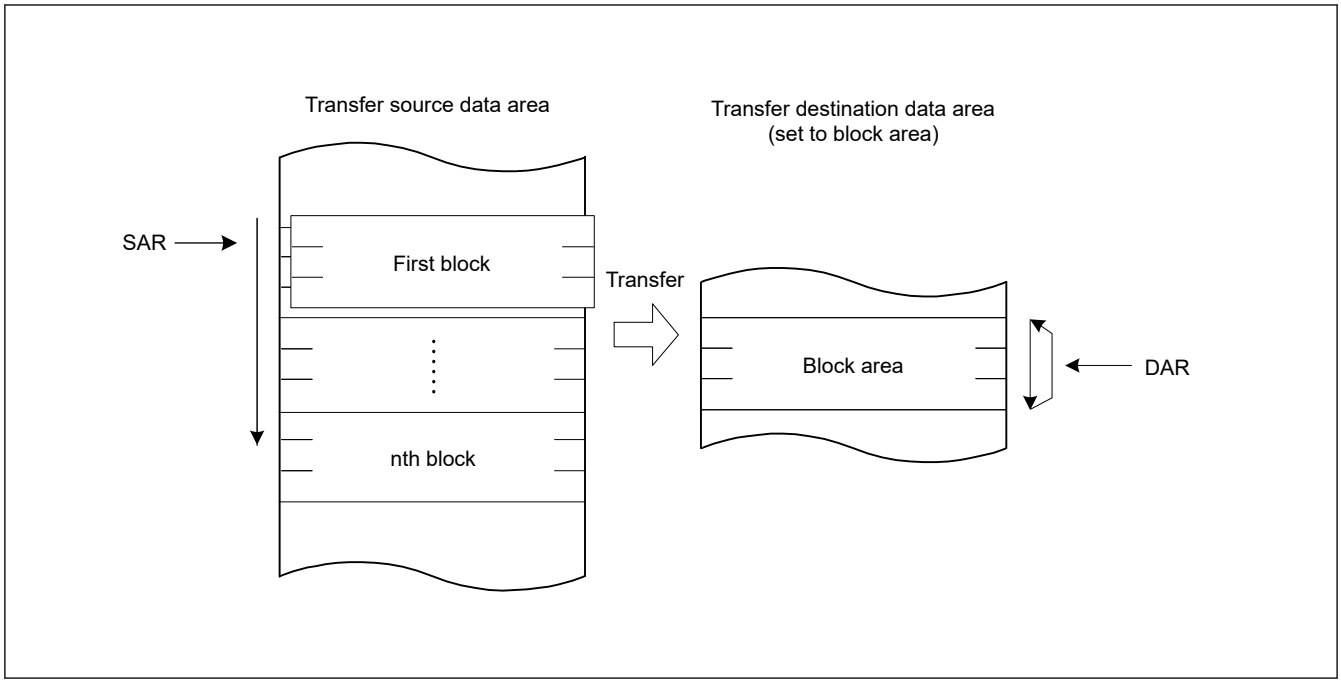


Figure 15.7 Memory map of block transfer mode

15.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single activation source. If the MRB.CHNE is set to 1 and CHNS to 0, an interrupt request to the CPU is not generated on completion of the specified number of rounds of transfer or by setting the MRB.DISEL bit to 1. An interrupt request is sent to the CPU each time DTC data transfer is performed. Data transfer has no effect on the ICU.IELSRn.IR flag of the activation source.

The SAR, DAR, CRA, CRB, MRA, and MRB registers can be set independently of each other to define the data transfer. Figure 15.8 shows a chain transfer operation.

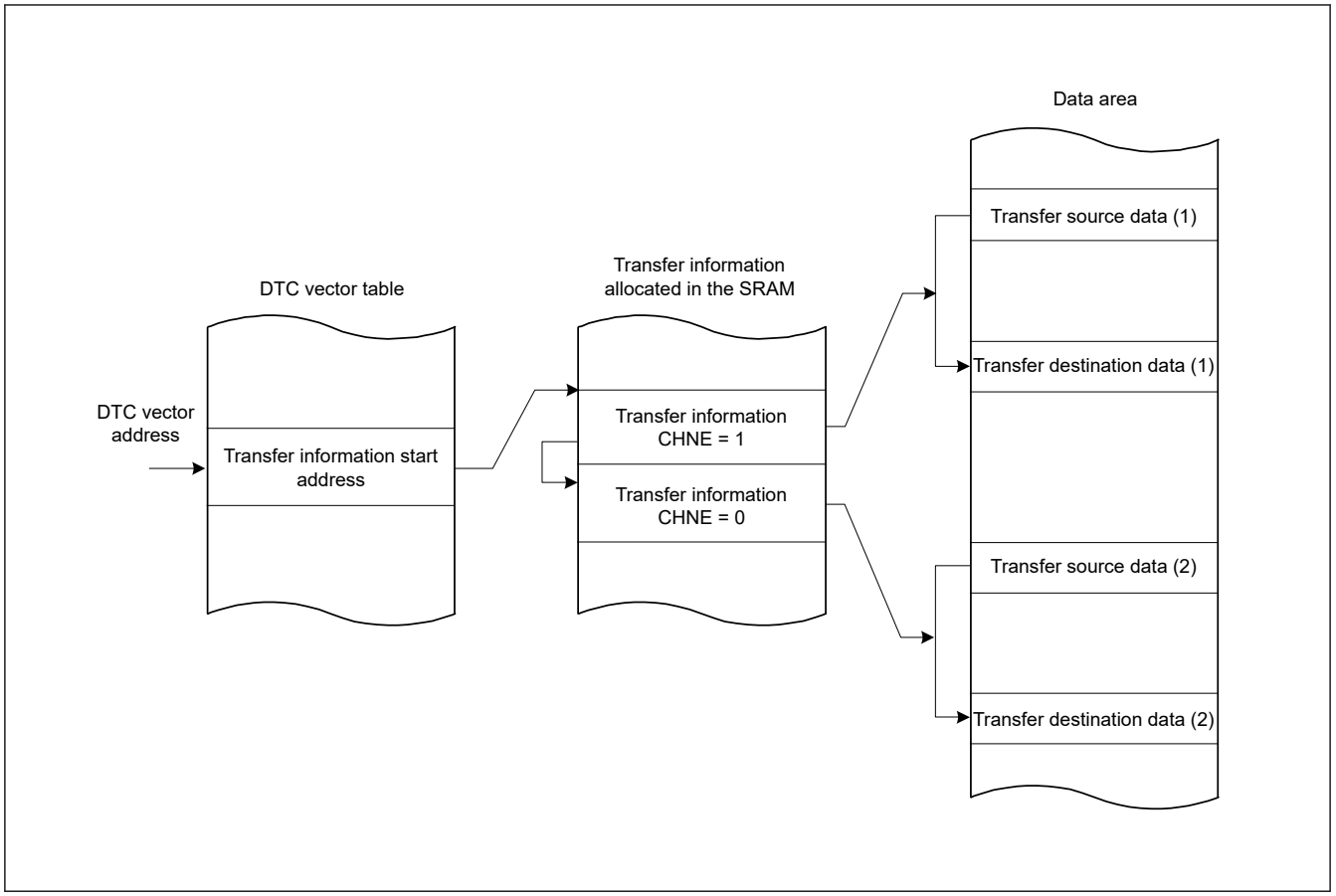


Figure 15.8 Chain transfer operation

Writing 1 to the MRB.CHNE and CHNS bits enables chain transfer to be performed only after completion of the specified data transfer. In repeat transfer mode, chain transfer is performed after completion of the specified data transfer. For details on chain transfer conditions, see Table 15.3.

15.4.7 Operation Timing

Figure 15.9 to Figure 15.12 are timing diagrams that show the minimum number of execution cycles.

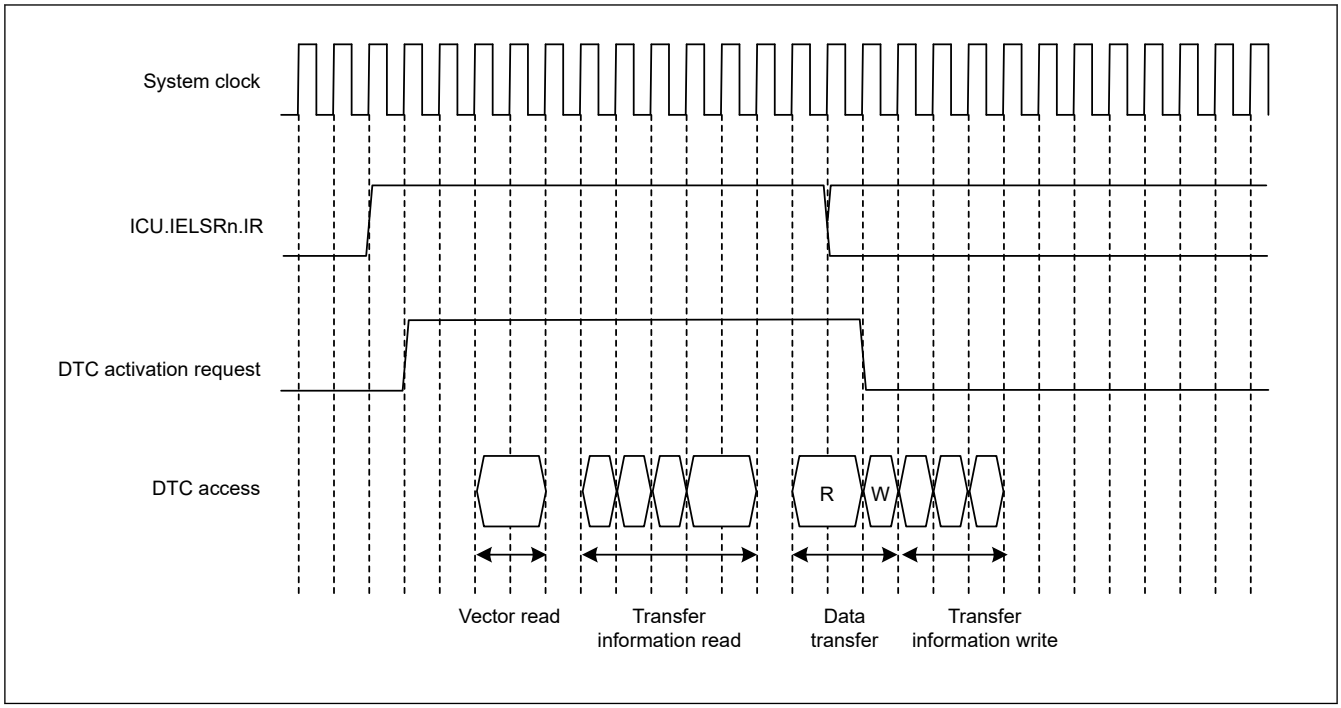


Figure 15.9 Example 1 of DTC operation timing in normal transfer and repeat transfer modes

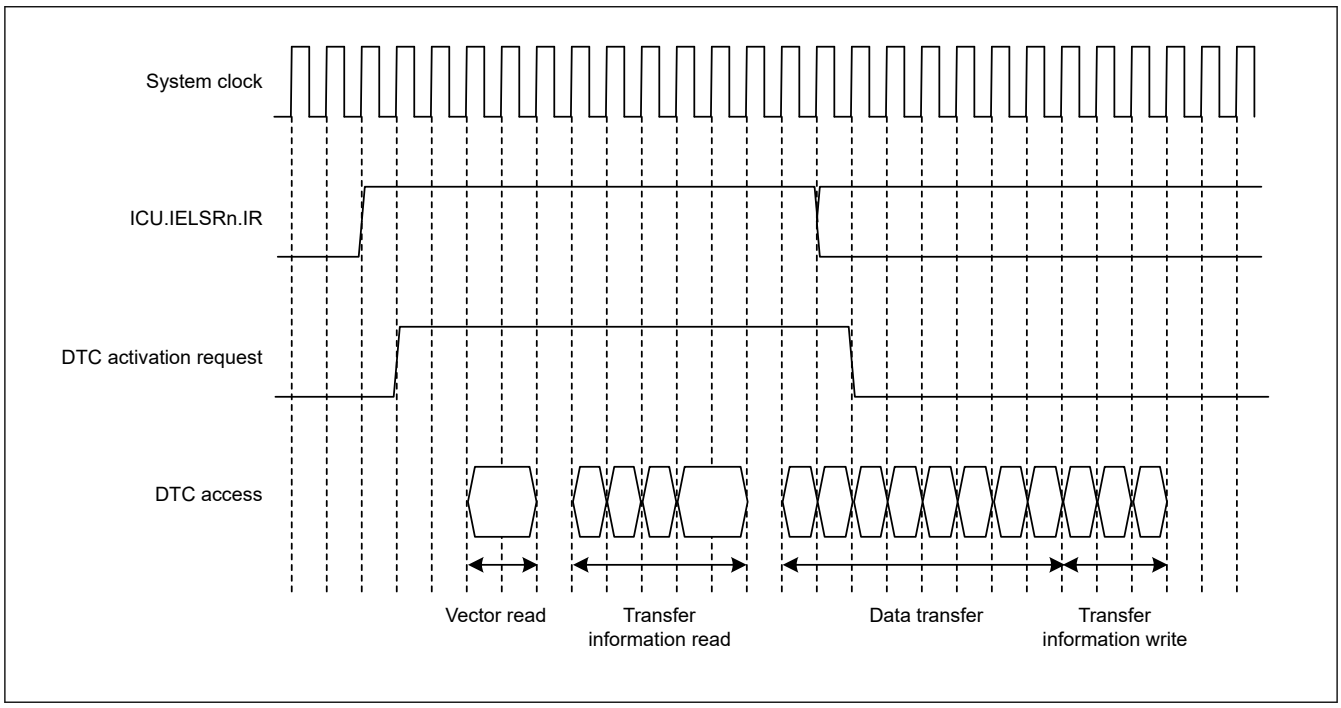


Figure 15.10 Example 2 of DTC operation timing in block transfer mode when the block size = 4

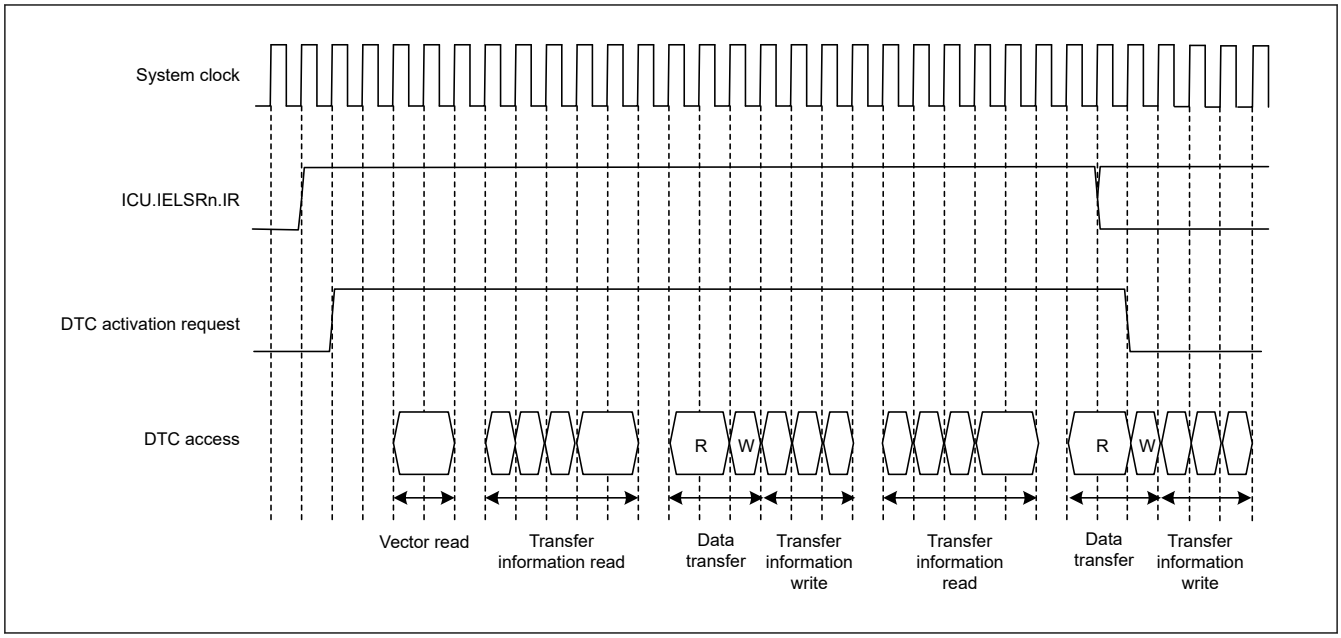
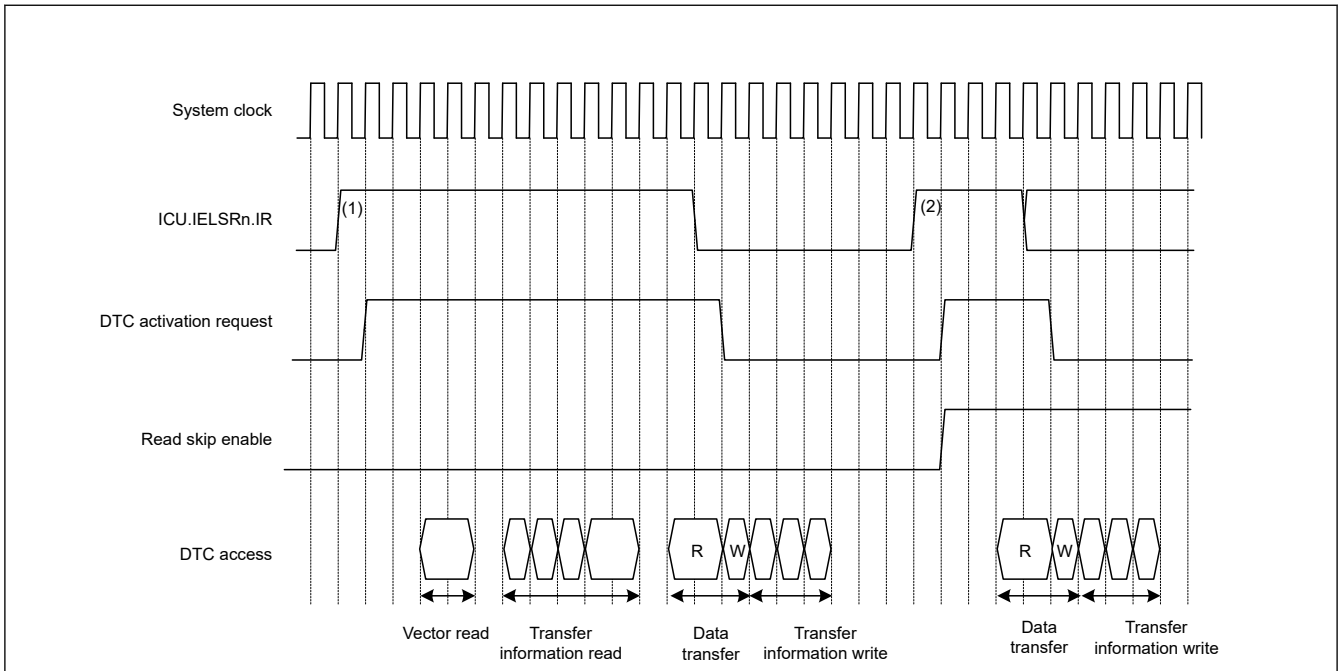


Figure 15.11 Example 3 of DTC operation timing for chain transfer



Note: When activation sources (vector numbers) of (1) and (2) are the same and the RRS = 1, the transfer information read for request (2) is skipped.

Figure 15.12 Example of operation when a transfer information read is skipped with the vector, transfer information, and transfer destination data on the SRAM, and the transfer source data on the peripheral module

15.4.8 Execution Cycles of DTC

Table 15.8 lists the execution cycles of single data transfer of the DTC. For the order of the execution states, see section 15.4.7. Operation Timing.

Table 15.8 Execution cycles of DTC

P: Block size (initial settings of CRAH and CRAL)
Cv: Cycles for access to vector transfer information storage destination
Ci: Cycles for access to transfer information storage destination address
Cr: Cycles for access to data read destination
Cw: Cycles for access to data write destination
The unit is for system clocks (ICLK) + 1 in the Vector read, Transfer information read, and Data transfer read columns and 2 in the Internal operation column.
Cv, Ci, Cr, and Cw vary depending on the corresponding access destination. For the number of cycles for respective access destinations, see section 32, SRAM, section 33, Flash Memory, and section 13, Buses.
The frequency ratio of the system clock and peripheral clock is also taken into consideration.
The DTC response time is the time from when the DTC activation source is detected until DTC transfer starts.
Table 15.8 does not include the time until DTC data transfer starts after the DTC activation source becomes active.

Transfer mode	Vector read		Transfer information read		Transfer information write			Data transfer		Internal operation	
								Read	Write		
Normal	Cv + 1	0*1	4 × Ci + 1	0*1	3 × Ci + 1*2	2 × Ci + 1*3	Ci*4	Cr + 1	Cw + 1	2	0*1
Repeat								Cr + 1	Cw + 1		
Block*5								P × Cr	P × Cw		

Note 1. When transfer information read is skipped.
Note 2. When neither SAR nor DAR is set to address-fixed mode.
Note 3. When SAR or DAR is set to address-fixed mode.
Note 4. When SAR and DAR are set to address-fixed mode.
Note 5. When the block size is 2 or more. If the block size is 1, the cycle number for normal transfer applies.

15.4.9 DTC Bus Mastership Release Timing

The DTC does not release the bus mastership during transfer information reads. Before the transfer information is read or written, the bus is arbitrated according to the priority determined by the bus master arbitrator. For bus arbitration, see section 13, Buses.

15.5 DTC Setting Procedure

Before using the DTC, set the DTC Vector Base Register (DTCVBR). Set the ICU.IELSRn.IELS[4:0] bits to 0 to disable the interrupt in the NVIC and follow the procedure in Table 15.9 to set the DTC.

Table 15.9 DTC setting procedure

No.	Step Name	Description
1	Set the DTCCR.RRS bit to 0	Set the DTCCR.RRS bit to 0 to reset the transfer information read skip flag. After that, the transfer information read is not skipped while the DTC is activated. Be sure to specify this setting when the transfer information is updated.
2	Set transfer information (MRA, MRB, SAR, DAR, CRA, and CRB)	Allocate transfer information (MRA, MRB, SAR, DAR, CRA, and CRB) in the data area. To set transfer information, see section 15.2. Register Descriptions. To allocate transfer information, see section 15.3.1. Allocating Transfer Information and DTC Vector Table.
3	Set transfer information start addresses in the DTC vector table	Set the transfer information start addresses in the DTC vector table. To set the DTC vector table, see section 15.3.1. Allocating Transfer Information and DTC Vector Table.
4	Set the DTCCR.RRS bit to 1	Set the DTCCR.RRS bit to 1 to enable skipping of the second and subsequent transfer information read cycles for continuous DTC activation from the same interrupt source. The RRS bit can be set to 1, but if this is set during DTC transfer, it becomes valid from the next transfer.
5	Set the ICU.IELSRn.DTCE bit to 1. Set the ICU.IELSRn.IELS[4:0] as interrupt source. The interrupt should be enabled in the NVIC.	Set the ICU.IELSRn.DTCE bit to 1. Set ICU.IELSRn.IELS[4:0] as interrupt sources that trigger DTC. The interrupt must be enabled in the NVIC. See section 12.3.2. Event Number in section 12, Interrupt Controller Unit (ICU).
6	Set the enable bit for an activation source interrupt	Set the enable bit for the activation source interrupts to 1. When a source interrupt is generated, the DTC is activated. To set the interrupt source enable bit, see the settings for the modules that are to be the activation sources.
7	Set the DTCST.DTCST bit to 1	Set the DTC Module Start bit (DTCST.DTCST) to 1.

Note: The DTCST.DTCST bit can be set even if the setting for each activation source is not completed.

15.6 Examples of DTC Usage

15.6.1 Normal Transfer

This section provides an example of DTC usage and its application when receiving 128 bytes of data from an SCI.

(1) Transfer information settings

In the MRA register, select a fixed source address (MRA.SM[1:0] = 00b), normal transfer mode (MRA.MD[1:0] = 00b), and byte-sized transfer (MRA.SZ[1:0] = 00b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] = 10b) and single data transfer by a single interrupt (MRB.CHNE = 0 and MRB.DISEL = 0). The MRB.DTS bit can be set to any value. Set the RDR register address of the SCI in the SAR register, the start address of the SRAM area for data storage in the DAR register, and 128 (0x0080) in the CRA register. The CRB register can be set to any value.

(2) DTC vector table settings

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

(3) ICU settings and DTC module activation

Set the ICU.IELSRn.DTCE bit to 1 and set ICU.IELSRn.IELS[4:0] as the SCI interrupt. The interrupt must be enabled in the NVIC. Set the DTCST.DTCST bit to 1.

(4) SCI settings

Enable the SCIn_RXI (n = 0 to 2, 9) interrupt by setting the SCR.RIE bit in the SCI to 1. If a reception error occurs during the SCI receive operation, reception stops. To manage this, use settings that allow the CPU to accept receive error interrupts.

(5) DTC transfer

Each time a reception of 1 byte by the SCI is complete, an SCIn_RXI interrupt is generated to activate the DTC. The DTC transfers the received byte from the RDR of the SCI to the SRAM, after which the DAR register is incremented and the CRA register is decremented.

(6) Interrupt handling

After 128 rounds of data transfer are complete and the value in the CRA register becomes 0, an SCIn_RXI interrupt request is generated for the CPU. Complete the process in the handling routine for this interrupt.

15.6.2 Chain transfer

This section provides an example of chain transfer by the DTC and describes its use in the output of pulses by the General PWM Timer (GPT). You can use chain transfer to transfer PWM timer compare data and change the period of the PWM timer for the GPT.

For the first of the chain transfers, normal transfer mode is specified for transfer to the GPTm.GTCCRC register (m = 160 to 163). For the second transfer, normal transfer mode is specified for transfer to the GPTm.GTCCRE register (m = 160 to 163). For the third transfer of the chained transfer, normal transfer mode for transfer to the GPTm.GTPBR register (m = 160 to 163) is specified. This is because clearing of the activation source and generation of an interrupt on completion of the specified number of transfers are restricted to the third of the chain transfers, that is, transfer while MRB.CHNE = 0.

The following example shows how to use the counter overflow interrupt with the GPT160.GTPR register as an activating source for the DTC.

(1) First transfer information setting

Set up transfer to the GPT160.GTCCRC register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up chain transfer (MRB.CHNE = 1 and MRB.CHNS = 0).

4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT160.GTCCRC register.
6. Set the CRAH and CRAL registers to the size of the data table. The CRB register can be set to any value.

(2) Second transfer information setting

Set up for transfer to the GPT160.GTCCRE register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up chain transfer (MRB.CHNE = 1, MRB.CHNS = 0).
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT160.GTCCRE register.
6. Set the CRAH and CRAL registers to the size of the data table. The CRB register can be set to any value.

(3) Third transfer information set

Set up transfer to the GPT160.GTPBR register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up single data transfer per interrupt (MRB.CHNE = 0, MRB.DISEL = 0). The MRB.DTS bit can be set to any value.
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT160.GTPBR register.
6. Set the CRA register to the size of the data table. The CRB register can be set to any value.

(4) Transfer information assignment

Place the transfer information for use in the transfer to the GPT160.GTPBR immediately after the transfer control information for use in the GPT160.GTCCRC and GPT160.GTCCRE registers.

(5) DTC vector table

In the DTC vector table, set the address where the transfer control information for use in transfer to the GPT160.GTCCRC and GPT160.GTCCRE registers starts.

(6) ICU setting and DTC module activation

1. Set the ICU.IELSRn.DTCE bit associated with the GPT160 counter overflow interrupt.
2. Set the ICU.IELSRn.IELS[4:0] bits and specify the GPT160 counter overflow.
3. Set the DTCST.DTCST bit to 1.

(7) GPT settings

1. Set the GPT160.GTIOR register so that the GTCCRA and GTCCRB registers operate as output compare registers.
2. Set the default PWM timer compare values in the GPT160.GTCCRA and GPT160.GTCCRB registers and the next PWM timer compare values in the GPT160.GTCCRC and GPT160.GTCCRE registers.
3. Set the default PWM timer period values in the GPT160.GTPR register and the next PWM timer period values in the GPT160.GTPBR register.
4. Set 1 to the output bit in PmnPFS.PDR, and set 00011b to the Peripheral Select bits in PmnPFS.PSEL[4:0].

(8) GPT activation

Set the GPT160.GTSTR.CSTRT bits to 1 to start the GPT160.GTCNT counter.

(9) DTC transfer

Each time a GPT160 counter overflow is generated with the GPT160.GTPR register, the next PWM timer compare values are transferred to the GPT160.GTCCRC and GPT160.GTCCRE registers. The setting for the next PWM timer period is transferred to the GPT160.GTPBR register.

(10) Interrupt handling

After the specified rounds of data transfer are complete, for example when the value in the CRA register for GPT transfer becomes 0, a GPT160 counter overflow interrupt request is issued for the CPU. Complete the process for this interrupt in the handling routine.

15.6.3 Chain Transfer when Counter = 0

The second data transfer is performed only when the transfer counter is set to 0 in the first data transfer, and the first data transfer information is repeatedly changed in the second transfer. Chain transfer enables transfers to be repeated 256 times or more.

The following procedure shows an example of configuring a 1-KB input buffer, where the input buffer is set so that its lower address starts with 0x00. Figure 15.13 shows a chain transfer when the counter = 0.

1. Set the normal transfer mode to input data for the first data transfer. Set the following:
   (a) Transfer source address = fixed.
   (b) CRA register = 0x0200 (512) times.
   (c) MRB.CHNE bit = 1 (chain transfer is enabled).
   (d) MRB.CHNS bit = 1 (chain transfer is performed only when the transfer counter is 0).
   (e) MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
2. Prepare the upper 8-bit address of the start address at every 512 times of the transfer destination address for the first data transfer in different area such as the flash. For example, when setting the input buffer to 0x8000 to 0x83FF, prepare 0x82 and 0x80.
3. For the second data transfer:
   (a) Set the repeat transfer mode (with transfer source and destination address = fixed.) to reset the transfer counter of the first data transfer.
   (b) Specify the CRA register in the first transfer information area for the transfer destination.
   (c) Set the MRB.CHNE bit = 1 (chain transfer is enabled).
   (d) Set the MRB.CHNS bit = 0 (select continuous chain transfer).
   (e) Set the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
   (f) CRA register = 0x0101 (The transfer count is 1).
4. For the third data transfer:
   (a) Set the repeat transfer mode (with the source as the repeat area) to reset the transfer destination address of the first data transfer.
   (b) Specify the upper 8 bits of the DAR register in the first transfer information area for the transfer destination.
   (c) Set the MRB.CHNE bit = 0 (chain transfer is disabled).
   (d) Set the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
   (e) When setting the input buffer to 0x8000 to 0x83FF, also set the transfer counter to 2.
5. The first data transfer is performed by an interrupt 512 times. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the transfer counter of the first data transfer to 0x0200. The lower 8 bits of the transfer destination address and the transfer counter of the first data transfer becomes 0x0200.
6. The second data transfer is performed by an interrupt 1 times. When the transfer counter of the first data transfer becomes 0, the third data transfer starts. Set the upper 8 bits of the transfer destination address of the first data transfer to

0x82. The lower 8 bits of the transfer destination address becomes 0x00 and the transfer counter of the first data transfer becomes 0x0200.

7. In succession, the first data transfer is performed by an interrupt 512 times as specified for the first data transfer. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the transfer counter of the first data transfer to 0x0200. The lower 8 bits of the transfer destination address and the transfer counter of the first data transfer becomes 0x0200.
8. The second data transfer is performed by an interrupt 1 times. When the transfer counter of the first data transfer becomes 0, the third data transfer starts. Set the upper 8 bits of the transfer destination address of the first data transfer to 0x80. The lower 8 bits of the transfer destination address becomes 0x00 and the transfer counter of the first data transfer becomes 0x0200.
9. Steps 5 to 8 are repeated indefinitely. Because the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

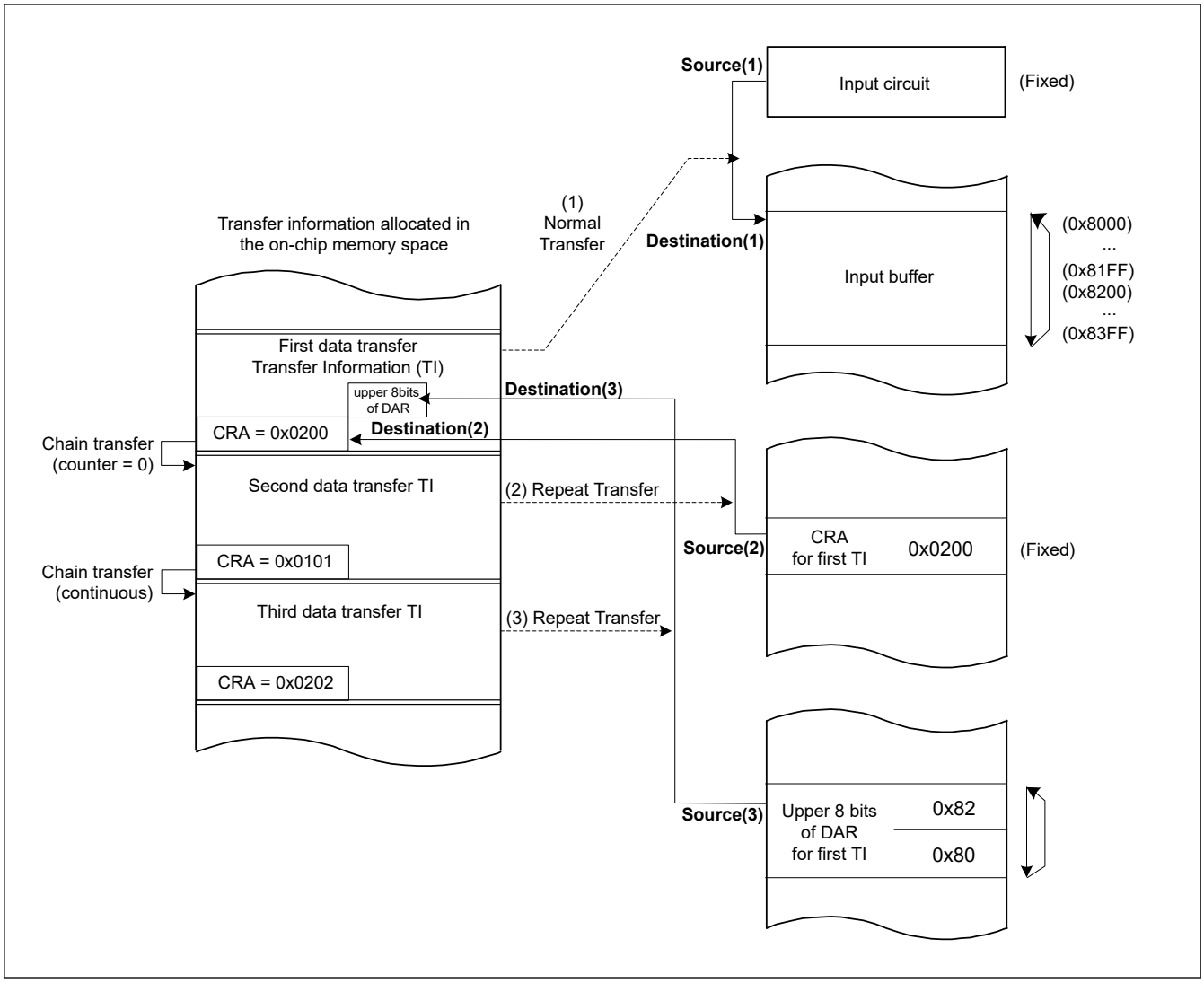


Figure 15.13 Chain transfer when counter = 0

15.7 Interrupt

15.7.1 Interrupt Sources

When the DTC completes data transfer of the specified count or when data transfer with MRB.DISEL set to 1 is complete, a DTC activation source generates an interrupt to the CPU. Two types of interrupt are available: interrupts triggered by a DTC activation (per channel) and an interrupt triggered by the event signal DTC_COMPLETE (common to all channels). Interrupts to the CPU are controlled according to the settings in the NVIC and the ICU.IELSRn.IELS[4:0] bits. See section

12, Interrupt Controller Unit (ICU). The DTC prioritizes activation sources by granting the smaller interrupt vector numbers higher priority. The priority of interrupts to the CPU is determined by the NVIC priority.

15.8 Event Link

The DTC can produce an event link request on completion of one transfer request.

15.9 Low Power Consumption Function

Before transitioning to the module-stop state, or Software Standby mode without Snooze mode transition, set the DTCST.DTCST bit to 0, and then perform the operations described in the following sections. The DTC is available in Snooze mode by setting the SYSTEM.SNZCR.SNZDTCEN bit to 1. See section 10, Low Power Modes.

(1) Module-Stop Function

Writing 1 to the MSTPCRA.MSTPA22 bit enables the module-stop function of the DTC. If a DTC transfer is in progress when 1 is written to the MSTPCRA.MSTPA22 bit, the transition to the module-stop state proceeds after the DTC transfer ends. While the MSTPCRA.MSTPA22 bit is 1, accessing the DTC registers is prohibited. Writing 0 to the MSTPCRA.MSTPA22 bit releases the DTC from the module-stop state.

(2) Software Standby Mode

Use the settings described in section 10.7.1. Transition to Software Standby Mode.

If DTC transfer operations are in progress when the WFI instruction is executed, the transition to Software Standby mode is executed after the completion of the DTC transfer.

(3) Snooze Mode

When the snooze control circuit receives a snooze request in Software Standby mode, the MCU transitions to Snooze mode. See section 10.8.1. Transition to Snooze Mode. DTC operation in Snooze mode can be selected in the SYSTEM.SNZCR.SNZDTCEN bit. If DTC operation is enabled in Snooze mode, before transitioning to Software Standby mode, set the DTCST.DTCST bit to 1. To return to Software Standby mode through DTC, set SYSTEM.SNZEDCR0.DTCZRED or SYSTEM.SNZEDCR0.DTCNZRED to 1. See section 10.8.3. Returning from Snooze Mode to Software Standby Mode. SYSTEM.SNZEDCR0.DTCZRED enables or disables a snooze end request on completion of the last DTC transmission, detected on DTC transmission completion when CRA and CRB are 0. SYSTEM.SNZEDCR0.DTCNZRED enables or disables a snooze end request on a not last DTC transmission completion (CRA and CRB are not 0), detected on DTC transmission completion when CRA and CRB are not 0. The DTC activation request from the ICU is stopped during Software Standby mode but not stopped during Snooze mode.

(4) Notes on Low Power Consumption Function

For the WFI instruction and the register setting procedure, see section 10, Low Power Modes.

To perform a DTC transfer after returning from a low power mode without a Snooze mode transition, set the DTCST.DTCST bit to 1 again.

To use a request that is generated in Software Standby mode as an interrupt request to the CPU but not as a DTC activation request, specify the CPU as the interrupt request destination as described in section 12.4.1. Detecting Interrupts, then execute the WFI instruction. If DTC operation is enabled in Snooze mode, do not use the module-stop function of the DTC.

15.10 Usage Notes

15.10.1 Transfer Information Start Address

You must set multiples of 4 for the transfer information start addresses in the vector table. Otherwise, such addresses are accessed with their lowest 2 bits regarded as 00b.

16. Event Link Controller (ELC)

16.1 Overview

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

Table 16.1 lists the ELC specifications, and Figure 16.1 shows a block diagram.

Table 16.1 ELC Specifications

Item	Description
Event link function	105 types of event signals can be directly connected to modules. The ELC generates the ELC event signal, and events that activate the DTC.
Module-stop function	Module-stop state can be set.

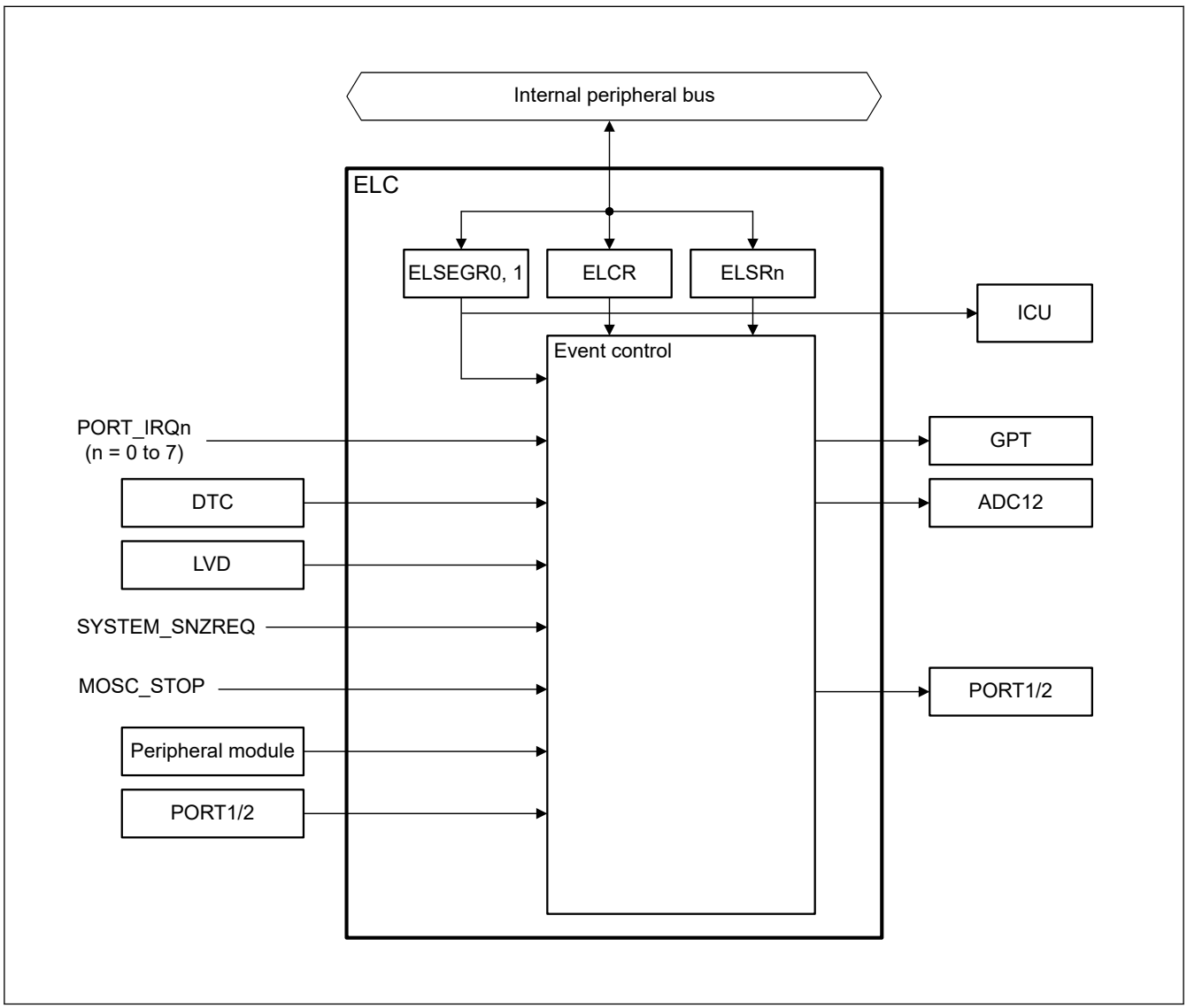


Figure 16.1 ELC block diagram

16.2 Register Descriptions

16.2.1 ELCR : Event Link Controller Register

Base address: ELC = 0x4004_1000

Offset address: 0x00

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

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	ELCON	All Event Link Enable 0: ELC function is disabled. 1: ELC function is enabled.	R/W

The ELCR register controls the ELC operation.

16.2.2 ELSEGRn : Event Link Software Event Generation Register n (n = 0, 1)

Base address: ELC = 0x4004_1000

Offset address: 0x02 + 0x02 × n

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

Bit	Symbol	Function	R/W
0	SEG	Software Event Generation 0: Normal operation 1: Software event is generated.	W
5:1	—	These bits are read as 0. The write value should be 0.	R/W
6	WE	SEG Bit Write Enable 0: Write to SEG bit disabled. 1: Write to SEG bit enabled.	R/W
7	WI	ELSEGR Register Write Disable 0: Write to ELSEGR register enabled. 1: Write to ELSEGR register disabled.	W

SEG bit (Software Event Generation)

When 1 is written to the SEG bit while the WE bit is 1, a software event is generated. This bit is read as 0. Even when 1 is written to this bit, data is not stored. The WE bit must be set to 1 before writing to this bit.

A software event can trigger a linked DTC event.

WE bit (SEG Bit Write Enable)

The SEG bit can only be written to when the WE bit is 1. Clear the WI bit to 0 before writing to this bit.

[Setting condition]

- If 1 is written to this bit while the WI bit is 0, this bit becomes 1.

[Clearing condition]

- If 0 is written to this bit while the WI bit is 0, this bit becomes 0.

WI bit (ELSEGR Register Write Disable)

The ELSEGR register can only be written to when the write value to the WI bit is 0. This bit is read as 1. Before setting the WE or SEG bit, the WI bit must be set to 0.

16.2.3 ELSRn : Event Link Setting Register n (n = 0 to 9, 14, 15)

Base address: ELC = 0x4004_1000

Offset address: 0x10 + 0x04 × n

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

Bit	Symbol	Function	R/W
7:0	ELS[7:0]	Event Link Select 0x00: Event output disabled for the associated peripheral module 0x01: Number setting for the event signal to be linked ⋮ 0xA2: Number setting for the event signal to be linked Others: Settings prohibited	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

The ELSRn register specifies an event signal to be linked to each peripheral module. Table 16.2 shows the association between the ELSRn register and the peripheral modules. Table 16.3 shows the association between the event signal names set in the ELSRn register and the signal numbers.

Table 16.2 Association between the ELSRn registers and peripheral functions

Register name	Peripheral function (module)	Event name
ELSR0	GPT (A)	ELC_GPTA
ELSR1	GPT (B)	ELC_GPTB
ELSR2	GPT (C)	ELC_GPTC
ELSR3	GPT (D)	ELC_GPTD
ELSR4	GPT (E)	ELC_GPTE
ELSR5	GPT (F)	ELC_GPTF
ELSR6	GPT (G)	ELC_GPTG
ELSR7	GPT (H)	ELC_GPTH
ELSR8	ADC12A	ELC_AD00
ELSR9	ADC12B	ELC_AD01
ELSR14	PORT 1	ELC_PORT1
ELSR15	PORT 2	ELC_PORT2

Table 16.3 Association between event signal names set in ELSRn.ELS[7:0] bits and signal numbers (1 of 3)

Event number	Interrupt request source	Name	Description
0x01	Port	PORT_IRQ0*1	External pin interrupt 0
0x02		PORT_IRQ1*1	External pin interrupt 1
0x03		PORT_IRQ2*1	External pin interrupt 2
0x04		PORT_IRQ3*1	External pin interrupt 3
0x05		PORT_IRQ4*1	External pin interrupt 4
0x06		PORT_IRQ5*1	External pin interrupt 5
0x07		PORT_IRQ6*1	External pin interrupt 6
0x08		PORT_IRQ7*1	External pin interrupt 7
0x0A	DTC	DTC_DTCEND*3	DTC transfer end
0x0D	LVD	LVD_LVD1	Voltage monitor 1 interrupt
0x0E		LVD_LVD2	Voltage monitor 2 interrupt
0x0F	MOSC	MOSC_STOP	Main clock oscillation stop
0x10	Low Power Mode	SYSTEM_SNZREQ*2 *3	Snooze entry
0x11	AGT0	AGT0_AGTI	AGT interrupt
0x12		AGT0_AGTCMAI	Compare match A
0x13		AGT0_AGTCMBI	Compare match B
0x14	AGT1	AGT1_AGTI	AGT interrupt
0x15		AGT1_AGTCMAI	Compare match A
0x16		AGT1_AGTCMBI	Compare match B
0x17	IWDT	IWDT_NMIUNDF	IWDT underflow
0x18	WDT	WDT_NMIUNDF	WDT underflow
0x1C	ADC12	ADC120_ADI	A/D scan end interrupt
0x20		ADC120_WCMPM*3	Compare match
0x21		ADC120_WCMPUM*3	Compare mismatch
0x22	ACMPHS	ACMP_HS0*1	High-speed analog comparator interrupt 0
0x23		ACMP_HS1*1	High-speed analog comparator interrupt 1
0x27	IIC0	IIC0_RXI	Receive data full
0x28		IIC0_TXI	Transmit data empty
0x29		IIC0_TEI	Transmit end
0x2A		IIC0_EEI	Transfer error
0x34	DOC	DOC_DOPCI*3	Data operation circuit interrupt
0x3D	I/O Ports	IOPORT_GROUP1	Port 1 event
0x3E		IOPORT_GROUP2	Port 2 event
0x3F	ELC	ELC_SWEVT0	Software event 0
0x40		ELC_SWEVT1	Software event 1

Table 16.3 Association between event signal names set in ELSRn.ELS[7:0] bits and signal numbers (2 of 3)

Event number	Interrupt request source	Name	Description
0x46	GPT160	GPT0_CCMPA	Compare match A
0x47		GPT0_CCMPB	Compare match B
0x48		GPT0_CMPC	Compare match C
0x49		GPT0_CMPD	Compare match D
0x4A		GPT0_CMPE	Compare match E
0x4B		GPT0_CMPF	Compare match F
0x4C		GPT0_OVF	Overflow
0x4D		GPT0_UDF	Underflow
0x4E		GPT0_PC	Cycle count function end
0x4F		GPT0_ADTRGA	A/D converter start request A
0x50		GPT0_ADTRGB	A/D converter start request B
0x51	GPT161	GPT1_CCMPA	Compare match A
0x52		GPT1_CCMPB	Compare match B
0x53		GPT1_CMPC	Compare match C
0x54		GPT1_CMPD	Compare match D
0x55		GPT1_CMPE	Compare match E
0x56		GPT1_CMPF	Compare match F
0x57		GPT1_OVF	Overflow
0x58		GPT1_UDF	Underflow
0x59		GPT1_PC	Cycle count function end
0x5A		GPT1_ADTRGA	A/D converter start request A
0x5B		GPT1_ADTRGB	A/D converter start request B
0x5C	GPT162	GPT2_CCMPA	Compare match A
0x5D		GPT2_CCMPB	Compare match B
0x5E		GPT2_CMPC	Compare match C
0x5F		GPT2_CMPD	Compare match D
0x60		GPT2_CMPE	Compare match E
0x61		GPT2_CMPF	Compare match F
0x62		GPT2_OVF	Overflow
0x63		GPT2_UDF	Underflow
0x64		GPT2_PC	Cycle count function end
0x65		GPT2_ADTRGA	A/D converter start request A
0x66		GPT2_ADTRGB	A/D converter start request B
0x70	GPT	GPT_UVWEDGE	UVW edge event
0x71	SCI0	SCI0_RXI*4	Receive data full
0x72		SCI0_TXI*4	Transmit data empty
0x73		SCI0_TEI	Transmit end
0x74		SCI0_ERI*4	Receive error
0x75		SCI0_AM	Address match event

Table 16.3 Association between event signal names set in ELSRn.ELS[7:0] bits and signal numbers (3 of 3)

Event number	Interrupt request source	Name	Description
0x77	SCI1	SCI1_RXI	Receive data full
0x78		SCI1_TXI	Transmit data empty
0x79		SCI1_TEI	Transmit end
0x7A		SCI1_ERI	Receive error
0x7B		SCI1_AM	Address match event
0x7C	SCI9	SCI9_RXI	Receive data full
0x7D		SCI9_TXI	Transmit data empty
0x7E		SCI9_TEI	Transmit end
0x7F		SCI9_ERI	Receive error
0x80		SCI9_AM	Address match event
0x81	SPI0	SPI0_SPRI	Receive buffer full
0x82		SPI0_SPTI	Transmit buffer empty
0x83		SPI0_SPII	Idle
0x84		SPI0_SPEI	Error
0x85		SPI0_SPTEND	Transmission completed event
0x8E	SCI2	SCI2_RXI	Receive data full
0x8F		SCI2_TXI	Transmit data empty
0x90		SCI2_TEI	Transmit end
0x91		SCI2_ERI	Receive error
0x92		SCI2_AM	Address match event
0x98	GPT163	GPT3_CCMPA	Compare match A
0x99		GPT3_CCMPB	Compare match B
0x9A		GPT3_CMPC	Compare match C
0x9B		GPT3_CMPD	Compare match D
0x9C		GPT3_CMPE	Compare match E
0x9D		GPT3_CMPF	Compare match F
0x9E		GPT3_OVF	Overflow
0x9F		GPT3_UDF	Underflow
0xA0		GPT3_PC	Cycle count function end
0xA1		GPT3_ADTRGA	A/D converter start request A
0xA2		GPT3_ADTRGB	A/D converter start request B

Note 1. Only pulse (edge detection) is supported.
Note 2. ELSR8, 9, 14, 15 can select this event.
Note 3. This event can occur in Snooze mode.
Note 4. This event is not supported in FIFO mode.

16.3 Operation

16.3.1 Relation between Interrupt Handling and Event Linking

Event number for an event link is the same as that for the associated interrupt source. For information on generating event signals, see the explanation in the chapter for each event source module.

16.3.2 Linking Events

When an event occurs and that event is already set as a trigger in the Event Link Setting Register (ELSRn), the associated module is activated. The operation of the module must be set up in advance. Table 16.4 lists the operations of modules when an event occurs.

Table 16.4 Module operations when event occurs

Module	Operations When Event is Input
GPT	• Start counting • Stop counting • Clear counting • Up counting • Down counting • Input capture
ADC12	Start A/D conversion
I/O Ports	• Change pin output based on the EORR (reset) or EOSR (set) • Latch pin state to EIDR • The following ports can be used for the ELC: Port 1 Port 2

16.3.3 Example of Procedure for Linking Events

To link events:

1. Set the operation of the module for which an event is to be linked.
2. Set the appropriate ELSRn.ELS[7:0] bits for the module to be linked.
3. Set the ELCR.ELCON bit to 1 to enable linkage of all events.
4. Configure the module from which an event is output and activate the module. The link between the two modules is now active.
5. To stop event linkage of modules individually, set 0 to the ELSRn.ELS[7:0] bit associated with the modules. To stop linkage of all the events, set the ELCR.ELCON bit to 0.

16.4 Usage Notes

16.4.1 Linking DTC Transfer End Signals as Events

When linking the DTC transfer end signals as events, do not set the same peripheral module as the DTC transfer destination and event link destination. If set, the peripheral module might be started before DTC transfer to the peripheral module is complete.

16.4.2 Setting Clocks

To link events, you must enable the ELC and the related modules. The modules cannot operate if the related modules are in the module-stop state or in low power mode in which the module is stopped (Software Standby mode).

Some modules can perform in Snooze mode. For more information, see Table 16.3 and section 10, Low Power Modes.

16.4.3 Module-Stop Function Setting

The Module Stop Control Register C (MSTPCRC) can enable or disable ELC operation. The ELC is initially stopped after reset. Releasing the module-stop state enables access to the registers. The ELCON bit must be set to 0 before disabling ELC operation using the Module Stop Control Register. For more information, see Table 16.3 and section 10, Low Power Modes.

16.4.4 ELC Delay Time

In Figure 16.2, module A accesses module B through the ELC. There is a delay time in the ELC between module A and module B. Table 16.5 shows the ELC delay time.

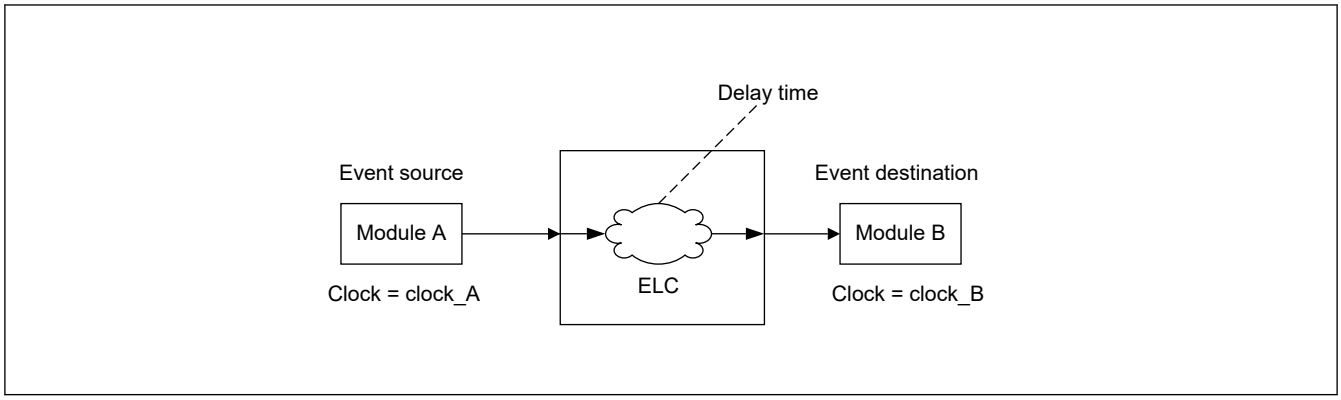


Figure 16.2 ELC delay time

Table 16.5 ELC delay time

Clock domain	Clock frequency	ELC delay time
clock_A = clock_B	clock_A = clock_B	0 cycle
clock_A ≠ clock_B	clock_A = clock_B	1 cycle to 2 cycles
	clock_A > clock_B	1 cycle to 2 cycles of B
	clock_A < clock_B	1 cycle to 2 cycles of A

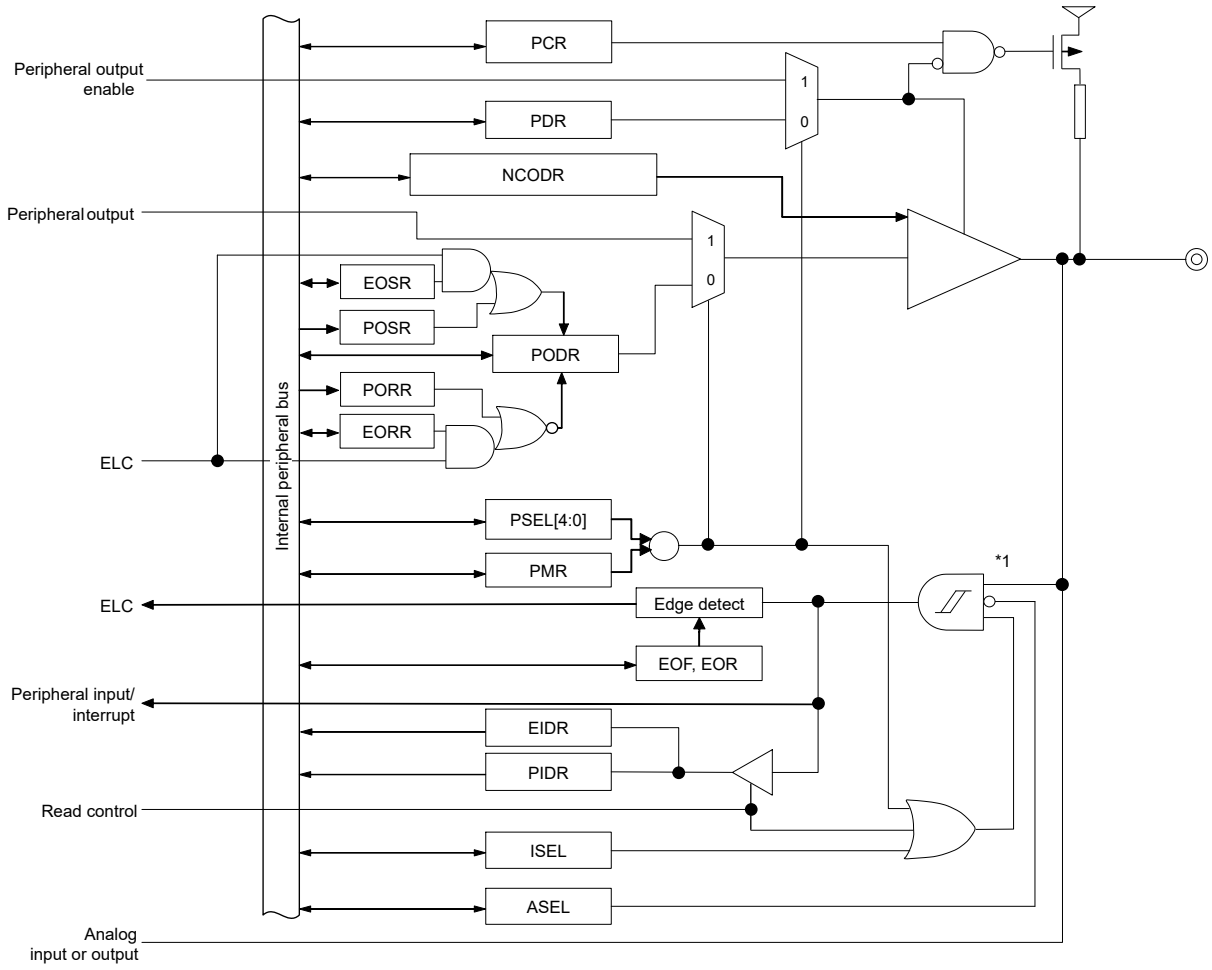
17. I/O Ports

17.1 Overview

The I/O port pins operate as general I/O port pins, I/O pins for peripheral modules, interrupt input pins, analog I/O, port group function for the ELC.

All pins operate as input pins immediately after a reset, and pin functions are switched by register settings. The I/O ports and peripheral modules for each pin are specified in the associated registers.

Figure 17.1 shows a connection diagram for the I/O port registers. The configuration of the I/O ports differs depending on the package. Table 17.1 lists the I/O port specifications by package, and Table 17.2 lists the port functions.



Note: This figure shows a basic port configuration. The configuration differs depending on the ports.
Note 1. Schmitt Trigger is enabled when PMR = 1 or ISEL = 1. For peripheral selection, see section 17.6. Peripheral Select Settings for Each Product.

Figure 17.1 Connection diagram for I/O port registers

Table 17.1 shows the I/O port specifications and Table 17.2 shows the port functions.

Table 17.1 I/O port specifications

Port	Package		Package		Package	
	48 pins	Number of pins	32 pins	Number of pins	24 pins	Number of pins
Port 0	P000 to P002, P010 to P015	9	P010 to P015	6	P010 to P011, P013 to P015	5
Port 1	P100 to P104, P108 to P112	10	P100 to P103, P108 to P110, P112	8	P100 to P103, P108 to P110	7
Port 2	P200 to P201, P206 to P208, P212 to P213	7	P200 to P201, P207 to P208, P212 to P213	6	P200 to P201, P212 to P213	4
Port 3	P300 to P302	3	P300	1	P300	1
Port 4	P400 to P403, P407 to P409	7	P402 to P403, P407	3	P407	1
Port 5	P500	1	—	0	—	0
Port 9	P913 to P915	3	P913 to P914	2	—	0

Table 17.2 I/O port functions

Port	Port name	Input pull-up	Input mode switching	Open drain output	5V tolerant	I/O
Port 0	P000 to P002, P010 to P015	✓	—	—	—	Input/ Output
Port 1	P100, P101	✓	CMOS/TTL	✓	—	Input/ Output
	P102 to P104, P108 to P112	✓	—	✓	—	Input/ Output
Port 2	P200	—	—	—	—	Input
	P201, P206, P207, P208, P212, P213	✓	—	✓	—	Input/ Output
Port 3	P300 to P302	✓	—	✓	—	Input/ Output
Port 4	P400, P401	✓	CMOS/TTL	✓	✓	Input/ Output
	P407, P408	✓	CMOS/TTL	✓	—	Input/ Output
	P402, P403, P409	✓	—	✓	—	Input/ Output
Port 5	P500	✓	—	✓	—	Input/ Output
Port 9	P913 to P915	✓	—	✓	—	Input/ Output

Note: ✓: Available
—: Setting prohibited

17.2 Register Descriptions

17.2.1 PCNTR1/PODR/PDR : Port Control Register 1

Base address: PORTm = 0x4004_0000 + 0x0020 × m (m = 0 to 5, 9)

Offset address: 0x000 (PCNTR1/PODR)
0x002 (PDR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PODR15	PODR14	PODR13	PODR12	PODR11	PODR10	PODR09	PODR08	PODR07	PODR06	PODR05	PODR04	PODR03	PODR02	PODR01	PODR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PDR15	PDR14	PDR13	PDR12	PDR11	PDR10	PDR09	PDR08	PDR07	PDR06	PDR05	PDR04	PDR03	PDR02	PDR01	PDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PDR15 to PDR00	Pmn Direction 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W
31:16	PODR15 to PODR00	Pmn Output Data 0: Low output 1: High output	R/W

Note: m = 0 to 5, 9, n = 00 to 15

The Port Control Register 1 (PCNTR1/PODR/PDR) is a 32-bit or 16-bit read/write register that controls port direction and port output data. The PCNTR1 specifies the port direction and output data, and is accessed in 32-bit units. The PDRn (bits [15:0] in PCNTR1) and PODRn (bits [31:16] in PCNTR1) respectively, are accessed in 16-bit units.

PDRn bits (Pmn Direction)

The PDRn bits select the input or output direction for individual pins on the associated port when the pins are configured as general I/O pins. Each pin on port m is associated with a PORTm.PCNTR1.PDRn bit. The I/O direction can be specified in 1-bit units. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0. P200 is input only, so PORT2.PCNTR1.PDR00 bit is reserved. The PDRn bit in the PORTm.PCNTR1 register serves the same function as the PDR bit in the PFS.PmnPFS register.

PODRn bits (Pmn Output Data)

The PODRn bits hold data to be output from the general I/O pins. Bits of non-existent port m are reserved. Reserved bits are read as 0. The write value should be 0. P200 is input only, so PORT2.PCNTR1.PODR00 bit is reserved. The PODRn bit in the PORTm.PCNTR1 register serves the same function as the PODR bit in the PFS.PmnPFS register.

17.2.2 PCNTR2/EIDR/PIDR : Port Control Register 2

Base address: PORTm = 0x4004_0000 + 0x0020 × m (m = 0 to 5, 9)

Offset address: 0x004 (PCNTR2/EIDR)
0x006 (PIDR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EIDR15	EIDR14	EIDR13	EIDR12	EIDR11	EIDR10	EIDR09	EIDR08	EIDR07	EIDR06	EIDR05	EIDR04	EIDR03	EIDR02	EIDR01	EIDR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PIDR15	PIDR14	PIDR13	PIDR12	PIDR11	PIDR10	PIDR09	PIDR08	PIDR07	PIDR06	PIDR05	PIDR04	PIDR03	PIDR02	PIDR01	PIDR00
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
15:0	PIDR15 to PIDR00	Pmn State 0: Low level 1: High level	R
31:16	EIDR15 to EIDR00 *2	Port Event Input Data*1 When an ELC_PORTx signal occurs 0: Low input 1: High input	R

Note: m = 0 to 5, 9, n = 00 to 15
Note 1. x = 1, 2 for EIDR only
Note 2. Supported for ports 1 to 2.

The Port Control Register 2 (PCNTR2//EIDR/PIDR) allows read access to the Pmn state and the port event input data using 32-bit or 16-bit access.

The PCNTR2 represents the Pmn state and the port event input data, and is accessed in 32-bit units.

The PIDRn (bits [15:0] in PCNTR2) and EIDRn (bits [31:16] in PCNTR2) respectively, are accessed in 16-bit units. Bits associated with non-existent pins are reserved. Reserved bits are read as undefined.

PIDRn bits (Pmn State)

The PIDRn bits reflect the individual pin states of the port, regardless of the values set in PmnPFS.PMR and PORTm.PCNTR1.PDRn. The PIDRn bit in the PORTm.PCNTR2 register serves the same function as the PIDR bit in the PFS.PmnPFS register.

A pin state cannot be reflected in PIDRn when one of the following functions is enabled:

- Main clock oscillator (MOSC)
- Analog function (ASEL = 1)

EIDRn bits (Port Event Input Data)

The EIDRn bits latch a pin state when an ELC_PORTx signal occurs. Pin states can only be input to EIDRn when PmnPFS.PMR and PORTm.PCNTR1.PDRn are 0. When the PmnPFS.ASEL bit is set to 1, the associated pin state is not reflected in EIDRn.

17.2.3 PCNTR3/PORR/POSR : Port Control Register 3

Base address: PORTm = 0x4004_0000 + 0x0020 × m (m = 0 to 5, 9)

Offset address: 0x008 (PCNTR3/PORR)
0x00A (POSR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PORR 15	PORR 14	PORR 13	PORR 12	PORR 11	PORR 10	PORR 09	PORR 08	PORR 07	PORR 06	PORR 05	PORR 04	PORR 03	PORR 02	PORR 01	PORR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	POSR 15	POSR 14	POSR 13	POSR 12	POSR 11	POSR 10	POSR 09	POSR 08	POSR 07	POSR 06	POSR 05	POSR 04	POSR 03	POSR 02	POSR 01	POSR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	POSR15 to POSR00	Pmn Output Set 0: No effect on output 1: High output	W
31:16	PORR15 to PORR00	Pmn Output Reset 0: No effect on output 1: Low output	W

Note: m = 0 to 5, 9, n = 00 to 15

The Port Control Register 3 (PCNTR3/PORR/POSR) is a 32-bit or 16-bit write register that controls the setting or resetting of the port output data.

The PCNTR3 controls the setting or resetting of the port output data, and is accessed in 32-bit units.

The POSRn (bits [15:0] in PCNTR3) and the PORRn (bits [31:16] in PCNTR3) respectively, are accessed in 16-bit units.

POSRn bits (Pmn Output Set)

POSR changes PODR when set by a software write. For example, for P100, when PORT1.PCNTR3.POSR00 = 1, PORT1.PCNTR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0. P200 is input only, so PORT2.PCNTR3.PORR00 bit is reserved.

PORRn bits (Pmn Output Reset)

PORR changes PODR when reset by a software write. For example, for P100, when PORT1.PCNTR3.POSR00 = 1, PORT1.PCNTR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. P200 is input only, so PORT2.PCNTR3.PORR00 bit is reserved.

Note: When EORRn or EOSRn is set, writing is prohibited to PODRn, PORRn, and POSRn.
Note: PORRn and POSRn should not be set at the same time.

17.2.4 PCNTR4/EORR/EOSR : Port Control Register 4

Base address: PORTm = 0x4004_0000 + 0x0020 × m (m = 1, 2)

Offset address: 0x00C (PCNTR4/EORR)
0x00E (EOSR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EORR 15	EORR 14	EORR 13	EORR 12	EORR 11	EORR 10	EORR 09	EORR 08	EORR 07	EORR 06	EORR 05	EORR 04	EORR 03	EORR 02	EORR 01	EORR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EOSR 15	EOSR 14	EOSR 13	EOSR 12	EOSR 11	EOSR 10	EOSR 09	EOSR 08	EOSR 07	EOSR 06	EOSR 05	EOSR 04	EOSR 03	EOSR 02	EOSR 01	EOSR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	EOSR15 to EOSR00	Pmn Event Output Set When an ELC_PORTx signal occurs 0: No effect on output 1: High output	R/W
31:16	EORR15 to EORR0	Pmn Event Output Reset When an ELC_PORTx signal occurs 0: No effect on output 1: Low output	R/W

Note: m = 1, 2, n = 00 to 15, x = 1, 2

The Port Control Register 4 (PCNTR4/EORR/EOSR) is a 32-bit or 16-bit read/write register that controls the setting or resetting of the port output data by an event input from the ELC.

The PCNTR4 controls the setting or resetting of the port output data by an event input from the ELC, and is accessed in 32-bit units.

The EOSRn (bits [15:0] in PCNTR4) and EORRn (bits [31:16] in PCNTR4) respectively, are accessed in 16-bit units.

EOSRn bits (Pmn Event Output Set)

EOSR changes PODR when set because an ELC_PORTx signal occurs. For example, for P100 if PORT1.PCNTR4.EOSR00 is set to 1 when the ELC_PORTx occurs, PORT1.PCNTR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0. P200 is input only, so PORT2.PCNTR4.EOSR00 bit is reserved.

EORRn bits (Pmn Event Output Reset)

EORR changes PODR when reset because an ELC_PORTx signal occurs. For example, for P100 if PORT1.PCNTR4.EORR00 = 1 when the ELC_PORTx occurs, PORT1.PCNTR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. P200 is input only, so PORT2.PCNTR4.EORR00 bit is reserved.

Note: When EORRn or EOSRn is set, writing is prohibited to PODRn, PORRn, and POSRn.

Note: EORRn and EOSRn should not be set at the same time.

17.2.5 PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 5, 9, n = 00 to 15)

Base address: PFS = 0x4004_0800

Offset address: 0x000 + 0x040 × m + 0x004 × n (PmnPFS)
0x002 + 0x040 × m + 0x004 × n (PmnPFS_HA)
0x003 + 0x040 × m + 0x004 × n (PmnPFS_BY)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	PSEL[4:0]					—	—	—	—	—	—	—	PMR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ASEL	ISEL	EOFR[1:0]		—	—	—	—	—	NCODR	—	PCR	—	PDR	PIDR	PODR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0*1	0	0	x	0

Bit	Symbol	Function	R/W
0	PODR	Port Output Data 0: Low output 1: High output	R/W
1	PIDR	Pmn State 0: Low level 1: High level	R
2	PDR	Port Direction 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	PCR	Pull-up Control 0: Disable input pull-up 1: Enable input pull-up	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	NCODR	N-Channel Open-Drain Control 0: CMOS output 1: NMOS open-drain output	R/W
11:7	—	These bits are read as 0. The write value should be 0.	R/W
13:12	EOFR[1:0]	Event on Falling/Event on Rising*2 0 0: Don't care 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges	R/W
14	ISEL	IRQ Input Enable 0: Not used as an IRQn input pin 1: Used as an IRQn input pin	R/W

Bit	Symbol	Function	R/W
15	ASEL	Analog Input Enable 0: Not used as an analog pin 1: Used as an analog pin	R/W
16	PMR	Port Mode Control 0: Used as a general I/O pin 1: Used as an I/O port for peripheral functions	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
28:24	PSEL[4:0]	Peripheral Select These bits select the peripheral function. For individual pin functions, see the associated tables in this chapter.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value of P108, P201, and P300 is not 0x00000000. The initial value of P108 is 0x00010010, P201 is 0x00000010, and P300 is 0x00010010.
Note 2. Supported for port 1 and port 2.

Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) is a 32-bit, 16-bit, or 8-bit read/write control register that selects the port mn pin function, and is accessed in 32-bit units. PmnPFS_HA (PmnPFS[15:0] bits) is accessed in 16-bit units. PmnPFS_BY (PmnPFS[7:0] bits) is accessed in 8-bit units.

PODR bit (Port Output Data), PIDR bit (Port State), PDR bit (Port Direction)

The PDR, PIDR, and PODR bits serve the same function as the PCNTR. When these bits are read, the PCNTR value is read.

PCR bit (Pull-up Control)

The PCR bit enables or disables an input pull-up resistor on the individual port pins. When a pin is in the input state with the associated bit in PmnPFS.PCR set to 1, the pull-up resistor connected to the pin is enabled. When a pin is set as a general port output pin, or a peripheral function output pin, the pull-up resistor for the pin is disabled regardless of the PCR setting. The pull-up resistor is also disabled in the reset state. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

NCODR bit (N-Channel Open-Drain Control)

The NCODR bit specifies the output type for the port pins. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

EOFR[1:0] bits (Event on Falling/Event on Rising)

The EOFR[1:0] bits select the edge detection method for the port group input signal. These bits support rising, falling, or both edge detections. When the EOFR[1:0] bits are set to 01b, 10b, or 11b, the input enable of the I/O cell is asserted. Following that, the event pulse is input from the external pin, and the GPIO outputs the event pulse to the ELC. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

ISEL bit (IRQ Input Enable)

The ISEL bit specifies IRQ input pins. This setting can be used in combination with the peripheral functions, although an IRQn (external pin interrupt) of the same number must only be enabled for one pin.

ASEL bit (Analog Input Enable)

The ASEL bit specifies analog pins. When a pin is set as an analog pin by this bit:

1. Specify it as a general I/O port in the Port Mode Control bit (PmnPFS.PMR).
2. Disable the pull-up resistor in the Pull-up Control bit (PmnPFS.PCR).
3. Specify the input in the Port Direction bit (PmnPFS.PDR). The pin state cannot be read at this point. The PmnPFS register is protected by the Write-Protect Register (PWPR). Release write-protect before modifying the register.

The ISEL bit for an unspecified IRQn is reserved. The ASEL bit for an unspecified analog I/O pin is reserved.

PMR bit (Port Mode Control)

The PMR bit specifies the port pin function. Bits associated with non-existent pins are reserved. The write value should be 0.

PSEL[4:0] bits (Peripheral Select)

The PSEL[4:0] bits assign the peripheral function. For details on the peripheral settings for each product, see section 17.6. Peripheral Select Settings for Each Product.

17.2.6 PWPR : Write-Protect Register

Base address: PFS = 0x4004_0800

Offset address: 0x503

Bit position:	7	6	5	4	3	2	1	0
Bit field:	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	PFSWE	PmnPFS Register Write Enable 0: Writing to the PmnPFS register is disabled 1: Writing to the PmnPFS register is enabled	R/W
7	B0WI	PFSWE Bit Write Disable 0: Writing to the PFSWE bit is enabled 1: Writing to the PFSWE bit is disabled	R/W

PFSWE bit (PmnPFS Register Write Enable)

Writing to the PmnPFS register is enabled only when the PFSWE bit is set to 1. You must first write 0 to the B0WI bit before setting PFSWE to 1.

B0WI bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

17.2.7 PRWCNTR : Port Read Wait Control Register

Base address: PFS = 0x4004_0800

Offset address: 0x50F

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

Bit	Symbol	Function	R/W
2:0	WAIT[2:0]	Wait Cycle Control 0 0 1: Insert a 1-cycle wait 0 1 0: Insert a 2-cycle wait 0 1 1: Insert a 3-cycle wait 1 0 0: Insert a 4-cycle wait Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

WAIT[2:0] bits (Wait Cycle Control)

The WAIT[2:0] bits specify the number of wait cycles for accessing the PCNTR2 and PFS registers, and reading the port pin state by ELC event.

17.3 Operation

17.3.1 General I/O Ports

All pins except P108 and P300 operate as general I/O ports after reset. General I/O ports are organized as 16 bits per port and can be accessed by port with the Port Control Registers (PCNTRn, where n = 1 to 4), or by individual pins with the Port mn Pin Function Select register. For details on these registers, see section 17.2. Register Descriptions.

Each port has the following bits:

- Port Direction bit (PDRn), which selects input or output direction
- Port Output Data bit (PODRn), which holds data for output
- Port Input Data bit (PIDRn), which indicates the pin states
- Event Input Data bit (EIDRn), which indicates the pin state when an ELC_PORT1 or 2 signal occurs
- Port Output Set bit (POSRn), which indicates the output value when a software write occurs
- Port Output Reset bit (PORRn), which indicates the output value when a software write occurs
- Event Output Set bit (EOSRn), which indicates the output value when an ELC_PORT1 or 2 signal occurs
- Event Output Reset bit (EORRn), which indicates the output value when an ELC_PORT1 or 2 signal occurs.

17.3.2 Port Function Select

The following port functions are available for configuring each pin:

- I/O configuration: CMOS output or NMOS open-drain output, pull-up control, and drive strength
- General I/O port: Port direction, output data setting, and read input data
- Alternate function: Configured function mapping to the pin.

Each pin is associated with a Port mn Pin Function Select register (PmnPFS), which includes the associated PODR, PIDR, and PDR bits. In addition, the PmnPFS register includes the following:

- PCR: Pull-up resistor control bit that turns the input pull-up MOS on or off
- NCODR: N-channel open-drain control bit that selects the output type for each pin
- EOFR[1:0]: For selecting the edge of the event that input from the port group
- ISEL: IRQ input enable bit to specify an IRQ input pin
- ASEL: Analog input enable bit to specify an analog pin
- PMR: Port mode bit to specify the pin function of each port
- PSEL[4:0]: Port function select bits to select the associated peripheral function.

These configurations can be made by a single-register access to the Port mn Pin Function Select register. For details, see section 17.2.5. PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 5, 9, n = 00 to 15).

17.3.3 Port Group Function for ELC

In the MCU, Port 1 and Port 2 are assigned for the ELC port group function.

17.3.3.1 Behavior When ELC_PORT1 or 2 is Input from ELC

The MCU supports the two functions described in this section when an ELC_PORT1 or 2 signal comes from the ELC.

(1) Input to EIDR

For the GPI function (PDR = 0 and PMR = 0 in the PmnPFS register), when an ELC_PORT1 or 2 signal comes from the ELC, the input enable of the I/O cell is asserted, and data from the external pins are read into the EIDR bit. See Figure 17.2

For the GPO function (PDR = 1) or the peripheral mode (PMR = 1), 0 is input into the EIDR bit from the external pins.

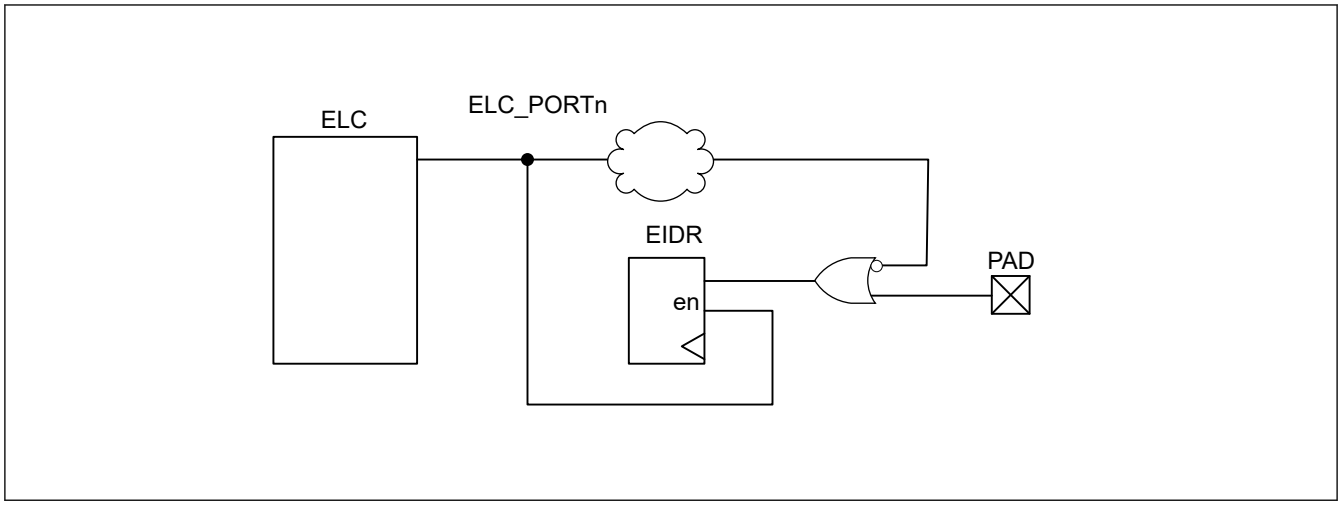


Figure 17.2 Event ports input data

(2) Output from PODR by EOSR and EORR

When an ELC_PORT1 or 2 signal occurs, the data is output from the PODR to the external pin based on the settings in the EOSR and EORR registers.

- If EOSR is set to 1, when an ELC_PORT1 or 2 signal occurs, the PODR register outputs 1 to the external pin. Otherwise, when EOSR = 0, the PODR value is retained.
- If EORR is set to 1, when ELC_PORT1 or 2 signal occurs, the PODR register outputs 0 to the external pin. Otherwise, when EORR = 0, the PODR value is retained.

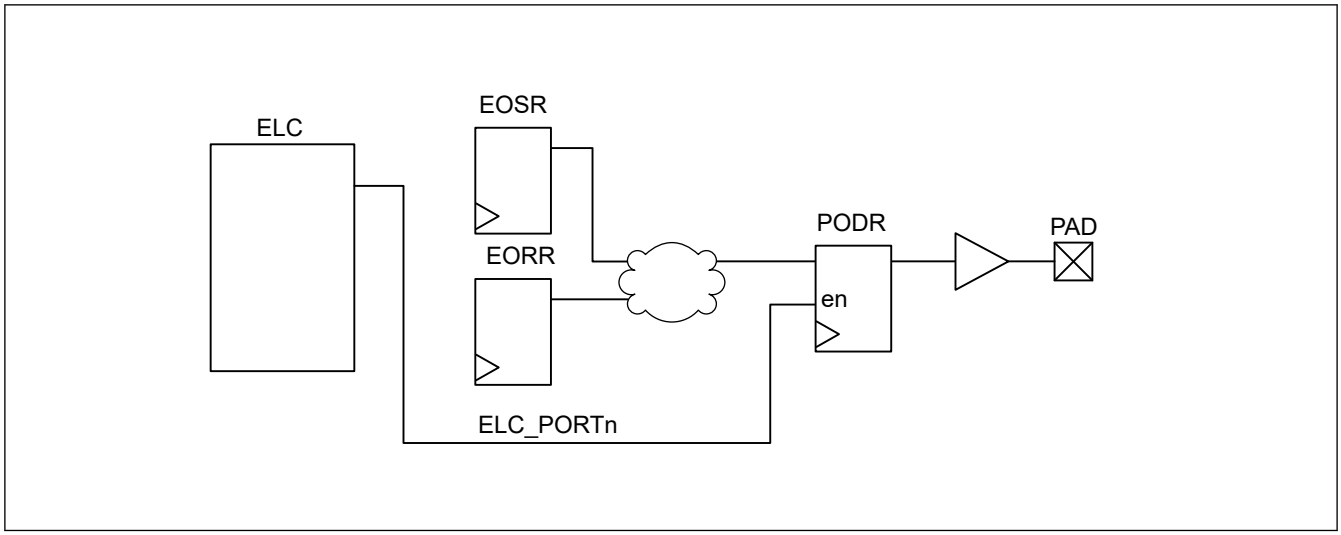


Figure 17.3 Event ports output data

17.3.3.2 Behavior When an Event Pulse is Output to ELC

To output the event pulse from the external pins to the ELC, set the EOFR[1:0] bits in the PmnPFS register. For details, see section 17.2.5. PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 5, 9, n = 00 to 15). When the EOFR[1:0] bits are set, the input enable of the I/O cell is asserted.

Data from the external pin is the input. For example, for Port 1, when the data is input from P100 to P104 and P108 to P112, the data of those 10 pins is organized by OR logic. This data is formed into a one-shot pulse that goes to the ELC. The operation of Port 2 is also the same as Port 1. See Figure 17.4.

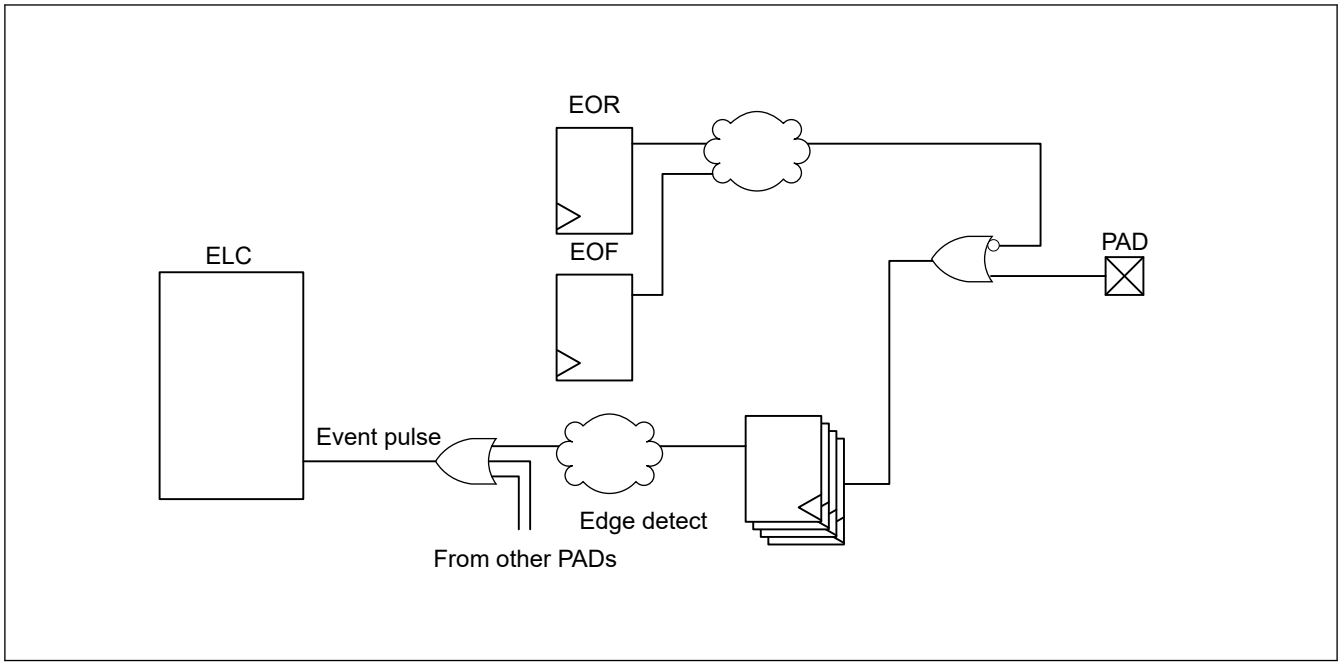


Figure 17.4 Generation of event pulse

17.3.4 Wait Function for Port Read

Wait cycles for reading the port input data can be set in the PRWCNTR.WAIT[2:0] bits as follows:

- Read port input data (PIDR) by reading the PCNTR2 or PFS register.
- Latch port pin state to Event Input Data Register (EIDR) when an ELC_PORT1 or 2 signal occurs.

The number of access cycles is the value in PRWCNTR.WAIT[2:0] bits plus 1. For example, if PRWCNTR.WAIT[2:0] bits are set as 010b, then the wait is 2 cycles, and access is 3 cycles.

Table 17.3 shows the relationship of voltage, frequency, and wait cycles.

Table 17.3 Relationship between voltage, frequency, and wait cycles

VCC = AVCC0	Access cycles*1	Wait cycles*2
More than 2.7 V	3 to 5	2 to 4
2.4 to 2.7 V	4 to 5	3 to 4
1.8 to 2.4 V	5	4
1.6 to 1.8 V	2 to 4	1 to 3

Note 1. Bus latency is not included.
Note 2. Number of wait cycles to set in Port Read Wait Control Register (PRWCNTR).

17.4 Handling of Unused Pins

Table 17.4 shows how to handle unused pins.

Table 17.4 Handling of unused pins (1 of 2)

Pin name	Description
P201/MD	Use as a mode pin
RES	Connect to VCC through a resistor (pulling up)
P200/NMI	Connect to VCC through a resistor (pulling up)
P212/EXTAL	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P212). When this pin is not used as port P212, configure it in the same way as ports 0 to 5, 9.

Table 17.4 Handling of unused pins (2 of 2)

Pin name	Description
P213/XTAL	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P213). When this pin is not used as port P213,it is configured in the same way as ports 0 to 5, 9. When the external clock is input to the EXTAL pin, leave this pin open.
P1x to P5x, P9x	• If the direction setting is for input (PCNTR1.PDRn = 0), connect the associated pin to VCC (pulled up) through a resistor or to VSS (pulled down) through a resistor*1,*2 • If the direction setting is for output (PCNTR1.PDRn = 1), release the pin*1
P000 to P002, P010 to P015	If the direction setting is for input (PCNTR1.PDRn = 0), connect the associated pin to AVCC0 (pulled up) through a resistor or to AVSS0 (pulled down) through a resistor.*1
VREFH0	Connect to AVCC0
VREFL0	Connect to AVSS0

Note 1. Clear the PmnPFS.PMR, PmnPFS.ISEL, PmnPFS.PCR, and PmnPFS.ASEL bits to 0.
Note 2. P108, P201, and P300 should be enabled for input pull-up from the initial value (PmnPFS.PCR = 1).

17.5 Usage Notes

17.5.1 Procedure for Specifying the Pin Functions

To specify the I/O pin functions:

1. Clear the B0WI bit in the PWPR register. This enables writing to the PFSWE bit in the PWPR register.
2. Set 1 to the PFSWE bit in the PWPR register. This enables writing to the PmnPFS register.
3. Clear the Port Mode Control bit in the PMR to 0 for the target pin to select the general I/O port.
4. Specify the I/O function for the pin through the PSEL[4:0] bits settings in the PmnPFS register.
5. Set the PMR bit to 1 as required to switch to the selected I/O function for the pin.
6. Clear the PFSWE bit in the PWPR register. This disables writing to the PmnPFS register.
7. Set 1 to the B0WI bit in the PWPR register. This disables writing to the PFSWE bit in the PWPR register.

17.5.2 Procedure for Using Port Group Input

To use the port group input (port 1 and port 2):

1. Set the ELSRx.ELS[7:0] bits to all 0 to ignore unexpected pulses. For more information, see section 16, Event Link Controller (ELC).
2. Set the EOFR[1:0] bits of the PmnPFS register to specify the rising, falling, or both edge detections.
3. Execute a dummy read or wait for a short time, for example 100 ns. Ignoring of unexpected pulses depends on the initial value of the external pin.
4. Set the ELSRx.ELS[7:0] bits to enable the event signals.

17.5.3 Port Output Data Register (PODR) Summary

This register outputs data as follows:

1. Outputs 0 if PCNTR4.EORR is set to 1 when ELC_PORT1 or 2 signal occurs.
2. Outputs 1 if PCNTR4.EOSR is set to 1 when ELC_PORT1 or 2 signal occurs.
3. Outputs 0 if PCNTR3.PORR is set to 1.
4. Outputs 1 if PCNTR3.POSR is set to 1.
5. Outputs 0 or 1 because PCNTR1.PODRn is set.
6. Outputs 0 or 1 because PmnPFS.PODRn is set.

Numbers in this list correspond to the priority for writing to the PODRn. For example, if 1. and 3. from the list occur at the same time, the higher priority event 1. is executed.

17.5.4 Notes on Using Analog Functions

To use an analog function, set the Port Mode Control bit (PMR) and the Port Direction bit (PDRn) to 0 so that the pin acts as a general input port. Next, set the Analog Input Enable bit (ASEL) in the Port mn Pin Function Select Register (PmnPFS.ASEL) to 1.

17.6 Peripheral Select Settings for Each Product

This section describes the pin function select configuration by the PmnPFS register. Some pin names have a _A, _B, _C, _D, _E, or _F suffix. The suffix can be ignored when assigning functionality, but assigning the same function to two or more pins simultaneously is prohibited. Only the allowed values (functions) should be specified in the PSEL bits of PmnPFS. If a value that is not allowed for the register is specified, the correct operation is not guaranteed.

Table 17.5 Register settings for input/output pin function (PORT0)

PSEL[4:0] settings	Function	Pin								
		P000	P001	P002	P010	P011	P012	P013	P014	P015
00000b	(initial)	Hi-Z								
00010b	GPT	—	—	—	—	—	—	GTIW_A	GTIV_A	GTIU_A
ASEL bit		AN008	AN009	AN010	AN005/ VREFH0	AN006/ VREFL0	AN007	AN000/ CMPIN03	AN001/ CMPIN02	AN002/ CMPIN01
ISEL bit		—	IRQ7_A	—	—	—	—	IRQ1_C	IRQ3_B	IRQ2_C
NCODR bit		—	—	—	—	—	—	—	—	—
PCR bit		✓	✓	✓	✓	✓	✓	✓	✓	✓
48-pin product		✓	✓	✓	✓	✓	✓	✓	✓	✓
32-pin product		—	—	—	✓	✓	✓	✓	✓	✓
24-pin product		—	—	—	✓	✓	—	✓	✓	✓

✓: Available
—: Setting prohibited

Table 17.6 Register settings for input/output pin function (PORT1)

PSEL[4:0] settings	Function	Pin									
		P100	P101	P102	P103	P104	P108	P109	P110	P111	P112
00000b	(initial)	Hi-Z					SWDIO	Hi-Z			
00001b	AGT	AGTIO0_A	AGTEE0	AGTO0	—	—	AGTEE1_B	AGTOA1_B	AGTIO1_A	AGTOA0	AGTOB0
00010b	GPT	GTOVLO_B	GTOVUP_B	GTOWLO_B	GTOWUP_B	GTETRGB_B	GTETRGB_C	GTOWUP_A	GTOWLO_A	—	—
00011b	GPT	GTIOC1B_B	GTIOC1A_B	GTIOC2B_A	GTIOC2A_A	—	GTIOC3B_A	GTIOC2A_B	GTIOC2B_B	GTIOC1A_C	GTIOC1B_C
00100b	SCI	RXD0_A/ MISO0_A/ SCL0_A	TXD0_A/ MOSI0_A/ SDA0_A	SCK0_A	CTS0_RTS0_ A/SS0_A	RXD0_C/ MISO0_C/ SCL0_C	RXD1/ MISO1/SCL1	SCK1_E	CTS2_RTS2_ B/SS2_B	SCK2_B	—
00101b	SCI	SCK1_A	CTS1_RTS1_ A/SS1_A	TXD2_D/ MOSI2_D/ SDA2_D	RXD2_D/ MISO2_D/ SCL2_D	—	CTS9_RTS9_ B/SS9_B	TXD9_B/ MOSI9_B/ SDA9_B	RXD9_B/ MISO9_B/ SCL9_B	SCK9_B	SCK1_D
00110b	SPI	MISOA_A	MOSIA_A	RSPCKA_A	SSLA0_A	SSLA1_A	—	—	—	—	—
00111b	IIC	SCL0_D	SDA0_C	—	—	—	—	—	—	—	—
01000b	KINT	KRM00	KRM01	KRM02	KRM03	KRM04	—	—	—	—	—
01001b	CLKOUT/ ACMPHS	—	—	—	—	—	—	CLKOUT_B	VCOUT	—	—
01010b	CAC/ADC12	—	—	ADTRG0_A	—	—	—	—	—	—	—
11011b	GPT	—	—	—	GTCPPO3	—	GTCPPO2	—	—	—	—
ASEL bit		AN022	AN021	AN020/ CMPREF0	AN019/ CMPIN1	—	—	—	—	—	—
ISEL bit		IRQ0_C	IRQ2_A	IRQ1_A	IRQ4_B	IRQ1_B	IRQ6_C	—	IRQ3_A	IRQ4_A	—
NCODR bit		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
48-pin product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
32-pin product		✓	✓	✓	✓	—	✓	✓	✓	—	✓
24-pin product		✓	✓	✓	✓	—	✓	✓	✓	—	—

✓: Available
—: Setting prohibited

Table 17.7 Register settings for input/output pin function (PORT2)

PSEL[4:0] settings	Function	Pin						
		P200	P201	P206	P207	P208	P212	P213
00000b	(initial)	Hi-Z						
00001b	AGT	—	—	—	—	AGTOB0_A	AGTEE1_A	—
00010b	GPT	—	—	GTIU_B	—	GTOUUP_B	GTOULO_A	GTOUUP_A
00011b	GPT	—	—	—	—	GTIOC0A_B	GTIOC0B_A	GTIOC0A_A
00100b	SCI	—	—	TXD0_C/ MOSI0_C/SDA0_C	—	—	—	—
ASEL bit		—	—	—	—	—	—	—
ISEL bit		—	—	IRQ0_B	—	—	IRQ7_C	IRQ2_B
NCODR bit		—	✓	✓	✓	✓	✓	✓
PCR bit		—	✓	✓	✓	✓	✓	✓
48-pin product		✓	✓	✓	✓	✓	✓	✓
32-pin product		✓	✓	—	✓	✓	✓	✓
24-pin product		✓	✓	—	—	—	✓	✓

✓: Available
—: Setting prohibited

Table 17.8 Register settings for input/output pin function (PORT3)

PSEL[4:0] settings	Function	Pin		
		P300	P301	P302
00000b	(initial)	SWCLK	Hi-Z	
00001b	AGT	—	AGTIO0_D	—
00010b	GPT	GTETRGA_C	GTETRGB_A	GTETRGA_D
00011b	GPT	GTIOC3A_A	—	—
00100b	SCI	—	RXD2_A/MISO2_A/SCL2_A	TXD2_A/MOSI2_A/SDA2_A
00101b	SCI	TXD1/MOSI1/SDA1	CTS9_RTS9_D/SS9_D	—
11011b	GPT	GTCPPO1	—	—
ASEL bit		—	—	—
ISEL bit		IRQ5_B	IRQ6_A	IRQ5_A
NCODR bit		✓	✓	✓
PCR bit		✓	✓	✓
48-pin product		✓	✓	✓
32-pin product		✓	—	—
24-pin product		✓	—	—

✓: Available
—: Setting prohibited

Table 17.9 Register settings for input/output pin function (PORT4) (1 of 2)

PSEL[4:0] settings	Function	Pin						
		P400	P401	P402	P403	P407	P408	P409
00000b	(initial)	Hi-Z						
00001b	AGT	AGTIO1_C	—	—	—	AGTIO0_C	—	—
00010b	GPT	—	GTETRGA_B	—	—	GTOULO_B	GTIW_B	GTIV_B
00011b	GPT	GTIOC3A_B	GTIOC3B_B	GTADSM0	GTADSM1	GTIOC0B_B	—	—
00100b	SCI	SCK0_B	CTS0_RTS0_B/ SS0_B	—	—	CTS0_RTS0_D/ SS0_D	CTS1_RTS1_D/ SS1_D	—
00101b	SCI	SCK1_B	—	—	—	—	—	—
00111b	IIC	SCL0_A	SDA0_A	—	—	SDA0_B	SCL0_C	—
01010b	CAC/AD	CACREF_C	—	—	—	ADTRG0_B	—	—
11011b	GPT	—	—	—	—	GTCPPO0	—	—
ASEL bit		—	—	—	—	CMPREF1	—	—
ISEL bit		IRQ0_A	IRQ5_C	—	—	—	IRQ7_B	IRQ6_B
NCODR bit		✓	✓	✓	✓	✓	✓	✓

Table 17.9 Register settings for input/output pin function (PORT4) (2 of 2)

PSEL[4:0] settings	Function	Pin						
		P400	P401	P402	P403	P407	P408	P409
PCR bit		✓	✓	✓	✓	✓	✓	✓
48-pin product		✓	✓	✓	✓	✓	✓	✓
32-pin product		—	—	✓	✓	✓	—	—
24-pin product		—	—	—	—	✓	—	—

✓: Available
—: Setting prohibited

Table 17.10 Register settings for input/output pin function (PORT5)

PSEL[4:0] settings	Function	Pin
		P500
00000b	(initial)	Hi-Z
00010b	GPT	GTETRGA_A
00011b	GPT	GTIOC2A_C
ASEL bit		—
ISEL bit		—
NCODR bit		✓
PCR bit		✓
48-pin product		✓
32-pin product		—
24-pin product		—

✓: Available
—: Setting prohibited

Table 17.11 Register settings for input/output pin function (PORT9)

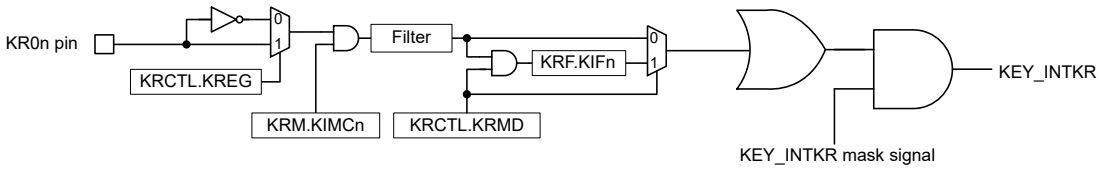
PSEL[4:0] settings	Function	Pin		
		P913	P914	P915
00000b	(initial)	Hi-Z		
00001b	AGT	AGTIO1_F	AGTOA1_A	—
00010b	GPT	GTOVLO_A	GTOVUP_A	—
00011b	GPT	GTIOC1B_A	GTIOC1A_A	—
NCODR bit		✓	✓	✓
PCR bit		✓	✓	✓
48-pin product		✓	✓	✓
32-pin product		✓	✓	—
24-pin product		—	—	—

✓: Available
—: Setting prohibited

18. Key Interrupt Function (KINT)

18.1 Overview

The key interrupt function (KINT) generates the key interrupt by detecting rising or falling edge on the key interrupt input pins. Figure 18.1 shows a block diagram and Table 18.1 lists the input pins.



Note: n = 0 to 4

Figure 18.1 KINT block diagram

All key return factors are merged by an OR gate, and the key interrupt signal, KEY_INTKR, is the output of the AND gate to mask the merged key return factor by the KEY_INTKR mask signal. When using KRF.KIFn flag (KRCTL.KRMD = 1), the KEY_INTKR mask signal is used as the output mask that is asserted by clearing KRF.KIFn flag.

Table 18.1 KINT I/O pins

Pin name	I/O	Function
KR00 to KR04	Input	Key interrupt input pins

18.2 Register Descriptions

18.2.1 KRCTL : Key Return Control Register

Base address: KINT = 0x4008_0000

Offset address: 0x00

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

Bit	Symbol	Function	R/W
0	KREG	Detection Edge Selection (KR00 to KR04 pins) 0: Falling edge 1: Rising edge	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	KRMD	Usage of Key Interrupt Flags (KRF.KIF0 to KRF.KIF4) 0: Do not use key interrupt flags 1: Use key interrupt flags	R/W

The KRCTL register controls the usage of the key interrupt flags, KRF.KIFn (n = 0 to 4), and sets the detection edge.

18.2.2 KRF : Key Return Flag Register

Base address: KINT = 0x4008_0000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	KIF4	KIF3	KIF2	KIF1	KIF0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	KIF0 to KIF4	Key Interrupt Flag n 0: No interrupt detected 1: Interrupt detected	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The KRF register controls the key interrupt flags, KIFn.

When KRCTL.KRMD = 0, setting the KIFn flag to 1 is prohibited. When setting the KIFn flag to 1, the KIFn value does not change.

To clear the KIFn flag, confirm the target flag is 1 before writing 0 to the bit, then write 1 to the other flags.

18.2.3 KRM : Key Return Mode Register

Base address: KINT = 0x4008_0000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	KIMC4	KIMC3	KIMC2	KIMC1	KIMC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	KIMC0 to KIMC4	Key Interrupt Mode Control n 0: Do not detect key interrupt signals 1: Detect key interrupt signals	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The KRM register sets the key interrupt mode.

An interrupt is generated when the target bit in the KRM register is set while a low level (KRCTL.KREG = 0) or a high level (KRCTL.KREG = 1) is being input to the KR0n pin. To ignore this interrupt, set the KRM register after disabling the interrupt handling.

KINT can be assigned in the PmnPFS.PSEL[4:0] bits. The on-chip pull-up resistors can also be applied by setting the associated key interrupt input pin in the pull-up resistor. For details, see section 17, I/O Ports.

18.3 Operation

18.3.1 Operation When Not Using the Key Interrupt Flags (KRCTL.KRMD = 0)

A key interrupt signal, KEY_INTKR, is generated when the valid edge specified in the KRCTL.KREG bit is input to a KR0n pin. To identify the channel to which the valid edge is input, read the port register and check the port level after the KEY_INTKR signal is generated.

The KEY_INTKR signal changes based on the input level of the KR0n pin.

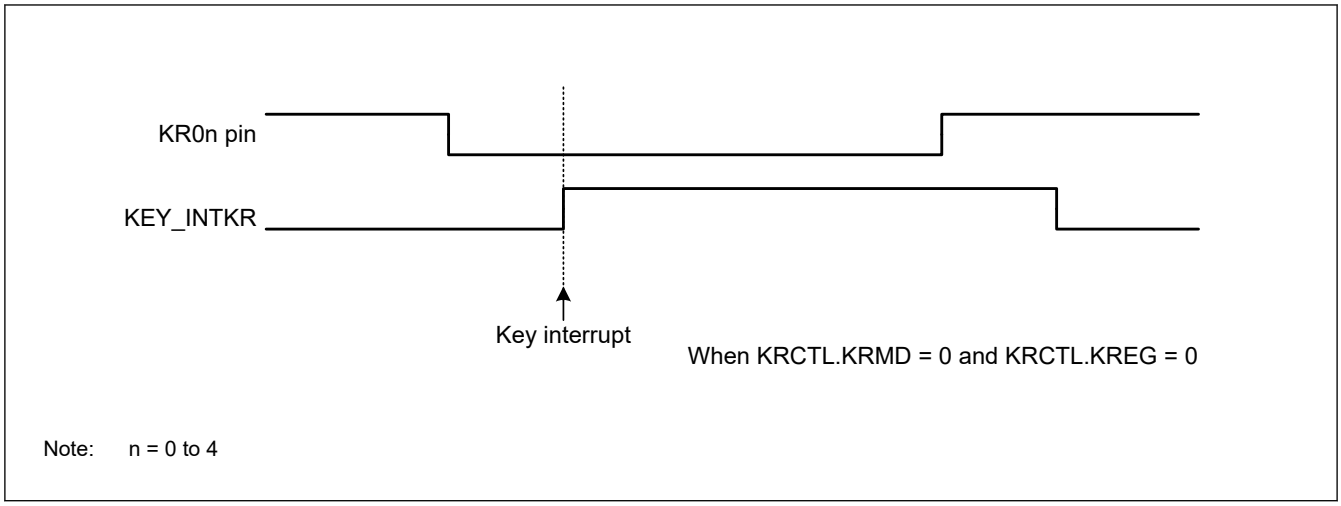


Figure 18.2 Operation of KEY_INTKR signal when a key interrupt is input to a single channel

Figure 18.3 shows the operation when a valid edge is input to multiple KR0n pins. The KEY_INTKR signal is set while a low level is being input to one pin (when KRCTL.KREG = 0). Therefore, even if a falling edge is input to another pin in this period, the KEY_INTKR signal is not generated again. See [1] in Figure 18.3.

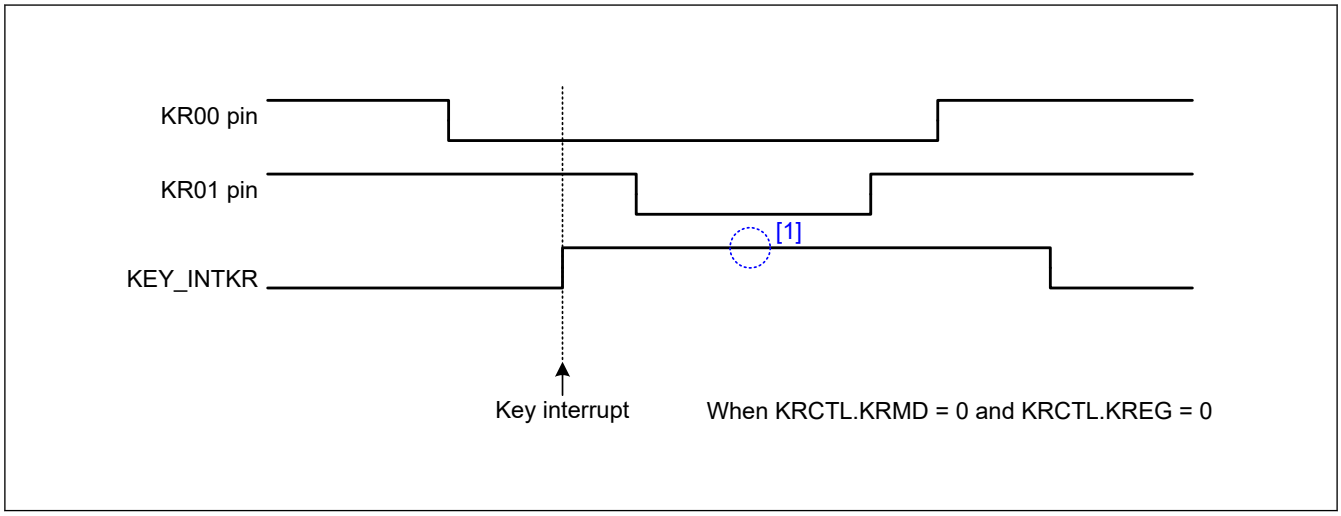


Figure 18.3 Operation of KEY_INTKR signal when key interrupts are input to multiple channels

18.3.2 Operation When Using the Key Interrupt Flags (KRCTL.KRMD = 1)

The KEY_INTKR signal is generated when the valid edge specified in the KRCTL.KREG bit is input to KR0n pins. To identify the channels to which the valid edge is input, read the KRF register after the KEY_INTKR signal is generated. If the KRCTL.KRMD bit is set to 1, clear the KEY_INTKR signal by clearing the associated bit in the KRF register.

As Figure 18.4 shows, only one interrupt is generated each time a falling edge is input to one channel, (when KRCTL.KREG = 0), regardless of whether the KRF.KIFn flag is cleared before or after a rising edge is input.

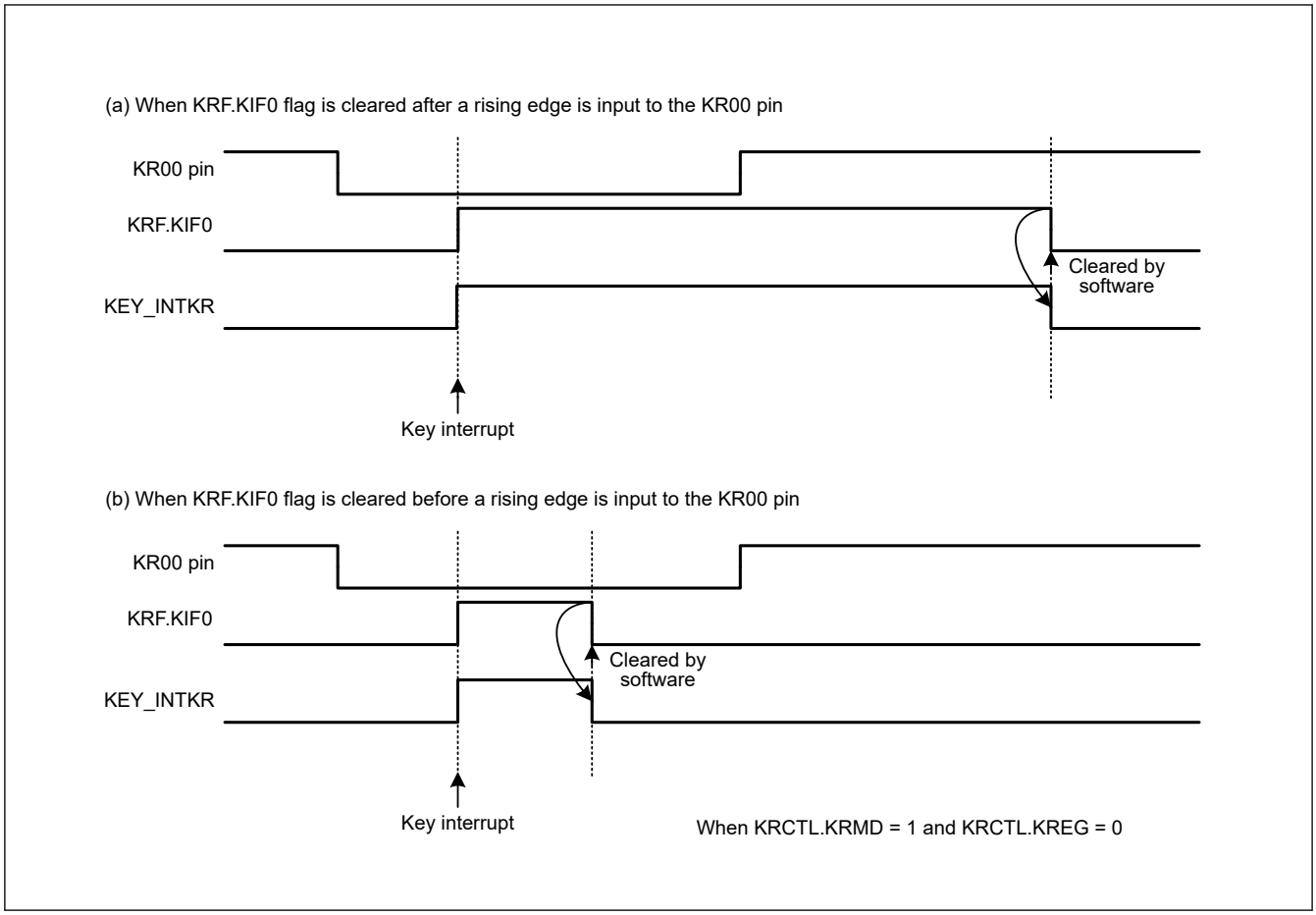


Figure 18.4 Basic operation of KEY_INTKR signal when key interrupt flag is used

Figure 18.5 shows the operation when a valid edge is input to multiple KR0n pins. A falling edge is also input to the KR01 and KR04 pins after a falling edge is input to the KR00 pin (when KRCTL.KREG = 0). The KRF.KIF1 flag is set when the KRF.KIF0 flag is cleared. The KEY_INTKR signal is negated 1 PCLKB clock cycle, after the KRF.KIF0 flag is cleared. See [1] in Figure 18.5.

Also, after a falling edge is input to the KR04 pin, the KRF.KIF4 flag is set. The KRF.KIF1 flag is cleared at time [2] in the figure. The KEY_INTKR signal is negated 1 PCLKB clock cycle, after the KRF.KIF1 flag is cleared. See [3] in the figure. It is therefore possible to generate each key interrupt when a valid edge is input to multiple channels.

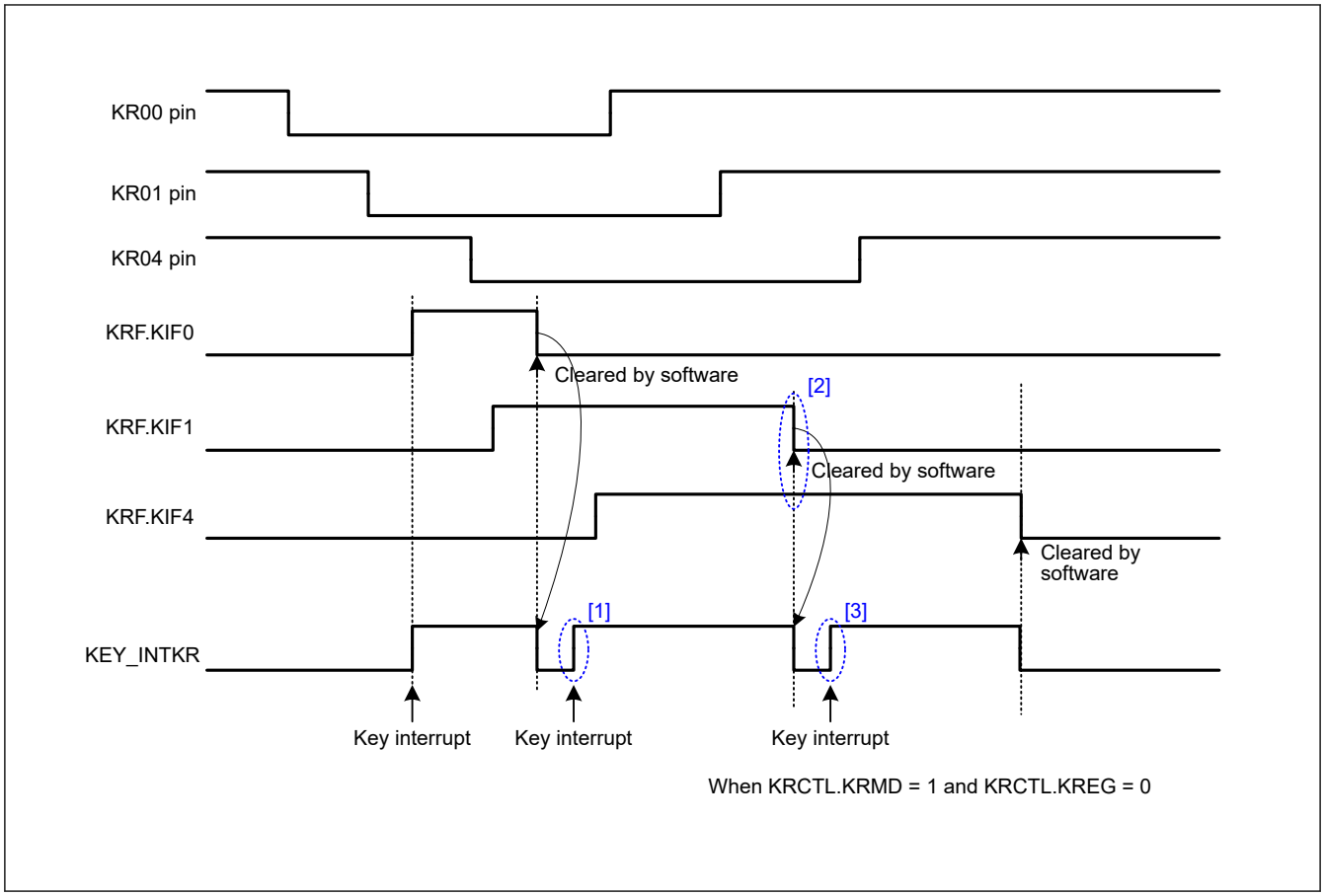


Figure 18.5 Operation of KEY_INTKR signal when key interrupts are input to multiple channels

18.4 Usage Notes

- If the KEY_INTKR signal is used as the snooze request, the KRCTL.KRMD bit should be set to 0.
- If the KEY_INTKR signal is used as the interrupt source for returning to Normal mode from Snooze and Software Standby modes, the KRCTL.KRMD bit should be set to 1.
- When KINT is assigned to a pin, this pin input is always enabled in the Software Standby mode, and if the pin level changes, the associated KRF.KIFn flag can be set. Therefore, a KEY_INTKR signal might be generated on canceling Software Standby mode. To ignore changes to the KR0n pin during a Software Standby, clear the associated KRM.KIMCn bit before entering Software Standby. After canceling Software Standby mode, the KRF.KIFn flag should be cleared before the associated KRM.KIMCn bit can be set.

19. Port Output Enable for GPT (POEG)

19.1 Overview

The Port Output Enable (POEG) function can place the General PWM Timer (GPT) output pins in the output disable state in one of the following ways:

- Input level detection of the GTETRGn pin (n = A, B)
- Detection from the GPT to stop output
- Detection by comparator (edge detection or level detection)
- Detection of stopping of oscillation by the oscillation stop detection circuit for the main clock
- Register setting

The GTETRGn pin can be used for output to the external trigger input pins of the GPT.

Table 19.1 shows specifications, Figure 19.1 shows a block diagram, and Table 19.2 shows input pins.

Table 19.1 POEG specifications

Item	Description
Request of output stopping in response to input level detection	• The request to stop output is issued to the GPT when a POEGGn.PIDF flag is set in response to the detection of input of the selected level on the corresponding GTETRGn pin (n = A, B). • The request to stop output is issued to the GPT immediately upon detection of input of the selected level on the corresponding GTETRGn pin.
Requests to stop output in the form of an output stopping signal from the GPT	• The request to stop output is issued to the GPT when the GPT detects the active level (high or low) on the GTIOCA and GTIOCB pins at the same time while the corresponding POEGGn.IOCF flag is set. • The request to stop output is issued to the GPT when the GPT detects a deadtime error while the corresponding POEGGn.IOCF flag is set.
Requests to stop output in response to detection by a comparator	• The request to stop output is issued to the GPT when a POEGGn.IOCF flag is set in response to edge-detection by a comparator. • The request to stop output is issued directly to the GPT upon detecting level on a comparator.
Requests to stop output in response to detecting the stopping of oscillation	A request to stop output is issued to the GPT when the oscillation stop detection circuit for the main clock detects the stopping of oscillation while the corresponding POEGGn.OSTPF flag is set.
Requests by software to stop output	The request to stop output is issued to the GPT when the software sets the POEGGn.SSF flag.
Interrupt	• An interrupt is generated in response to the request to stop output in the form of the POEGGn.PIDF flag. • An interrupt is generated in response to the request to stop output in the form of the POEGGn.IOCF flag.
External trigger output to the GPT	The GTETRGn pin is used for output as external triggers.
Noise removal	• Each GTETRGn pin has a digital noise filter. • Four types of sampling clock are selectable for the filter.

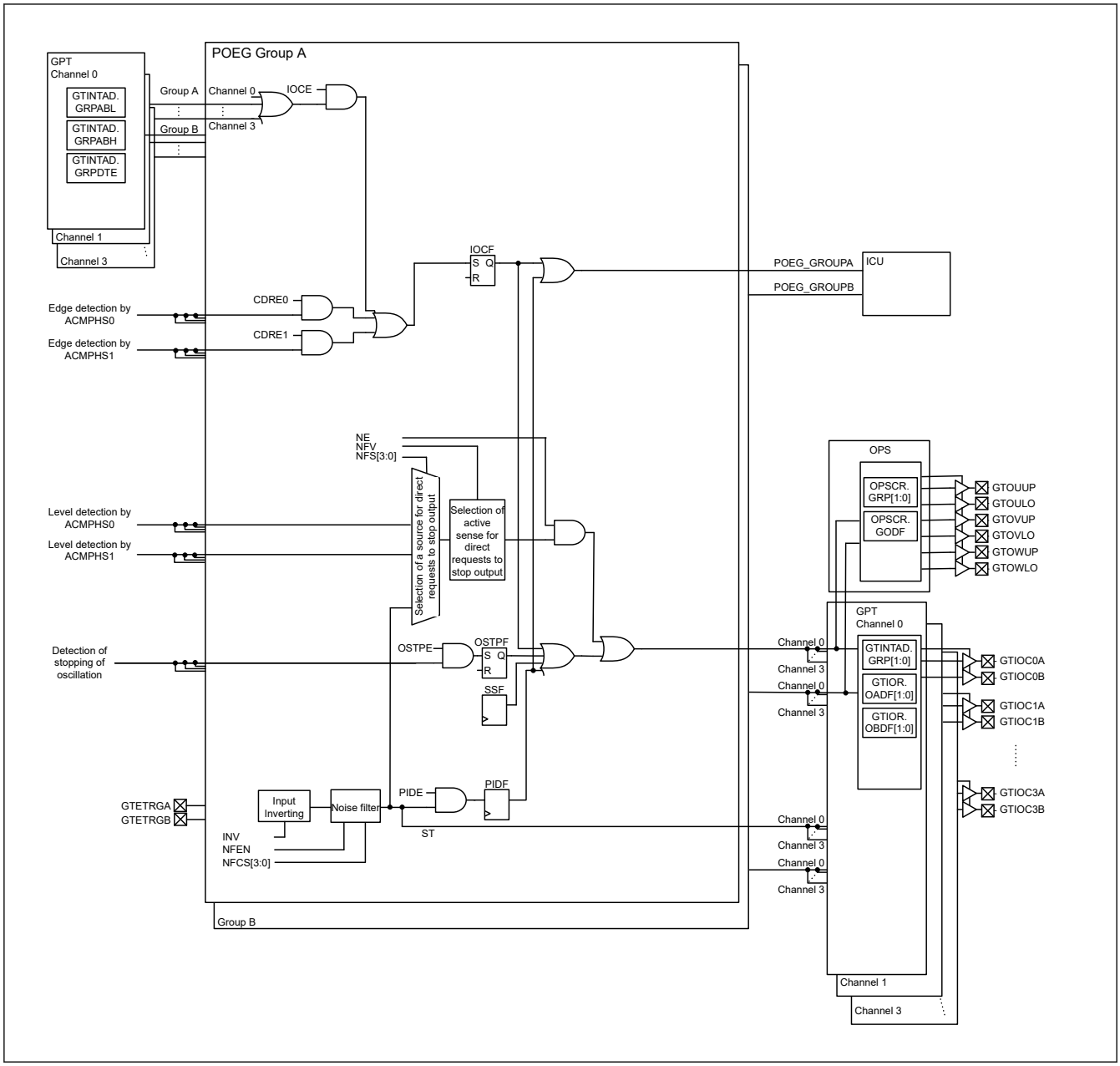


Figure 19.1 POEG Block Diagram

Table 19.2 POEG I/O pins

Pin name	I/O	Function
GTETRGA	Input	Output disable detection signal for GPT output pin and GPT external trigger input pin A
GTETRGB	Input	Output disable detection signal for GPT output pin and GPT external trigger input pin B

19.2 Register Descriptions

19.2.1 POEGGn : POEG Group n Setting Register (n = A, B)

Base address: POEG = 0x4004_2000

Offset address: 0x000 (POEGGA)
0x100 (POEGGB)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCS[1:0]		NFEN	INV	—	—	—	—	—	—	—	—	—	—	—	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CDRE1	CDRE0	—	OSTPE	IOCE	PIDE	SSF	OSTPF	IOCF	PIDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIDF	Port Input Detection Flag 0: The selected input level was not detected on the GTETRGn pin 1: The selected input level was detected on the GTETRGn pin	R/W*1
1	IOCF	GPT or ACMPHS Output Stop Request Detection Flag 0: Neither stopping of GPT output nor a comparator edge was detected 1: Either stopping of GPT output or comparator edge was detected	R/W*1
2	OSTPF	Oscillation Stop Detection Flag 0: Stopping of oscillation was not detected 1: Stopping of oscillation was detected	R/W*1
3	SSF	Software Stop Flag 0: Software has not stopped output 1: Software has stopped output	R/W
4	PIDE	Port Input Detection Enable 0: Detection of input levels on the corresponding GTETRGn pin is disabled 1: Detection of input levels on the corresponding GTETRGn pin is enabled	R/W*2
5	IOCE	GPT Output Stop Request Enable 0: Detection of stopping of output from the GPT is disabled 1: Detection of stopping of output from the GPT is enabled	R/W*2
6	OSTPE	Enable Stopping Output on Stopping of Oscillation 0: Detection of stopping of oscillation is disabled 1: Detection of stopping of oscillation is enabled	R/W*2
7	—	This bit is read as 0. The write value should be 0.	R/W
8	CDRE0	ACMPHS0 Enable 0: Comparator edge detection 0 is disabled 1: Comparator edge detection 0 is enabled	R/W*2
9	CDRE1	ACMPHS1 Enable 0: Comparator edge detection 1 is disabled 1: Comparator edge detection 1 is enabled	R/W*2
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	ST	GTETRGn Input Status Flag 0: The corresponding external trigger for output to the GPT is 0 1: The corresponding external trigger for output to the GPT is 1	R
27:17	—	These bits are read as 0. The write value should be 0.	R/W
28	INV	GTETRGn Input Inverting 0: Input on the GTETRGn pin is not inverted 1: Input on the GTETRGn pin is inverted	R/W

Bit	Symbol	Function	R/W
29	NFEN	Noise filter Enable 0: Digital noise filter on the GTETRGn pin is disabled 1: Digital noise filter on the GTETRGn pin is enabled	R/W
31:30	NFCS[1:0]	Noise filter Clock Select 0 0: Samples the input level of GTETRGn pin three times per PCLKB/1 clock 0 1: Samples the input level of GTETRGn pin three times per PCLKB/8 clock 1 0: Samples the input level of GTETRGn pin three times per PCLKB/32 clock 1 1: Samples the input level of GTETRGn pin three times per PCLKB/128 clock	R/W

Note 1. Only 0 can be written to clear the flag.
Note 2. Can be modified only once after a reset.

The POEGGn register (n = A, B) controls requests for stopping output and an external trigger for the GPT based in response to the detection of various signals.

SSF Flag (Software Stop Flag)

Writing 1 to the SSF flag leads to a request to stop output being issued to the GPT, and writing 0 to the flag releases the GPT from the request to stop output. In addition, requests to stop output issued by software can be monitored by reading the flag.

19.2.2 GTONCWPn : GPT Output Stopping Control Group n Write Protection Register (n = A, B)

Base address: POEG = 0x4004_2000

Offset address: 0x040 (GTONCWPA)
0x140 (GTONCWPB)

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

Bit	Symbol	Function	R/W
0	WP	Register Writing Disable 0: Writing to the GTONCCRn register is enabled 1: Writing to the GTONCCRn register is disabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	Key Code These bits control whether or not writing new values to the GTONCWPn register is possible. To write to the GTONCWPn register, write 0xA5 to the 8 higher-order bits and write any value to make up a 16-bit unit to the 8 lower-order bits at the same time. These bits are read as 0x00.	R/W

The GTONCWPn register (n = A, B) enables writing to the GTONCCRn register in order to avoid incorrect writing to registers.

19.2.3 GTONCCRn : GPT Output Stopping Control Group n Controlling Register (n = A, B)

Base address: POEG = 0x4004_2000

Offset address: 0x044 (GTONCCRA)
0x144 (GTONCCRB)

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

Bit	Symbol	Function	R/W
0	NE	Direct Stopping Request Setting 0: The signal for detection is not set as a direct stopping request signal 1: The signal for detection is set as a direct stopping request signal	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
7:4	NFS[3:0]	Direct Stopping Request Selection Other settings are prohibited. 0x0: Comparator level detection 0 0x1: Comparator level detection 1 0x7: GTETRGn pin input level detection (n = A, B) Others: Setting prohibited	R/W
8	NFV	Direct Stopping Request Active Sense 0: Stopping output is requested when the output stopping detection signal is 0 1: Stopping output is requested when the output stopping detection signal is 1	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

The GTONCCRn register (n = A, B) set up requests for stopping output in response to detected signals.

NE bit (Direct Stopping Request Setting)

Writing 1 to the NE bit leads to direct output of the stopping request signal in response to the signal for detection selected by the NFS[3:0] bits.

19.3 Operation

19.3.1 Request to Stop Output in Response to Detection of Input Level on the Corresponding GTETRGn Pin (n = A, B)

There are two types of output stopping request: requests in response to setting of the POEGGn.PIDF flag (n = A, B), and requests directly in response to the detected signal.

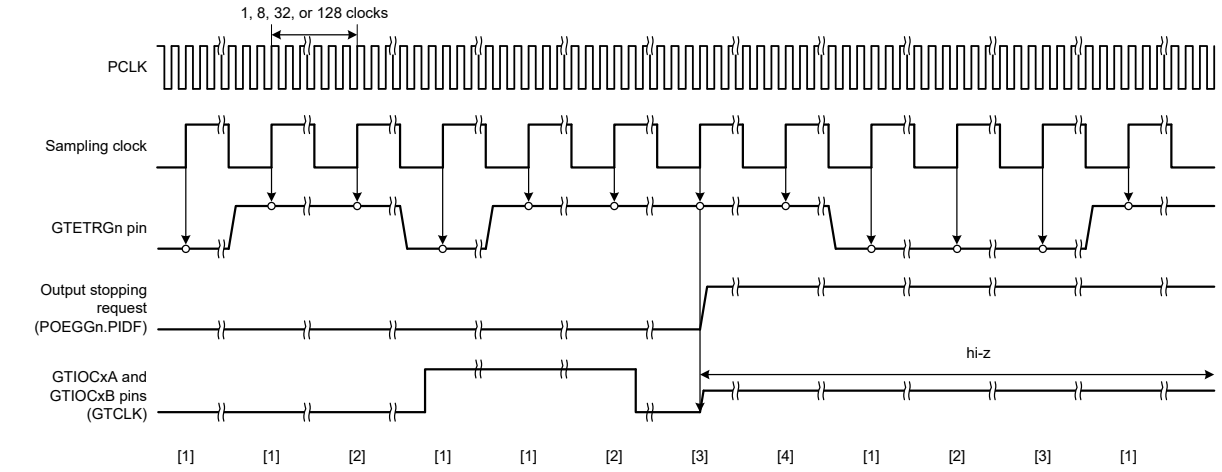
1. To request stopping of output in response to setting of the PIDF flag, the input set in the POEGGn register (inversion or non-inversion set in the INV bit, filtering enabled or disabled by the NFEN bit, sample clock for filtering set in the NFCS[1:0] bits) is detected while the POEGGn.PIDE bit is 1. After the POEGGn.PIDF flag is set to 1 in response to this, the request to disable output is issued per group to each channel of the GPT. To de-assert the request signal for stopping output, clear the POEGGn.PIDF flag. For cancelling requests to stop output, see section 19.3.6. Cancelling Requests to Stop Output.
2. To request stopping of output in direct response to detection of the input level of a selected signal, the input set in the POEGGn (inversion or non-inversion set in the INV bit, filtering enabled or disabled by the NFEN bit, sample clock for filtering set in the NFCS[1:0] bits) and GTONCCRn registers (active sense set in the NFV bit) to the GTETRGn pin selected in the GTONCCRn.NFS[3:0] bits is detected while the GTONCCRn.NE bit is 1, and a request to stop output is then directly issued per channel group of the GPT. The signal to request stopping of output is de-asserted when the detected input level does not match the conditions for issuing a request. For details, see section 19.3.7. Requests to Stop Output in Response to Detected Signals and Cancelling the Requests.

19.3.1.1 Digital Noise Filter

Each GTETRGn pin input has a digital noise filter. Figure 19.2 shows the operation example in detection of the high level with the use of the filter. When the digital noise filter is enabled (POEGGn.NFEN bit = 1) and the high level is detected 3 consecutive times with the sampling clock cycle selected by the POEGGn.NFCS[1:0] bits while inverted or non-inverted according to the setting of the POEGGn.INV bit, this is regarded as detection of the high level, and the request to stop output is issued to the GPT.

At this time, if the low level is detected even once in the sequence, it is not regarded as detection of the high level. In addition, changes in the levels on pins GTETRGA through GTETRGB are ignored while the sampling clock is not being output.

Digital noise filters can be used with requests to stop output in response to setting of the POEGGn.PIDF flags (n = A, B), requests to stop output in direct response to a detected signal, and the output of external triggers to the GPT.



Note: Each channel output can be set by the GPT16x.
Note: The polarity can be inverted by setting the POEGGn.INV bit.
Note: PCLK = PCLKB

Figure 19.2 Example for Operation of Digital Noise Filter

19.3.2 Requests to Stop Output in Response to Detection of Output Stopping from GPT

When any among a dead-time error or simultaneous high- or low-level output from the GPT is detected, the corresponding POEGGn.IOCF flag is set to 1 and the request to stop output per group is issued to each channel in the GPT. The POEGGn.IOCF flag is used to indicate both edge detection by the comparator and requests to stop output. To cancel a request to stop output, clear the given POEGGn.IOCF flag. For details, see section 19.3.6. Cancelling Requests to Stop Output.

To detect any among a dead-time error or simultaneous high- or low-level output from the GPT, detection of output stopping must be permitted in the GRPDTE, GRPABH, and GRPABL bits of the given GPT16n.GTINTAD register. Specify the group of the GPT for which stopping is to be detected in the GPT16n.GTINTAD.GRP[1:0] bits. For details, see section 20.2.15. GTINTAD : General PWM Timer Interrupt Output Setting Register.

19.3.3 Request to Stop Output in Response to Comparator Detection

A request to stop output can be issued to the GPT in response to detection by a comparator. There are two types of request to stop output: requests output in response to setting of a POEGGn.IOCF flag (n = A, B) due to edge detection by the comparator and requests that are directly output in response to level detection by the comparator.

1. To stop output in response to setting of an IOCF flag, the corresponding edge is detected by the comparator while the POEGGn.CDRE[1:0] bits are 1. The POEGGn.IOCF flag is then set to 1, which leads to a request to stop output from the given group for each channel of the GPT. The POEGGn.IOCF flag is used for both comparator edge detection and the request to stop output. To cancel the request to stop output, clear the POEGGn.IOCF flag. For details, see section 19.3.6. Cancelling Requests to Stop Output.
2. The output can also be stopped by using a detected signal as a direct request to stop output. When the comparator level detection signal selected in the GTONCCRn.NFS[3:0] bits matches the level set in the GTONCCRn.NFV bit, the request to stop output per group is issued to each channel of the GPT. The request to stop output is released when the comparator level detection does not match the issuing conditions. For details, see section 19.3.7. Requests to Stop Output in Response to Detected Signals and Cancelling the Requests.

19.3.4 Requests to Stop Output by Oscillation Stop Detection

When the circuit for detecting stopping of oscillation by the main clock oscillator detects the stopping of the oscillation while a POEGGn.OSTPE bit is 1, the POEGGn.OSTPF flag is set to 1 and a request to stop output per group is issued to each channel of the GPT. To cancel the request to stop output, clear the POEGGn.OSTPF flag. For details, see section 19.3.6. Cancelling Requests to Stop Output.

19.3.5 Requests to Stop Output by a Register

Writing 1 to the software stop flag (POEGGn.SSF) leads to a request to stop output per group to each channel in the GPT. To release from the request to disable output, clear the POEGGn.SSF flag. For details, see section 19.3.6. Cancelling Requests to Stop Output.

19.3.6 Cancelling Requests to Stop Output

Requests to stop output are cancelled in any of the following three ways.

1. Cancellation by a reset (return to the initial state)
2. Cancellation by clearing all flags in the POEGGn register
3. Cancellation in response to direct input of a detected signal

(1) Cancellation by a reset

Any type of reset can cancel a request to stop output. For details, see section 5, Resets.

(2) Cancellation by clearing all flags in the POEGGn register

- POEGGn.PIDF
- POEGGn.IOCF
- POEGGn.OSTPF
- POEGGn.SSF

Cancellation of the request is taken into the GPT at the end of the cycle of counting by the GPT, and the output pins are released from being stopped no less than 3 GTCLK cycles from that time. Figure 19.3 shows the timing of release from the output-stopped state. To clear each of the flags, read the status flag for each source to check that the source condition is not being detected, and then write 0. Sources other than the edge detection by the comparators cannot be cleared even if the flag is cleared in the detected state. Since the comparators detect edges in edge detection, writing 0 in the detected state clears the flag, and the flag is not set until the relevant source generates the next edge. Status flags for the respective sources are listed below.

Detection of input level	POEGGn.ST (GTETRGn input status flag)
Comparator edge detection	ACMPHSn.CMPMON.CMPMON (comparator output monitoring flag)
Oscillation stop detection	OSTDSR.OSTDF (oscillation stop detection flag)
Stopping detection from the GPT	GPT16n.GTST.DTEF (dead time error flag) GPT16n.GTST.OABLF (same time output level low flag) GPT16n.GTST.OABHF (same time output level high flag)

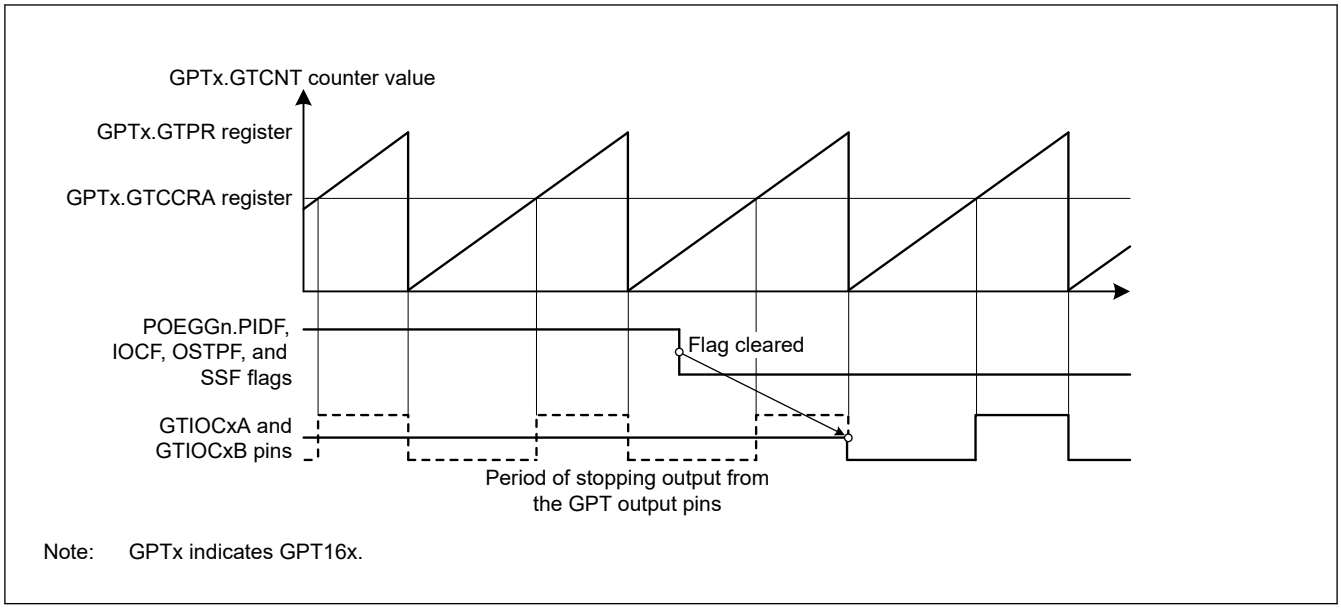


Figure 19.3 Timing of Re-enabling Output from the GPT Output Pins Following Cancellation of a Request to Stop Output

(3) Cancellation in response to direct input of a detected signal

For details, see section 19.3.7. Requests to Stop Output in Response to Detected Signals and Cancelling the Requests.

19.3.7 Requests to Stop Output in Response to Detected Signals and Cancelling the Requests

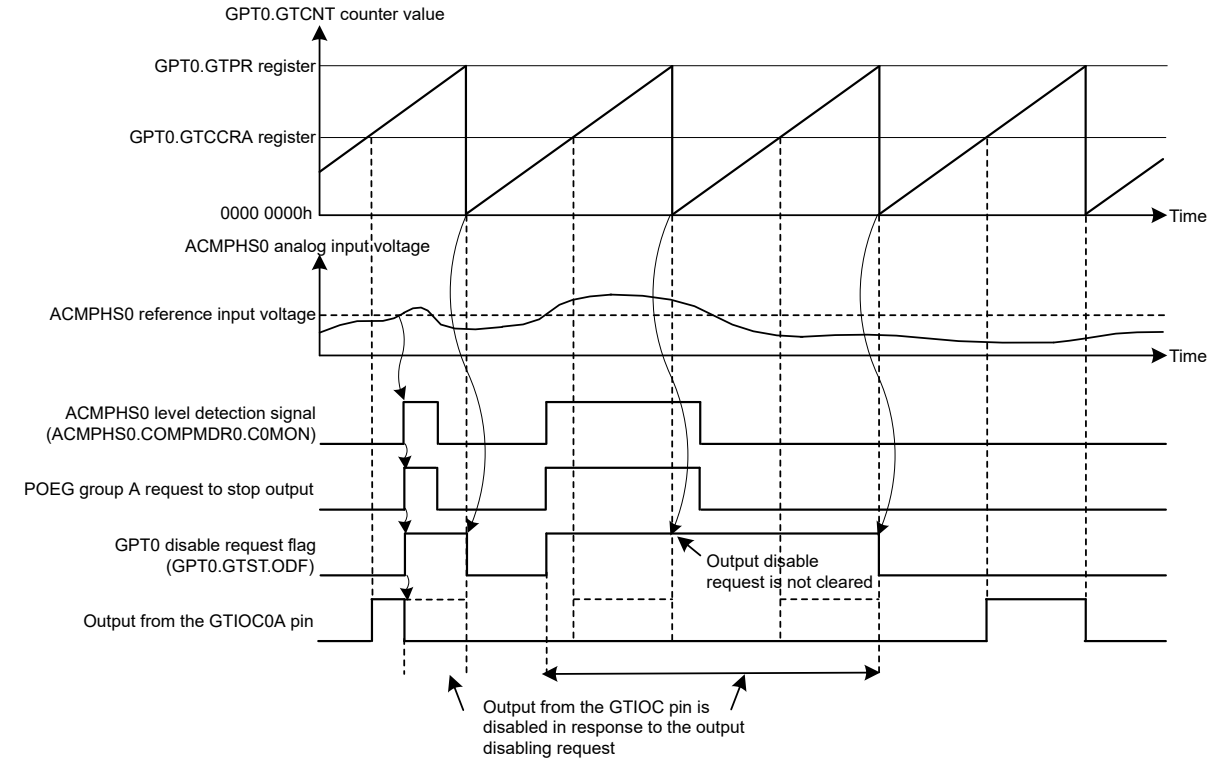
The input level detection signals on the GTETRGn pin (n = A, B) and comparator level detection signals on the ACMPHSm (m = 0, 1) can be used as direct requests to stop output in response to detected signals. The sources of the signals for detection are selected in the GTONCCRn.NFS[3:0] bits, and the active senses of the signals for detection are set in the GTONCCRn.NFV bit. Setting a GTONCCRn.NE bit to activate direct requests to stop output results in the issuing of a request to stop output to the GPT when the selected source signal for output stopping detection is generated.

The request to stop output is released when the input level detection signals on the GTETRGn pin or comparator level detection does not match the issuing conditions. To cancel a request, confirm that the values of the PIDF and IOCF flags in the POEGGn register have become 0.

Figure 19.4 shows the operation of a request to stop output issued in response to level detection by the ACMPHS0. The example shows the case where the comparator detects the analog input voltage on ACMPHS0 becoming higher than the reference voltage while the GPT is producing PWM waveforms on the GTIOC0A pin on a cyclic-counting basis, and the level-detection signal from the comparator is input to the POEG. The POEG outputs a request to stop output to the GPT on the basis of detection as described above, and the GPT remains stopped until the end of the cycle of counting, even if the request to stop output is cancelled before then.

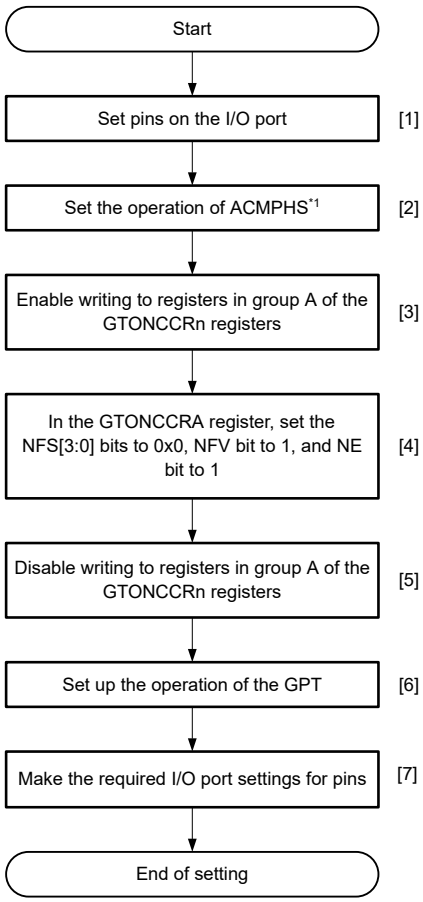
If the request to stop output has not been cancelled at the end of the cycle of counting cycle, output from the GPT remains stopped until the end of the next cycle of counting, and so on.

The output-stopped state can be checked by reading the ODF flag in General PWM Timer Status Register (GTST) of the GPT. For details, see the description of the ODF flag in section 20.2.16. GTST : General PWM Timer Status Register. Figure 19.5 shows the procedure for setting requests to stop output in response to detected signals.



Note: GPT0 indicates GPT160.

Figure 19.4 Example of Operation to Stop Output from the GTIOC Pin In Response to Level Detection by a Comparator



[1] Make the I/O port settings.
Set the corresponding pins of the I/O port to general purpose I/O port pins in order to use the ACMPHS0 analog input pin function and GPT output pin function.

[2] Set the operation of ACMPHS0 to output a level detection signal to the POEG in response to comparing the analog input voltage and reference input voltage.

[3] Write 0xA500 to the GTONCWPA register to enable writing to the GTONCCRA register.

[4] Select level detection for the comparator by setting the GTONCCRA.NFS[3:0] bits to 0x0. Set the GTONCCRA.NFV bit to 1 so that the request to stop output is sent when the selected source indicates 1.
Set the GTONCCRA.NE bit to 1 to enable the output of requests to stop output.

[5] Write 0xA501 to the GTONCWPA register to disable writing to the GTONCCRA register.

[6] Set the GPT for the output of a PWM waveform on its GTIOCn pin.

[7] Set the I/O port in order to use the required analog input pin of ACMPHS0 and output pin of the GPT.

The settings described above lead to the output of requests to stop output in response to comparison of the analog input voltage and reference input voltage while a PWM waveform is being output from the GPT pin, stopping output from the GPT pin.

Note 1. In this example, the comparator output is non-inverted.

Figure 19.5 Example of Settings to Stop Output on the GTIOC Pin in Response to Level Detection by a Comparator

19.4 Interrupt Sources

The POEG generates interrupts for the interrupt controller when any of the following is detected.

- Detection of level of input, indicated by the POEGGn.PIDF flag
- Detection of stopping of output from the GPT, indicated by the POEGGn.IOCF flag
- Detection of edges by the comparator, indicated by the POEGGn.IOCF flag

Table 19.3 shows interrupt sources and conditions.

Table 19.3 Interrupt sources and conditions (1 of 2)

Interrupt source	Symbol	Corresponding flag	Trigger condition
POEG Group A Interrupt	POEG_GROUPA	POEGGA.IOCF	Detection of stopping of output from the GPT
			Detection of edges by the comparator
		POEGGA.PIDF	Detection of the level input from the GTETRGA pin

Table 19.3 Interrupt sources and conditions (2 of 2)

Interrupt source	Symbol	Corresponding flag	Trigger condition
POEG Group B Interrupt	POEG_GROUPB	POEGGB.IOCF	Detection of stopping of output from the GPT
			Detection of edges by the comparator
		POEGGB.PIDF	Detection of the level input from the GTETRGB pin

19.5 External Trigger Output to GPT

Detection of the level input from the GTETRGn pin (n = A, B) can be monitored by the POEGGn.ST flag via active-sense selection and digital noise filtering. The GPT can function as follows in response to external trigger signals.

- Count start
- Count stop
- Counter clearing
- Up-counting
- Down-counting
- Input capture

For details of the above functions, see section 20, General PWM Timer (GPT).

19.6 Usage Notes

19.6.1 Transitions to Low-Power Modes

When the POEG is used, it should not be transitioned to Software Standby mode. In this mode, the stopping of output cannot be requested since the POEG is stopped.

19.6.2 Setting the Function for Stopping the Module

The module-stop control register can be set to enable or disable operation of the POEG. The POEG is stopped immediately after a reset. Registers of the POEG become accessible following release from the module-stopped state. For details, see section 10, Low Power Modes.

19.6.3 Duplication of Requests to Stop Output

While either the PIDF or IOCF flag in the POEGGn register is 1, cancellation of requests to stop by the detection signal set in the GTONCCRn register does not work due to the request to stop still being output because of the value of the flag. That is, note that requests to stop output will not be cancelled when a corresponding flag is set stop output in response to detection. Request signals to stop output in response to flag setting are obtained as the logical OR of the corresponding detection signals for stopping output.

20. General PWM Timer (GPT)

20.1 Overview

The General PWM Timer (GPT) is a 16-bit timer with GPT16 × 4 channels. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. In addition, PWM waveforms can be generated for controlling brushless DC motors. The GPT can also be used as a general-purpose timer.

This GPT provides the A/D conversion start request function, the asymmetric automatic dead time setting function for channels 0 to 2, the symmetric automatic dead time setting function for channel 3, and the enhanced interrupt skipping function. In addition, it is enhanced with the external pulse width measuring function for channel 3, the additional PWM modes for channels 0 to 2, the extended buffer transfer function, the GTCPPO pin output function, and the inter-channel cooperation function.

Table 20.1 lists the GPT specifications, Table 20.2 shows the GPT functions, and Figure 20.1 shows a block diagram.

Table 20.1 GPT specifications

Item	Description
Functions	• 16 bits × 4 channels (GPT16n (n = 0 to 3)) • Up-counting or down-counting (saw waves) or up/down-counting (triangle waves) for each counter • Clock sources independently selectable for each channel • Two input/output pins per channel • Two output compare/input capture registers per channel • For the two output compare/input capture registers of each channel, four registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use • In output compare operation, buffer switching can be at crests or troughs, enabling the generation of laterally asymmetric PWM waveforms • Registers for setting up frame cycles in each channel with capability for generating interrupts at overflow or underflow • Generation of dead times in PWM operation • Generation of high accuracy duty in the vicinity of duty 0% and 100% PWM waveform • In output compare operation, setting compare register is immediately used to generate PWM waveform with dead times • Synchronous starting, stopping and clearing counters for arbitrary channels • Count start, count stop, count clear, up-count, down-count, or input capture operation in response to a maximum of 8 ELC events • Count start, count stop, count clear, up-count, down-count, or input capture operation in response to the status of two input pins • Count start, count stop, count clear, up-count, down-count, or input capture operation in response to a maximum of 2 external triggers • Output pin disable function by dead time error and detected short-circuits between output pins • A/D conversion start request generation function • PWM waveform for controlling brushless DC motors can be generated • Compare match A to F event, overflow/underflow event, and input UVW edge event can be output to the ELC • Enables the noise filter for input capture and input UVW • Period count function • External pulse width measuring function • Logical operation between the channel output • Synchronous counter clearing/counter setting/input capture among channels • GTIOCnA/B input/output pin polarity reversal function • Bus clock: PCLKB, Core clock: GTCLK*1 • Frequency ratio: PCLKB:PCLKD = 1:N (N = 1/2/4/8/16/32/64) (when using synchronous clock)

Note 1. GPT core clock (GTCLK) is PCLKD.

Table 20.2 GPT functions (1 of 3)

Item	GPT0	GPT1	GPT2	GPT3
Timer counter	16-bit			

Table 20.2 GPT functions (2 of 3)

Item			GPT0	GPT1	GPT2	GPT3
Count clock			GTCLK GTCLK/2 GTCLK/4 GTCLK/8 GTCLK/16 GTCLK/32 GTCLK/64 GTCLK/256 GTCLK/1024 GTETRGA, GTETRGB			
Output compare/input capture registers (GTCCR)			GTCCRA GTCCRB			
Compare/buffer registers			GTCCRC GTCCRD GTCCRE GTCCRF			
Cycle setting register			GTPR			
Cycle setting buffer register			GTPBR GTPDBR			
I/O pins			GTIOCnA GTIOCnB (n = 0 to 3)			
External trigger input pin*1			GTETRGA GTETRGB			
Counter clear sources			GTPR register compare match Input capture Input pin status ELC event input External trigger input GTCCR register compare match Other channel's counter clear sources			
Period count function			Available			
Compare match output	Low output		✓			
	High output		✓			
	Toggle output		✓			
Input capture function			✓			
Automatic addition of dead time			✓			
PWM mode		Saw-wave PWM mode 1	✓	✓	✓	✓
		Saw-wave PWM mode 2	✓	✓	✓	—
		Saw-wave one-shot pulse mode	✓	✓	✓	✓
		Triangle-wave PWM mode 1/2/3	✓	✓	✓	✓
		Complementary PWM mode 1/2/3/4	✓	✓	✓	—
Phase counting function			—	—	—	✓
External pulse width measuring function			—	—	—	✓

Table 20.2 GPT functions (3 of 3)

Item		GPT0	GPT1	GPT2	GPT3
Buffer operation	Double buffer	✓			
	Simultaneous operation disable control for multiple channels	✓			
	Buffer transfer by counter clear/ compare match	✓			
One-shot operation		✓			
DTC activation		All the interrupt sources			
A/D conversion start trigger		Compare match of GTADTRA or GTADTRB register			
Brushless DC motor control function		✓			
Interrupt sources		11 sources ● GTCCRA compare match/input capture (GPTn_CCMPA) ● GTCCRB compare match/input capture (GPTn_CCMPB) ● GTCCRC compare match (GPTn_CMPC) ● GTCCRD compare match (GPTn_CMPD) ● GTCCRE compare match (GPTn_CMPE) ● GTCCRF compare match (GPTn_CMPF) ● GTADTRA compare match (GPTn_ADTRGA) ● GTADTRB compare match (GPTn_ADTRGB) ● GTCNT overflow (GTPR compare match) (GPTn_OVF) ● GTCNT underflow (GPTn_UDF) ● GTPC count stop (GPTn_PC)			
Interrupt skipping function		● Skipping of interrupts of GTCNT counter overflow (GTPR register compare match) (GTPn_OVF) and GTCNT counter underflow (GTPn_UDF) (interlocked with other interrupts and A/D conversion start requests) ● Skipping of GTADTRA and GTADTRB register compare match ● Buffer operation skipping function			
Event linking (ELC) function		✓*2			
Noise filtering function		✓			
Synchronous counter clearing/counter setting/input capture		✓			

Note: ✓: Possible
Note 1. GTETRGn connects to GPT through the POEG module. Therefore, to use the GPT function, supply the POEG clock by clearing the MSTPCRD.MSTPDn (n = 13 to 14) bit.
Note 2. See section 20.6. Operations Linked by ELC.

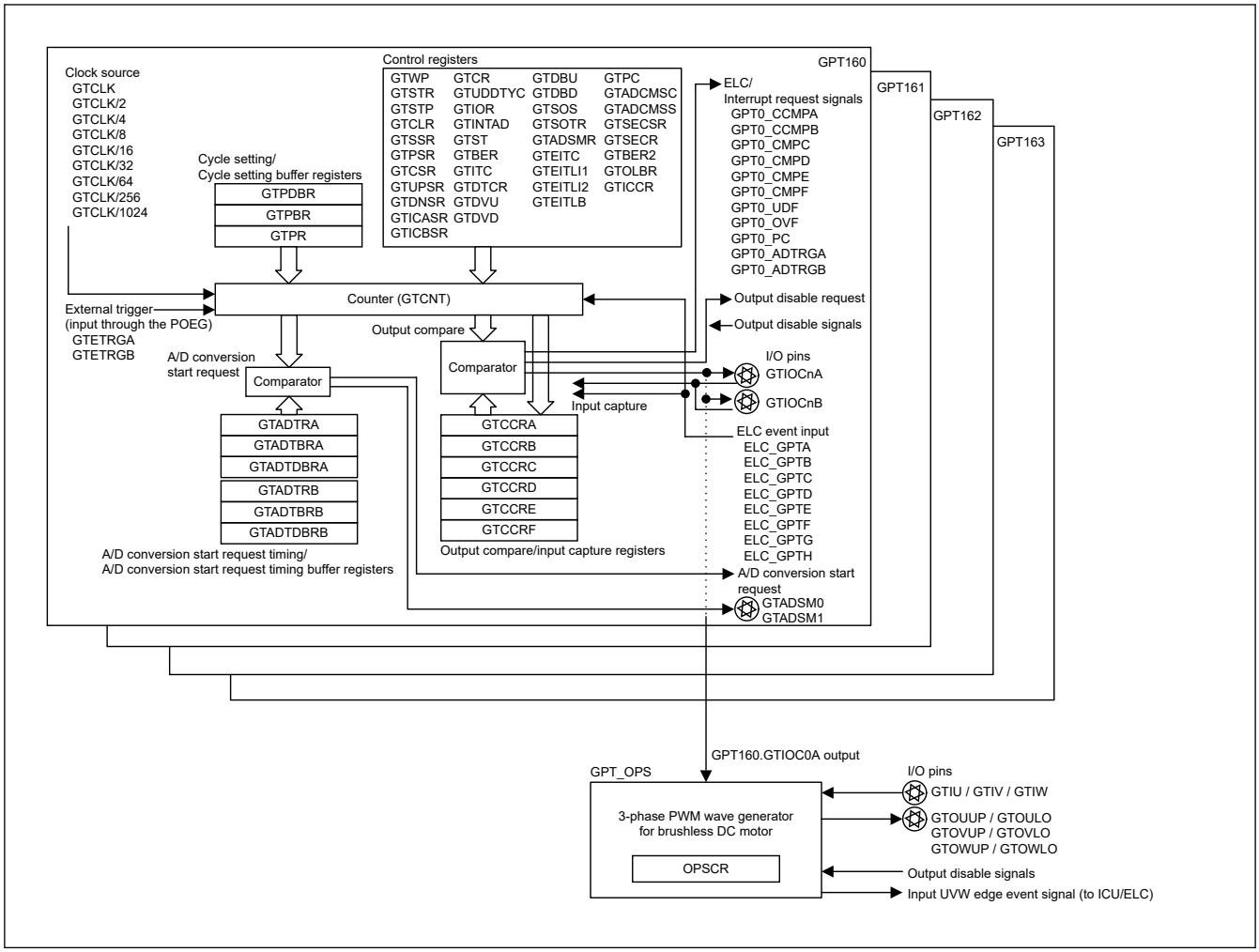


Figure 20.1 GPT block diagram (Saw-wave PWM mode 1, Saw-wave one-shot pulse mode, Triangle-wave PWM mode 1, 2, 3)

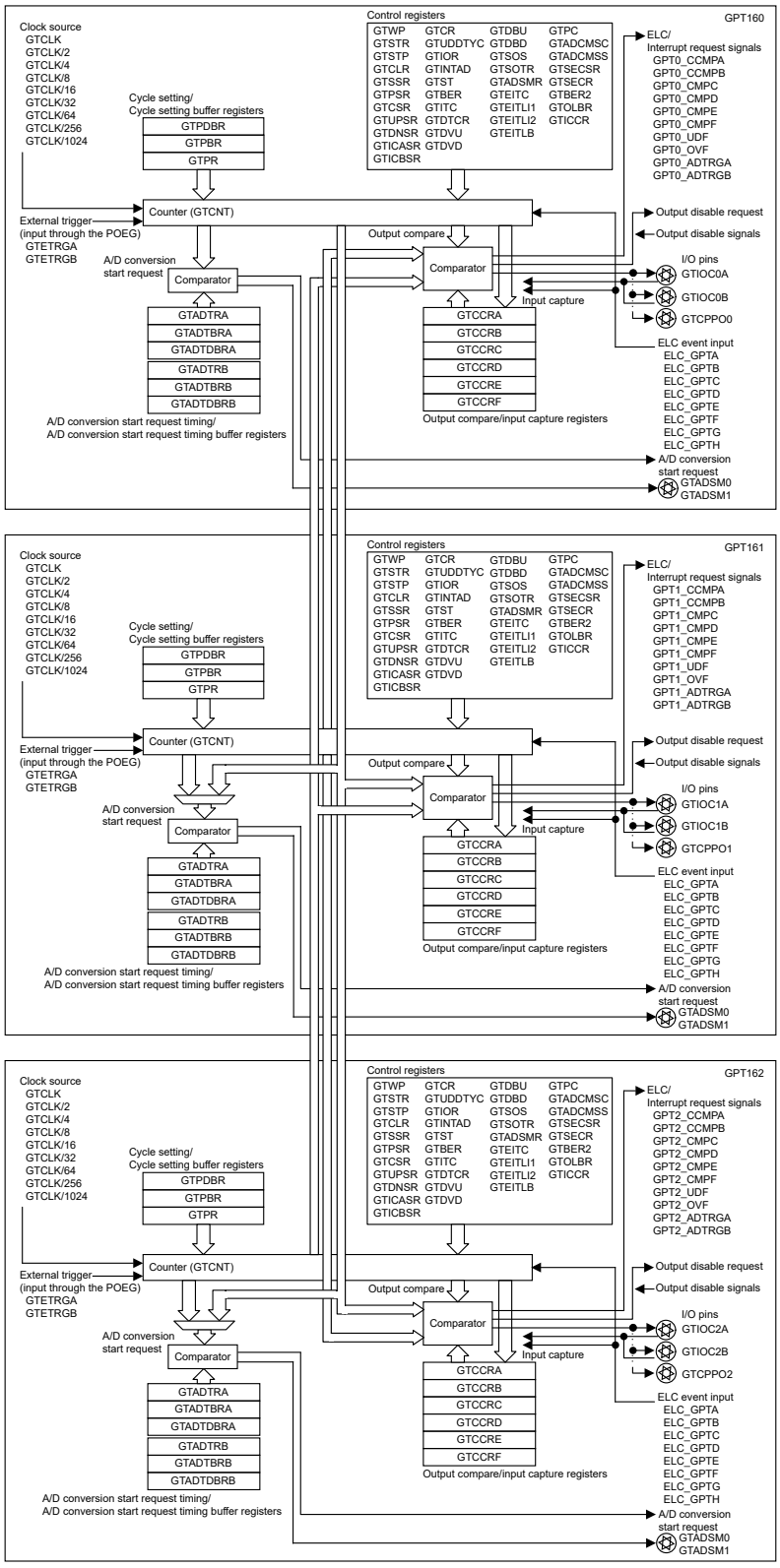


Figure 20.2 GPT block diagram (Saw-wave PWM mode 1, 2, Saw-wave one-shot pulse mode, Triangle-wave PWM mode 1, 2, 3, Complementary PWM mode 1, 2, 3, 4)

In this specification, three consecutive channels to configure complementary PWM mode is defined as complementary PWM mode channel group. The lowest position channel of complementary PWM mode channel group is defined as master channel. The second channel is defined as slave channel 1. The highest position channel is defined as slave channel 2.

Figure 20.3 shows an example using multiple GPTs.

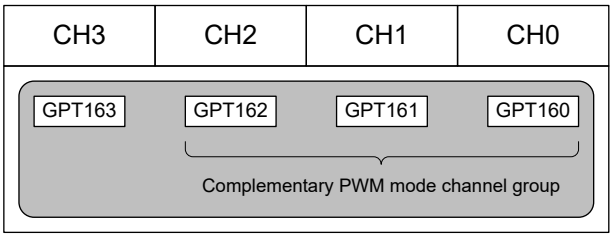


Figure 20.3 Association between GPT channels and module names

Table 20.3 lists the I/O pins.

Table 20.3 GPT I/O pins

Channel	Pin name	I/O	Function
Common	GTETRGx	Input	External trigger input pin x (input through the POEG)
	GTADSM0	Output	A/D conversion start request monitor 0 output pin
	GTADSM1	Output	A/D conversion start request monitor 1 output pin
GPT16n	GTIOCnA	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOCnB	I/O	GTCCRB register input capture input/output compare output/PWM output pin
	GTCPPOk	Output	Toggle output synchronized with PWM period
GPT_OPS	GTIU	Input	Hall sensor input pin U
	GTIV	Input	Hall sensor input pin V
	GTIW	Input	Hall sensor input pin W
	GTOUUP	Output	3-phase PWM output for BLDC motor control (positive U-phase)
	GTOULO	Output	3-phase PWM output for BLDC motor control (negative U-phase)
	GTOVUP	Output	3-phase PWM output for BLDC motor control (positive V-phase)
	GTOVLO	Output	3-phase PWM output for BLDC motor control (negative V-phase)
	GTOWUP	Output	3-phase PWM output for BLDC motor control (positive W-phase)
	GTOWLO	Output	3-phase PWM output for BLDC motor control (negative W-phase)

Note: x: A, B
n: 0 to 3
k: 0 to 3

20.2 Register Descriptions

20.2.1 GTWP : General PWM Timer Write-Protection Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x00

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]								—	—	—	CMNWP	CLRWP	STPWP	STRWP	WP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WP	Register Write Disable 0: Write to the register enabled 1: Write to the register disabled	R/W
1	STRWP	GTSTR.CSTRT Bit Write Disable 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
2	STPWP	GTSTP.CSTOP Bit Write Disable 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
3	CLRWP	GTCLR.CCLR Bit Write Disable 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
4	CMNWP	Common Register Write Disabled 0: Write to the register is enabled 1: Write to the register is disabled	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	GTWP Key Code When 0xA5 is written to these bits, writing to the WP, STRWP, STPWP, CLRWP, and CMNWP bits are permitted. These bits are read as 0.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

GTWP enables or disables writing to registers to prevent accidental modification. Protection by the GTWP register is only for the writes by the CPU. GTWP does not protect registers from updates that occur in association with CPU writes.

WP bit (Register Write Disable)

The following is a list of write enabled or disabled registers:

GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTITC, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTPDBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTDVD, GTDBU, GTDBD, GTSOS, GTSOTR, GTADSMR, GTEITC, GTEITLI1, GTEITLI2, GTEITLB, GTPC, GTADCMSC, GTADCMSS, GTBER2, GTOLBR, GTICCR.

STRWP bit (GTSTR.CSTRT Bit Write Disable)

The STRWP bit enables or disables starting the updating of counter values by writing to the CSTRTn bit (n = 0 to 3) corresponding to a channel number in the GTSTR register.

The bit position of each CSTRTn bit in the GTSTR register is allocated to the channel with the corresponding number, and writing to the GTSTR register for any channel results in writing to the registers of all channels. The STRWP bit for

each channel does not control writing but only controls updating of the CSTRT bit for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the CSTRT bits of a channel for which the setting of the STRWP bit is 1 (disabling writing), the CSTRT bit for the given channel is not updated, but the CSTRT bits corresponding to channel for which the setting of the STRWP bit is 0 (enabling writing) are updated. For example, when the setting of the GPT160.GTWP.STRWP bit is 0 (enabling writing), writing 1 to the GPT161.GTSTR.CSTRT0 bit when its current setting is 0 causes the value to be updated, and the GPT160.GTCNT counter starts to run. When the setting of the GPT160.GTWP.STRWP bit is 1 (disabling writing), writing 1 to the GPT161.GTSTR.CSTRT0 bit when its current setting is 0 leaves the bit with the value 0, and the GPT160.GTCNT counter does not run.

If you want to protect all bits in the GTSTR register from being updated, set the STRWP bits of all channels to 1.

STPWP bit (GTSTP.CSTOP Bit Write Disable)

The STPWP bit enables or disables starting the updating of counter values by writing to the CSTOPn bit (n = 0 to 3) corresponding to a channel number in the GTSTP register.

The bit position of each CSTOPn bit in the GTSTP registers is allocated to the channel with the corresponding number, and the writing to the GTSTP register for any channel results in writing to the registers of all channels. The STPWP bit for each channel does not control writing but only controls updating of the CSTOP bit for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the CSTOP bits of a channel for which the setting of the STPWP bit is 1 (disabling writing), the CSTOP bit for the given channel is not updated, but the CSTOP bits corresponding to channel for which the setting of the STPWP bit is 0 (enabling writing) are updated. For example, when the setting of the GPT160.GTWP.STPWP bit is 0 (enabling writing), writing 1 to the GPT161.GTSTP.CSTOP0 bit when its current setting is 0 causes the value to be updated, and the GPT160.GTCNT counter is stopped. When the setting of the GPT160.GTWP.STPWP bit is 1 (disabling writing), writing 1 to the GPT161.GTSTP.CSTOP0 bit when its current setting is 0 leaves the bit with the value 0, and the GPT160.GTCNT counter is not stopped.

If you want to protect all bits in the GTSTP register from being updated, set the STPWP bits of all channels to 1.

CLRWP bit (GTCLR.CCLR Bit Write Disable)

The CLRWP bit enables or disables starting the updating of counter values by writing to the CCLRn bit (n = 0 to 3) corresponding to a channel number in the GTCLR register.

The bit position of each CCLRn bit in the GTCLR registers is allocated to the channel with the corresponding number, and the writing to the GTCLR register for any channel results in writing to the registers of all channels. The CLRWP bit for each channel does not control writing but only controls updating of the CCLR bit for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the CCLR bits of a channel for which the setting of the CLRWP bit is 1 (disabling writing), the CCLR bit for the given channel is not updated, but the CCLR bits corresponding to channel for which the setting of the CLRWP bit is 0 (enabling writing) are updated. For example, when the setting of the GPT160.GTWP.CLRWP bit is 0 (enabling writing), writing 1 to the GPT161.GTCLR.CCLR0 bit when its current setting is 0 causes the value to be updated, and the GPT160.GTCNT counter is cleared. When the setting of the GPT160.GTWP.CLRWP bit is 1 (disabling writing), writing 1 to the GPT161.GTCLR.CCLR0 bit when its current setting is 0 leaves the bit with the value 0, and the GPT160.GTCNT counter is not cleared.

If you want to protect all bits in the GTCLR register from being updated, set the CLRWP bits of all channels to 1.

CMNWP bit (Common Register Write Disabled)

The CMNWP bit enables or disables starting the updating of counter values by writing to the SECSELn bit (n = 0 to 3) corresponding to a channel number in the GTSECSR register or to the GTSECR register.

The bit position of each SECSEL bit in the GTSECSR registers is allocated to the channel with the corresponding number, and the writing to the GTSECSR register for any channel results in writing to the registers of all channels. Writing to the GTSECR register of any channel leads to writing to the registers of all channels. The CMNWP bit for each channel does not control writing but only controls updating of the SECSEL bit and the GTSECR register value for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the SECSEL bit and the GTSECR register value of a channel for which the setting of the CMNWP bit is 1 (disabling writing), the SECSEL bit and the GTSECR register value for the given channel is not updated,

but the SECSEL bit and the GTSECR register value corresponding to channel for which the setting of the CMNWP bit is 0 (enabling writing) are updated.

For example, when the setting of the GPT160.GTWP.CMNWP bit is 0 (enabling writing), writing to the GPT161.GTSECSR.SECSEL0 bit causes the value of the GPT160.GTSECSR.SECSEL0 bit to be updated. In the same way, writing to the GPT161.GTSECR register updates the value of the GPT160.GTSECR register. When the setting of the GPT160.GTWP.CMNWP bit is 1 (disabling writing), writing to the GPT161.GTSECSR.SECSEL0 bit does not cause the value of the GPT160.GTSECSR.SECSEL0 bit to be updated. In the same way, writing to the GPT161.GTSECR register does not update the value of the GPT160.GTSECR register.

If you want to protect all bits in the GTSECSR and GTSECR registers from being updated, set the CMNWP bits of all channels to 1.

PRKEY[7:0] bits (GTWP Key Code)

The PRKEY[7:0] bits control whether the WP, STRWP, STPWP, CLRWP, and CMNWP bits can be overwritten.

20.2.2 GTSTR : General PWM Timer Software Start Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x04

Bit position:	31																												0
Bit field:	CSTRT31 to CSTRT0																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CSTRT0 to CSTRT31*1	Channel n GTCNT Count Start (n is the same as the bit position value) 0: GTCNT counter is not started 1: GTCNT counter is started	R/W

Note 1. The bits that can be used vary depending on the product. The n in CSTRTn is the same as the GPT channel number. For this product, n is 0 to 3.

The GTSTR starts the GTCNT counter operation for each channel n, where n = 0 to 3.

The GTSTR bit number represents the channel number. The GTSTR register of each channel is shared by all of the channels. The GTCNT counter starts for the channel associated with the GTSTR bit number where 1 is written. Writing 0 has no effect on the status of GTCNT counter and the value of GTSTR register.

In complementary PWM mode, writing to the CSTRTn bit of master channel is only valid. The bit on slave channels reflects the bit value of master channel.

For the association between module names and channel numbers, see Figure 20.3.

CSTRTn bits (Channel n GTCNT Count Start (n = 0 to 3))

The CSTRTn bits start channel n of the GTCNT counter operation. Writing to the GTSTR.CSTRTn bit (n = 0 to 3) has no effect unless the GTSSR.CSTRT bit is set to 1.

The read data shows the counter status of each channel (GTCR.CST bit). A value of 0 means the counter is stopped and 1 means the counter is running.

20.2.3 GTSTP : General PWM Timer Software Stop Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x08

Bit position:	31																												0
Bit field:	CSTOP31 to CSTOP0																												
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	CSTOP0 to CSTOP31*1	Channel n GTCNT Count Stop (n is the same as the bit position value) 0: GTCNT counter is not stopped 1: GTCNT counter stopped	R/W

Note 1. The bits that can be used vary depending on the product. The n in CSTOPn is the same as the GPT channel number. For this product, n is 0 to 3.

The GTSTP stops the GTCNT counter operation for each channel n, where n = 0 to 3.

The GTSTP bit number represents the channel number. The GTSTP register of each channel is shared by all the channels. The GTCNT counter stops for the channel associated with the GTSTP bit number where 1 is written. Writing 0 has no effect on the status of the GTCNT counter and the value of GTSTP register.

In complementary PWM mode, writing to the CSTOPn bit of master channel is only valid. The bit on slave channels reflects the bit value of master channel.

For the association between module names and channel numbers, see Figure 20.3.

CSTOPn bits (Channel n GTCNT Count Stop (n = 0 to 3))

The CSTOPn bits stop channel n of the GTCNT counter operation. Writing to the GTSTP.CSTOPn bit (n = 0 to 3) has no effect unless the GTPSR.CSTOP bit is set to 1. The read data shows the counter status of each channel (invert of GTCR.CST bit). A value of 0 means the counter is running and 1 means the counter is stopped.

20.2.4 GTCLR : General PWM Timer Software Clear Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x0C

Bit position:	31																												0
Bit field:	CCLR31 to CCLR0																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CCLR0 to CCLR31*1	Channel n GTCNT Count Clear (n : the same as bit position value) 0: GTCNT counter is not cleared 1: GTCNT counter is cleared	W

Note 1. The bits that can be used vary depending on the product. The n of CCLRn is the same as the GPT channel number. For this product, n is 0 to 3.

The GTCLR is a write-only register that clears the GTCNT counter operation for each channel n, where n = 0 to 3.

The GTCLR bit number represents the channel number. The GTCLR register of each channel is shared by all the channels. The GTCNT counter is cleared for the channel associated with the GTCLR bit number where 1 is written. Writing 0 has no effect on the status of GTCNT counter.

In complementary PWM mode, writing to the CCLRn bit of master channel is only valid. The bit on slave channels reflects the bit value of master channel.

For the association between module names and channel numbers, see Figure 20.3.

CCLRn bits (Channel n GTCNT Count Clear (n = 0 to 3))

When the counting direction flag is set for decrement (GTST.TUCF flag = 0) with saw-wave mode selected in the GTCR.MD[3:0] bits, the value of the GTCNT counter becomes that of the corresponding GTPR register in response to writing 1 to the CCLRn bit. The value of the counter becomes 0x0000 0000 with other settings. These bits are read as 0.

20.2.5 GTSSR : General PWM Timer Start Source Select Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CSTRT	—	—	—	—	—	—	—	SSELCH	SSELCG	SSELCF	SSELCE	SSELCD	SSELCC	SSELCB	SSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SSCBFAH	SSCBFAL	SSCBRAH	SSCBRAL	SSCAFBH	SSCAFBL	SSCARBH	SSCARBL	—	—	—	—	SSGTRGBF	SSGTRGBR	SSGTRGAF	SSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SSGTRGAR	GTETRGA Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGA input 1: Counter start enabled on the rising edge of GTETRGA input	R/W*1
1	SSGTRGAF	GTETRGA Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGA input 1: Counter start enabled on the falling edge of GTETRGA input	R/W*1
2	SSGTRGBR	GTETRGB Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGB input 1: Counter start enabled on the rising edge of GTETRGB input	R/W*1
3	SSGTRGBF	GTETRGB Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGB input 1: Counter start enabled on the falling edge of GTETRGB input	R/W*1
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	SSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter start enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	SSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter start enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	SSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter start enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	SSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter start enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W

Bit	Symbol	Function	R/W
12	SSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter start enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	SSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter start enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	SSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter start enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	SSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter start enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	SSELCA	ELC_GPTA Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTA input 1: Counter start enabled at the ELC_GPTA input	R/W*1
17	SSELCB	ELC_GPTB Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTB input 1: Counter start enabled at the ELC_GPTB input	R/W*1
18	SSELCC	ELC_GPTC Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTC input 1: Counter start enabled at the ELC_GPTC input	R/W*1
19	SSELCD	ELC_GPTD Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTD input 1: Counter start enabled at the ELC_GPTD input	R/W*1
20	SSELCE	ELC_GPTE Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTE input 1: Counter start enabled at the ELC_GPTE input	R/W*1
21	SSELCF	ELC_GPTF Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTF input 1: Counter start enabled at the ELC_GPTF input	R/W*1
22	SSELCG	ELC_GPTG Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTG input 1: Counter start enabled at the ELC_GPTG input	R/W*1
23	SSELCH	ELC_GPTH Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTH input 1: Counter start enabled at the ELC_GPTH input	R/W*1
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CSTRT	Software Source Counter Start Enable 0: Counter start disabled by the GTSTR register 1: Counter start enabled by the GTSTR register	R/W*1

Note 1. In complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTSSR sets the source to start the GTCNT counter.

Input from GTETRGn (n = A, B) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

SSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Start Enable)

The SSGTRGAR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGA pin input.

SSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Start Enable)

The SSGTRGAF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGA pin input.

SSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Start Enable)

The SSGTRGBR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGB pin input.

SSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Start Enable)

The SSGTRGBF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGB pin input.

SSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Start Enable)

The SSCARBL bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

SSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Start Enable)

The SSCARBH bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

SSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Start Enable)

The SSCAFBL bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

SSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Start Enable)

The SSCAFBH bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

SSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Start Enable)

The SSCBRAL bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

SSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Start Enable)

The SSCBRAH bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

SSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Start Enable)

The SSCBFAL bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

SSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Start Enable)

The SSCBFAH bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

SSELCm bit (ELC_GPTm Event Source Counter Start Enable) (m = A to H)

The SSELCm bit enables or disables the GTCNT counter start at the ELC_GPTm event input.

CSTRT bit (Software Source Counter Start Enable)

The CSTRT bit enables or disables the GTCNT counter start by GTSTR register.

20.2.6 GTPSR : General PWM Timer Stop Source Select Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CSTOP	—	—	—	—	—	—	—	PSELCH	PSELCG	PSELCF	PSELCE	PSELCD	PSELCC	PSELCB	PSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PSCBFAH	PSCBFAL	PSCBRAH	PSCBRAL	PSCAFBH	PSCAFBL	PSCARBH	PSCARBL	—	—	—	—	PSGTRGBF	PSGTRGBR	PSGTRGAF	PSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PSGTRGAR	GTETRGA Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGA input 1: Counter stop enabled on the rising edge of GTETRGA input	R/W*1
1	PSGTRGAF	GTETRGA Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGA input 1: Counter stop enabled on the falling edge of GTETRGA input	R/W*1
2	PSGTRGBR	GTETRGB Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGB input 1: Counter stop enabled on the rising edge of GTETRGB input	R/W*1
3	PSGTRGBF	GTETRGB Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGB input 1: Counter stop enabled on the falling edge of GTETRGB input	R/W*1
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	PSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter stop enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	PSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter stop enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	PSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter stop enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	PSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter stop enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	PSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter stop enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W

Bit	Symbol	Function	R/W
13	PSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter stop enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	PSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter stop enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	PSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter stop enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	PSELCA	ELC_GPTA Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTA input 1: Counter stop enabled at the ELC_GPTA input	R/W*1
17	PSELCB	ELC_GPTB Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTB input 1: Counter stop enabled at the ELC_GPTB input	R/W*1
18	PSELCC	ELC_GPTC Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTC input 1: Counter stop enabled at the ELC_GPTC input	R/W*1
19	PSELCD	ELC_GPTD Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTD input 1: Counter stop enabled at the ELC_GPTD input	R/W*1
20	PSELCE	ELC_GPTE Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTE input 1: Counter stop enabled at the ELC_GPTE input	R/W*1
21	PSELCF	ELC_GPTF Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTF input 1: Counter stop enabled at the ELC_GPTF input	R/W*1
22	PSELCG	ELC_GPTG Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTG input 1: Counter stop enabled at the ELC_GPTG input	R/W*1
23	PSELCH	ELC_GPTH Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTH input 1: Counter stop enabled at the ELC_GPTH input	R/W*1
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CSTOP	Software Source Counter Stop Enable 0: Counter stop disabled by the GTSTP register 1: Counter stop enabled by the GTSTP register	R/W*1

Note 1. In complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.

The GTPSR sets the source to stop the GTCNT counter.

Inputs from GTETRGn (n = A, B) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

PSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Stop Enable)

The PSGTRGAR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGA pin input.

PSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Stop Enable)

The PSGTRGAF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGA pin input.

PSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Stop Enable)

The PSGTRGBR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGB pin input.

PSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Stop Enable)

The PSGTRGBF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGB pin input.

PSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Stop Enable)

The PSCARBL bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

PSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Stop Enable)

The PSCARBH bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

PSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Stop Enable)

The PSCAFBL bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

PSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Stop Enable)

The PSCAFBH bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

PSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Stop Enable)

The PSCBRAL bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

PSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Stop Enable)

The PSCBRAH bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

PSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Stop Enable)

The PSCBFAL bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

PSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Stop Enable)

The PSCBFAH bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

PSELCm bit (ELCm Event Source Counter Stop Enable) (m = A to H)

The PSELCm bit enables or disables the GTCNT counter stop at the ELC_GPTm event input.

CSTOP bit (Software Source Counter Stop Enable)

The CSTOP bit enables or disables the GTCNT counter stop by the GTSTP register.

20.2.7 GTCSR : General PWM Timer Clear Source Select Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CCLR	—	—	—	CP1CCE	CSCMSC[2:0]			CSELCH	CSELCG	CSELCF	CSELCE	CSELCD	CSELCC	CSELCB	CSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CSCBFAH	CSCBFAL	CSCBRAH	CSCBRAL	CSCAFBH	CSCAFBL	CSCARBH	CSCARBL	—	—	—	—	CSGTRGBF	CSGTRGBR	CSGTRGAF	CSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CSGTRGAR	GTETRGA Pin Rising Input Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTETRGA input 1: Counter clear enabled on the rising edge of GTETRGA input	R/W*1
1	CSGTRGAF	GTETRGA Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGA input 1: Counter clear enabled on the falling edge of GTETRGA input	R/W*1
2	CSGTRGBR	GTETRGB Pin Rising Input Source Counter Clear Enable 0: Disable counter clear on the rising edge of GTETRGB input 1: Enable counter clear on the rising edge of GTETRGB input	R/W*1
3	CSGTRGBF	GTETRGB Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGB input 1: Counter clear enabled on the falling edge of GTETRGB input	R/W*1
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	CSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter clear enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	CSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter clear enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	CSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter clear enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	CSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter clear enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	CSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter clear enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W

Bit	Symbol	Function	R/W
13	CSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter clear enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	CSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter clear enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	CSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter clear enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	CSELCA	ELC_GPTA Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTA input 1: Counter clear enabled at the ELC_GPTA input	R/W*1
17	CSELCB	ELC_GPTB Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTB input 1: Counter clear enabled at the ELC_GPTB input	R/W*1
18	CSELCC	ELC_GPTC Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTC input 1: Counter clear enabled at the ELC_GPTC input	R/W*1
19	CSELCD	ELC_GPTD Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTD input 1: Counter clear enabled at the ELC_GPTD input	R/W*1
20	CSELCE	ELC_GPTE Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTE input 1: Counter clear enabled at the ELC_GPTE input	R/W*1
21	CSELCF	ELC_GPTF Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTF input 1: Counter clear enabled at the ELC_GPTF input	R/W*1
22	CSELCG	ELC_GPTG Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTG input 1: Counter clear enabled at the ELC_GPTG input	R/W*1
23	CSELCH	ELC_GPTH Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTH input 1: Counter clear enabled at the ELC_GPTH input	R/W*1
26:24	CSCMSC[2:0]	Compare Match/Input Capture/Synchronous Counter Clearing Source Counter Clear Enable 0 0 0: Counter clear disabled by Compare match/ Input capture/ Synchronous counter clearing group 0 0 1: Counter clear enabled at the GTCCRA register compare match/ Input capture 0 1 0: Counter clear enabled at the GTCCRB register compare match/ Input capture 0 1 1: Counter clear enabled at the GTCCRC register compare match 1 0 0: Counter clear enabled at the GTCCRD register compare match 1 0 1: Counter clear enabled at the GTCCRE register compare match 1 1 0: Counter clear enabled at the GTCCRF register compare match 1 1 1: Counter clear enabled at the synchronous counter clearing group	R/W
27	CP1CCE	Complementary PWM Mode 1 Crest Source Counter Clear Enable*2 0: Counter clear disabled at the crest of complementary PWM mode 1 1: Counter clear enabled at the crest of complementary PWM mode 1	R/W*1
30:28	—	These bits are read as 0. The write value should be 0.	R/W
31	CCLR	Software Source Counter Clear Enable 0: Counter clear disabled by the GTCLR register 1: Counter clear enabled by the GTCLR register	R/W*1

Note 1. In complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.
Note 2. This bit is only available in GPT160 to GPT162.
In GPT163, this bit is read as 0. The write value should be 0.

The GTCSR sets the source to clear the GTCNT counter.

Counter clearing can be executed whether the counter is running (GTCR.CST = 1) or stopped (GTCR.CST = 0).

Inputs from GTETRGn (n = A, B) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

CSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Clear Enable)

The CSGTRGAR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGA pin input.

CSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Clear Enable)

The CSGTRGAF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGA pin input.

CSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Clear Enable)

The CSGTRGBR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGB pin input.

CSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Clear Enable)

The CSGTRGBF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGB pin input.

CSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Clear Enable)

The CSCARBL bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

CSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Clear Enable)

The CSCARBH bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

CSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Clear Enable)

The CSCAFBL bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

In complementary PWM mode, this setting is invalid.

CSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Clear Enable)

The CSCAFBH bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

In complementary PWM mode, this setting is invalid.

CSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Clear Enable)

The CSCBRAL bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

CSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Clear Enable)

The CSCBRAH bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

CSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Clear Enable)

The CSCBFAL bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

In complementary PWM mode, this setting is invalid.

CSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Clear Enable)

The CSCBFAH bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

In complementary PWM mode, this setting is invalid.

CSELCm bit (ELCm Event Source Counter Clear Enable) (m = A to H)

The CSELCm bit enables or disables the GTCNT counter clear at the ELC_GPTm event input.

CSCMSC[2:0] bits (Compare Match/Input Capture/Synchronous Counter Clearing Source Counter Clear Enable)

The CSCMSC[2:0] bits enable or disable the counter clear of the GTCNT counter by compare match/input capture/ synchronous counter clearing group.

If counter clearing by compare match/input capture is enabled, it can be handled as synchronous clearing by inter channel cooperation which is written in section 20.3.8.3. Synchronous Clear Operation by Inter Channel Cooperation.

If this bit is set to 001b or 010b and counter clearing by input capture (excluding compare match) is allowed, set the same factor as the input capture factor selected in GTICmSR (m = A, B) register to the counter clearing factor in GTCSR register.

In addition, when the timer prescaler is not used (when GTCR.TPCS[3:0] = 0000b), the input capture by other channel factors (set GTICASR.ASOC or GTICBSR.BSOC to 1) can be used as a counter clear factor. The input capture by other channel factors cannot be used as the synchronous clearing factor for other channels but can be used as a counter clearing factor for its own channel. In this case, GTCSR configuration is not required.

Since the compare match by the register that is performing the buffer operation (including the wave mode specific case) does not occur, the counter clear enable setting that makes the target register of the buffer operation the compare match factor is invalid.

In complementary PWM mode, the counter clear enable setting for compare match of the GTCCRB register, GTCCRE register, and GTCCRF register is invalid even when the buffer operation is not performed.

CP1CCE bit (Complementary PWM Mode 1 Crest Source Counter Clear Enable)

The CP1CCE bit enables or disables the counter clear at the crest of complementary PWM mode 1.

To enable this bit, do not set 1 to the PSYE bit of the GTIOR register.

It is valid only for the master channel in complementary PWM mode. The master channel setting also clears the GTCNT counter of the slave channel in complementary PWM mode.

CCLR bit (Software Source Counter Clear Enable)

The CCLR bit enables or disables the GTCNT counter clear by the GTCLR register.

20.2.8 GTUPSR : General PWM Timer Up Count Source Select Register

Base address: GPT163 = 0x4008_9300

Offset address: 0x1C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	USILVL[3:0]				USELCH	USELCG	USELCF	USELCE	USELCD	USELCC	USELCB	USELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	USCBFAH	USCBFAL	USCBRAH	USCBRAL	USCAFBH	USCAFBL	USCARBH	USCARBL	—	—	—	—	USGTRGBF	USGTRGBR	USGTRGAF	USGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	USGTRGAR	GTETRGA Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGA input 1: Counter count up enabled on the rising edge of GTETRGA input	R/W
1	USGTRGAF	GTETRGA Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGA input 1: Counter count up enabled on the falling edge of GTETRGA input	R/W
2	USGTRGBR	GTETRGB Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGB input 1: Counter count up enabled on the rising edge of GTETRGB input	R/W
3	USGTRGBF	GTETRGB Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGB input 1: Counter count up enabled on the falling edge of GTETRGB input	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	USCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count up enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	USCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count up enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	USCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count up enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	USCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count up enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	USCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count up enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	USCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count up enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	USCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count up enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	USCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count up enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	USELCA	ELC_GPTA Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTA input 1: Counter count up enabled at the ELC_GPTA input	R/W
17	USELCB	ELC_GPTB Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTB input 1: Counter count up enabled at the ELC_GPTB input	R/W

Bit	Symbol	Function	R/W
18	USELCC	ELC_GPTC Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTC input 1: Counter count up enabled at the ELC_GPTC input	R/W
19	USELCD	ELC_GPTD Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTD input 1: Counter count up enabled at the ELC_GPTD input	R/W
20	USELCE	ELC_GPTE Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTE input 1: Counter count up enabled at the ELC_GPTE input	R/W
21	USELCF	ELC_GPTF Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTF input 1: Counter count up enabled at the ELC_GPTF input	R/W
22	USELCG	ELC_GPTG Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTG input 1: Counter count up enabled at the ELC_GPTG input	R/W
23	USELCH	ELC_GPTH Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTH input 1: Counter count up enabled at the ELC_GPTH input	R/W
27:24	USILVL[3:0]	External Input Level Source Count-Up Enable 0 0 0 0: Disables count-up by external input level 0 0 1 0: Enables count-up by GTIOCnA pin input level 0 0 0 1 1: Enables count-up by GTIOCnA pin input level 1 0 1 0 0: Enables count-up by GTIOCnB pin input level 0 0 1 0 1: Enables count-up by GTIOCnB pin input level 1 1 0 0 0: Enables count-up by GTETRGA pin input level 0 1 0 0 1: Enables count-up by GTETRGA pin input level 1 1 0 1 0: Enables count-up by GTETRGB pin input level 0 1 0 1 1: Enables count-up by GTETRGB pin input level 1 Others: Setting prohibited	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The GTUPSR sets the source to count up the GTCNT counter.

When at least one bit in the GTUPSR register is set to 1, the GTCNT counter is counted up by the source that is set to 1 in this register. In this case, GTCR.TPCS has no effect.

Number of increment in counting is one even when multiple sources are generated simultaneously.

Inputs from GTETRGn (n = A, B) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

USGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Up Enable)

The USGTRGAR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGA pin input.

USGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Up Enable)

The USGTRGAF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGA pin input.

USGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Up Enable)

The USGTRGBR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGB pin input.

USGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Up Enable)

The USGTRGBF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGB pin input.

USCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Up Enable)

The USCARBL bit enables or disables GTCNT counter count up on the rising edge of GTIOCnA pin input, when GTIOCnB input is 0.

USCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Up Enable)

The USCARBH bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

USCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Up Enable)

The USCAFBL bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

USCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Up Enable)

The USCAFBH bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

USCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Up Enable)

The USCBRAL bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

USCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Up Enable)

The USCBRAH bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnB pin input, when the GTIOCnA input is 1.

USCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Up Enable)

The USCBFAL bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnB pin input, when the GTIOCnA input is 0.

USCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Up Enable)

The USCBFAH bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnB pin input, when the GTIOCnA input is 1.

USELCm bit (ELC_GPTm Event Source Counter Count Up Enable) (m = A to H)

The USELCm bit enables or disables the GTCNT counter count up at the ELC_GPTm event input.

USILVL[3:0] bits (External Input Level Source Count-Up Enable)

The USILVL[3:0] bits enable or disable the count-up of the GTCNT counter by the GTIOCnA pin input level, GTIOCnB pin input level, and GTETRGA/GTETRGB input level.

20.2.9 GTDNSR : General PWM Timer Down Count Source Select Register

Base address: GPT163 = 0x4008_9300

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	DSILVL[3:0]				DSELCH	DSELCG	DSELCF	DSELCE	DSELCD	DSELCC	DSELCB	DSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DSCBFAH	DSCBFAL	DSCBRAH	DSCBRAL	DSCAFBH	DSCAFBL	DSCARBH	DSCARBL	—	—	—	—	DSGTRGBF	DSGTRGBR	DSGTRGAF	DSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DSGTRGAR	GTETRGA Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGA input 1: Counter count down enabled on the rising edge of GTETRGA input	R/W

Bit	Symbol	Function	R/W
1	DSGTRGAF	GTETRGA Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGA input 1: Counter count down enabled on the falling edge of GTETRGA input	R/W
2	DSGTRGBR	GTETRGB Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGB input 1: Counter count down enabled on the rising edge of GTETRGB input	R/W
3	DSGTRGBF	GTETRGB Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGB input 1: Counter count down enabled on the falling edge of GTETRGB input	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	DSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count down enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	DSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count down enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	DSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count down enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	DSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count down enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	DSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count down enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	DSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count down enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	DSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count down enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	DSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count down enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W

Bit	Symbol	Function	R/W
16	DSELCA	ELC_GPTA Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTA input 1: Counter count down enabled at the ELC_GPTA input	R/W
17	DSELCB	ELC_GPTB Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTB input 1: Counter count down enabled at the ELC_GPTB input	R/W
18	DSELCC	ELC_GPTC Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTC input 1: Counter count down enabled at the ELC_GPTC input	R/W
19	DSELCD	ELC_GPTD Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTD input 1: Counter count down enabled at the ELC_GPTD input	R/W
20	DSELCE	ELC_GPTE Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTE input 1: Counter count down enabled at the ELC_GPTE input	R/W
21	DSELCF	ELC_GPTF Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTF input 1: Counter count down enabled at the ELC_GPTF input	R/W
22	DSELCG	ELC_GPTG Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTG input 1: Counter count down enabled at the ELC_GPTG input	R/W
23	DSELCH	ELC_GPTH Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTH input 1: Counter count down enabled at the ELC_GPTH input	R/W
27:24	DSILVL[3:0]	External Input Level Source Count-Down Enable 0 0 0 0: Disables count-down by external input level 0 0 1 0: Enables count-down by GTIOCnA pin input level 0 0 0 1 1: Enables count-down by GTIOCnA pin input level 1 0 1 0 0: Enables count-down by GTIOCnB pin input level 0 0 1 0 1: Enables count-down by GTIOCnB pin input level 1 1 0 0 0: Enables count-down by GTETRGA pin input level 0 1 0 0 1: Enables count-down by GTETRGA pin input level 1 1 0 1 0: Enables count-down by GTETRGB pin input level 0 1 0 1 1: Enables count-down by GTETRGB pin input level 1 Others: Setting prohibited	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The GTDNSR sets the source to count down the GTCNT counter.

When at least one bit in the GTDNSR register is set to 1, the GTCNT counter is counted down by the source that is set to 1 in this register. In this case, GTCR.TPCS has no effect.

Number of decrement in counting is one even when multiple sources are generated simultaneously.

Inputs from GTETRGn (n = A, B) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

DSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Down Enable)

The DSGTRGAR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGA pin input.

DSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Down Enable)

The DSGTRGAF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGA pin input.

DSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Down Enable)

The DSGTRGBR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGB pin input.

DSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Down Enable)

The DSGTRGBF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGB pin input.

DSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Down Enable)

The DSCARBL bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnA pin input, when the GTIOCnB input is 0.

DSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Down Enable)

The DSCARBH bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

DSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Down Enable)

The DSCAFBL bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

DSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Down Enable)

The DSCAFBH bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

DSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Down Enable)

The DSCBRAL bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

DSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Down Enable)

The DSCBRAH bit enables or disables the GTCNT counter count down on the rising edge of GTIOCnB pin input, when GTIOCnA input is 1.

DSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Down Enable)

The DSCBFAL bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

DSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Down Enable)

The DSCBFAH bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

DSELCm bit (ELC_GPTm Event Source Counter Count Down Enable) (m = A to H)

The DSELCm bit enables or disables the GTCNT counter count down at the ELC_GPTm event input.

DSILVL[3:0] bits (External Input Level Source Count-Down Enable)

The DSILVL[3:0] bits enable or disable the count-down of the GTCNT counter by the GTIOCnA pin input level, GTIOCnB pin input level, and GTETRGA/GTETRGB input level.

20.2.10 GTICASR : General PWM Timer Input Capture Source Select Register A

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	ASOC	ASELCH	ASELCG	ASELCF	ASELCE	ASELCD	ASELCC	ASELCB	ASELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ASCBFAH	ASCBFAL	ASCBRAH	ASCBRAL	ASCAFBH	ASCAFBL	ASCARBH	ASCARBL	—	—	—	—	ASGTRGBF	ASGTRGBR	ASGTRGAF	ASGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ASGTRGAR	GTETRGA Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGA input 1: GTCCRA input capture enabled on the rising edge of GTETRGA input	R/W
1	ASGTRGAF	GTETRGA Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGA input 1: GTCCRA input capture enabled on the falling edge of GTETRGA input	R/W
2	ASGTRGBR	GTETRGB Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGB input 1: GTCCRA input capture enabled on the rising edge of GTETRGB input	R/W
3	ASGTRGBF	GTETRGB Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGB input 1: GTCCRA input capture enabled on the falling edge of GTETRGB input	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	ASCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRA input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	ASCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRA input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	ASCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRA input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	ASCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRA input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	ASCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRA input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W

Bit	Symbol	Function	R/W
13	ASCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRA input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	ASCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRA input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	ASCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRA input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	ASELCA	ELC_GPTA Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTA input 1: GTCCRA input capture enabled at the ELC_GPTA input	R/W
17	ASELCB	ELC_GPTB Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTB input 1: GTCCRA input capture enabled at the ELC_GPTB input	R/W
18	ASELCC	ELC_GPTC Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTC input 1: GTCCRA input capture enabled at the ELC_GPTC input	R/W
19	ASELCD	ELC_GPTD Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTD input 1: GTCCRA input capture enabled at the ELC_GPTD input	R/W
20	ASELCE	ELC_GPTE Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTE input 1: GTCCRA input capture enabled at the ELC_GPTE input	R/W
21	ASELCF	ELC_GPTF Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTF input 1: GTCCRA input capture enabled at the ELC_GPTF input	R/W
22	ASELCG	ELC_GPTG Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTG input 1: GTCCRA input capture enabled at the ELC_GPTG input	R/W
23	ASELCH	ELC_GPTH Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTH input 1: GTCCRA input capture enabled at the ELC_GPTH input	R/W
24	ASOC	Other channel Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by other channel factor 1: Enables GTCCRA input capture by other channel factor	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The GTICASR sets the source of input capture for GTCCRA.

When at least one bit among bits in the GTICASR register is set to 1, input capture operation making the GTCCRA register as an input capture register is performed.

Inputs from GTETRGn (n = A, B) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

ASGTRGAR bit (GTETRGA Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGAR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGA pin input.

ASGTRGAF bit (GTETRGA Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGAF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGA pin input.

ASGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRA Input Capture Enable)

The ASGTRGBR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGB pin input.

ASGTRGBF bit (GTETRGB Pin Falling Input Source GTCCRA Input Capture Enable)

The ASGTRGBF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGB pin input.

ASCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable)

The ASCARBL bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

ASCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRA Input Capture Enable)

The ASCARBH bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

ASCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable)

The ASCAFBL bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

ASCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRA Input Capture Enable)

The ASCAFBH bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnA pin input, when the GTIOCnB input is 1.

ASCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable)

The ASCBRAL bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnB pin input, when the GTIOCnA input is 0.

ASCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRA Input Capture Enable)

The ASCBRAH bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

ASCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable)

The ASCBFAL bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

ASCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRA Input Capture Enable)

The ASCBFAH bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

ASELCm bit (ELC_GPTm Event Source Counter GTCCRA Input Capture Enable) (m = A to H)

The ASELCm bit enables or disables the input capture for GTCCRA at the ELC_GPTm event input.

ASOC bit (Other channel Source GTCCRA Input Capture Enable)

The ASOC bit enables or disables an input capture to the GTCCRA register by other channel factor.

Input capture of other channel factors is not subject to input capture factors to other channels set by the ICAFA and ICBFA bits of the GTICCR register.

20.2.11 GTICBSR : General PWM Timer Input Capture Source Select Register B

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	BSOC	BSELCH	BSELCG	BSELCF	BSELCE	BSELCD	BSELCC	BSELCB	BSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSCBFAH	BSCBFAL	BSCBRAH	BSCBRAL	BSCAFBH	BSCAFBL	BSCARBH	BSCARBL	—	—	—	—	BSGTRGBF	BSGTRGBR	BSGTRGAF	BSGTRGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BSGTRGAR	GTETRGA Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGA input 1: GTCCRB input capture enabled on the rising edge of GTETRGA input	R/W
1	BSGTRGAF	GTETRGA Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGA input 1: GTCCRB input capture enabled on the falling edge of GTETRGA input	R/W
2	BSGTRGBR	GTETRGB Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGB input 1: GTCCRB input capture enabled on the rising edge of GTETRGB input	R/W
3	BSGTRGBF	GTETRGB Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGB input 1: GTCCRB input capture enabled on the falling edge of GTETRGB input	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	BSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRB input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	BSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRB input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	BSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRB input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	BSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRB input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	BSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRB input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W

Bit	Symbol	Function	R/W
13	BSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRB input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	BSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRB input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	BSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRB input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	BSELCA	ELC_GPTA Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTA input 1: GTCCRB input capture enabled at the ELC_GPTA input	R/W
17	BSELCB	ELC_GPTB Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTB input 1: GTCCRB input capture enabled at the ELC_GPTB input	R/W
18	BSELCC	ELC_GPTC Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTC input 1: GTCCRB input capture enabled at the ELC_GPTC input	R/W
19	BSELCD	ELC_GPTD Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTD input 1: GTCCRB input capture enabled at the ELC_GPTD input	R/W
20	BSELCE	ELC_GPTE Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTE input 1: GTCCRB input capture enabled at the ELC_GPTE input	R/W
21	BSELCF	ELC_GPTF Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTF input 1: GTCCRB input capture enabled at the ELC_GPTF input	R/W
22	BSELCG	ELC_GPTG Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTG input 1: GTCCRB input capture enabled at the ELC_GPTG input	R/W
23	BSELCH	ELC_GPTH Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTH input 1: GTCCRB input capture enabled at the ELC_GPTH input	R/W
24	BSOC	Other Channel Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled by other channel factor 1: GTCCRB input capture enabled by other channel factor	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The GTICBSR sets the source of input capture for GTCCRB.

When at least one bit among bits in the GTICBSR register is set to 1, input capture operation making the GTCCRB register as an input capture register is performed.

Inputs from GTETRGn (n = A, B) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

BSGTRGAR bit (GTETRGA Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGAR bit enables or disables the input capture for GTCCRB on the rising edge of the GTETRGA pin input.

BSGTRGAF bit (GTETRGA Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGAF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGA pin input.

BSGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRB Input Capture Enable)

The BSGTRGBR bit enables or disables the input capture for GTCCRB on the rising edge of GTETRGB pin input.

BSGTRGBF bit (GTETRGB Pin Falling Input Source GTCCRB Input Capture Enable)

The BSGTRGBF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGB pin input.

BSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable)

The BSCARBL bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnA pin input, when the GTIOCnB input is 0.

BSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRB Input Capture Enable)

The BSCARBH bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

BSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable)

The BSCAFBL bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

BSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRB Input Capture Enable)

The BSCAFBH bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

BSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable)

The BSCBRAL bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

BSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRB Input Capture Enable)

The BSCBRAH bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

BSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable)

The BSCBFAL bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

BSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRB Input Capture Enable)

The BSCBFAH bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

BSELCm bit (ELC_GPTm Event Source Counter GTCCRB Input Capture Enable) (m = A to H)

The BSELCm bit enables or disables the input capture for GTCCRB at the ELC_GPTm event input.

BSOC bit (Other Channel Source GTCCRB Input Capture Enable)

The BSOC bit enables or disables the input capture to the GTCCRB register by other channel factor.

Input capture of other channel factors is not subject to input capture factors to other channels set by the ICAFB and ICBFB bits of the GTICCR register.

20.2.12 GTCR : General PWM Timer Control Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	TPCS[3:0]				—	—	—	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SSCEN	—	—	—	SSCGRP[1:0]		SCGTIOC	ICDS	—	—	BINV	AINV	—	—	—	CST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CST	Count Start 0: Count operation is stopped 1: Count operation is performed	R/W*1
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	AINV	GTIOCnA input/output pin polarity reversal control 0: Not invert 1: Invert	R/W
5	BINV	GTIOCnB input/output pin polarity reversal control 0: Not invert 1: Invert	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	ICDS	Input Capture Operation Select During Count Stop 0: Input capture is operated during count stop. 1: Input capture is not operated during count stop.	R/W
9	SCGTIOC	GTIOC Input Source Synchronous Clear Enable 0: Disables the use of counter clear by GTIOC input as the clear factor for other channels 1: Enables the use of counter clear by GTIOC input as the clear factor for other channels	R/W
11:10	SSCGRP[1:0]	Synchronous Set/Clear Group Select 0 0: Selects synchronous set/clear group A 0 1: Selects synchronous set/clear group B 1 0: Selects synchronous set/clear group C 1 1: Selects synchronous set/clear group D	R/W*1
14:12	—	These bits are read as 0. The write value should be 0.	R/W
15	SSCEN	Synchronous Set/Clear Enable 0: Disables synchronous set/clear of the GTCNT counter 1: Enables synchronous set/clear of the GTCNT counter	R/W*1
19:16	MD[3:0]	Mode Select*2 0x0: Saw-wave PWM mode 1 (single buffer or double buffer possible) 0x1: Saw-wave one-shot pulse mode (fixed buffer operation) 0x2: Saw-wave PWM mode 2 (single buffer or double buffer possible) 0x4: Triangle-wave PWM mode 1 (32-bit transfer at trough) (single buffer or double buffer possible) 0x5: Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) (single buffer or double buffer possible) 0x6: Triangle-wave PWM mode 3 (64-bit transfer at trough) (fixed buffer operation) 0xC: Complementary PWM mode 1 (transfer at crest) 0xD: Complementary PWM mode 2 (transfer at trough) 0xE: Complementary PWM mode 3 (transfer at crest and trough) 0xF: Complementary PWM mode 4 (immediate transfer) Others: Setting prohibited	R/W*1

Bit	Symbol	Function	R/W
22:20	—	These bits are read as 0. The write value should be 0.	R/W
26:23	TPCS[3:0]	Timer Prescaler Select 0x0: GTCLK/1 0x1: GTCLK/2 0x2: GTCLK/4 0x3: GTCLK/8 0x4: GTCLK/16 0x5: GTCLK/32 0x6: GTCLK/64 0x8: GTCLK/256 0xA: GTCLK/1024 0xC: GTETRGA (through the POEG) 0xD: GTETRGB (through the POEG) Others: Setting prohibited	R/W*1
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.
Note 2. MD[3] bit is only available in GPT160 to GPT162. GPT163 only supports Saw-wave PWM mode and Triangle-wave PWM mode except Saw-wave PWM mode 2.

The GTCR controls GTCNT.

Access in 8 bit unit to GTCR is prohibited.

CST bit (Count Start)

The CST bit controls the GTCNT counter start and stop.

[Setting conditions]

- The GTSTR value where the channel number associated with the bit number is set to 1 with the GTSSR.CSTRT bit at 1
- The ELC event input, the external trigger, or the GTIOCnA/GTIOCnB input that are enabled by GTSSR for the starting counter source, occurs (n = 0 to 3)
- 1 is written by software directly.

[Clearing conditions]

- The GTSTP value where the channel number associated with the bit number is set to 1 with the GTPSR.CSTOP bit at 1
- The ELC event input, the external trigger, or the GTIOCnA/GTIOCnB input enabled by GTPSR as the counter stop source, occurs (n = 0 to 3)
- 0 is written by software directly.
- When the period count function is finished while the GTPC.ASTP bit is 1.

AINV bit (GTIOCnA input/output pin polarity reversal control)

The AINV bit selects the GTIOCnA input/output pin polarity reversal control.

BINV bit (GTIOCnB input/output pin polarity reversal control)

The BINV bit selects the GTIOCnB input/output pin polarity reversal control.

ICDS bit (Input Capture Operation Select During Count Stop)

The ICDS bit selects the input capture operation during count stop, when the input capture function is selected.

SCGTIOC bit (GTIOC Input Source Synchronous Clear Enable)

The SCGTIOC bit enables or disables the use of the counter clear by GTIOCnA/GTIOCnB input pin selected by the GTCSR register as the counter clear factor for other channels.

SSCGRP[1:0] bits (Synchronous Set/Clear Group Select)

The SSCGRP[1:0] bits select the channel group of synchronous set/clear.

In complementary PWM mode, slave channels are also controlled by setting the SSCGRP[1:0] bits of the master channel.

SSCEN bit (Synchronous Set/Clear Enable)

The SSCEN bit enables or disables synchronous set/clear.

In complementary PWM mode, slave channels are also controlled by setting the SSCEN bits of the master channel.

MD[3:0] bits (Mode Select)

The MD[3:0] bits select the GPT operating mode.

In complementary PWM mode, slave channels are also controlled by setting the MD bits of the master channel.

Only the MD[3:2] bits (MD[2] for GPT163) are valid at input capture. Counting in saw-wave mode is performed by setting the MD[3:0] bits to 0x0, 0x1 or 0x2 (0x0 or 0x1 in GPT163), and counting in triangle-wave mode is performed by setting the MD[3:0] bits to 0x4, 0x5 or 0x6, and counting in complementary PWM mode is performed by setting the MD[3] bit to 1 (only available in GPT160 to GPT162).

The MD bits must be set while the GTCNT operation is stopped.

During the event count operation (when at least one bit among the bits for the GTUPSR and GTDNSR registers is set to 1), setting of the MD bits are ignored, where counting in saw-wave or triangle-wave or complementary PWM modes is not performed. Instead, up-counting or down-counting by a source set by the GTUPSR and GTDNSR registers is performed.

TPCS[3:0] bits (Timer Prescaler Select)

The TPCS[3:0] bits select the clock for GTCNT. A clock prescaler can be selected independently for each channel. The TPCS[3:0] bits must be set while the GTCNT operation is stopped.

20.2.13 GTUDDTYC : General PWM Timer Count Direction and Duty Setting Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	OABDTYT	OBDTYR	OBDTYF	OBDTY[1:0]		—	—	—	—	OADTYR	OADTYF	OADTY[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	UD	Count Direction Setting 0: GTCNT counts down 1: GTCNT counts up	R/W
1	UDF	Forcible Count Direction Setting 0: Not forcibly set 1: Forcibly set	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
17:16	OADTY[1:0]	GTIOCnA Output Duty Setting 0 0: GTIOCnA pin duty depends on the compare match 0 1: GTIOCnA pin duty depends on the compare match 1 0: GTIOCnA pin duty 0% 1 1: GTIOCnA pin duty 100%	R/W
18	OADTYF	Forcible GTIOCnA Output Duty Setting 0: Not forcibly set 1: Forcibly set	R/W

Bit	Symbol	Function	R/W
19	OADTYR	GTIOCnA Output Value Selecting after Releasing 0%/100% Duty Setting 0: The function selected by the GTIOA[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting. 1: The function selected by the GTIOA[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting.	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
25:24	OBDTY[1:0]	GTIOCnB Output Duty Setting 0 0: GTIOCnB pin duty depends on the compare match 0 1: GTIOCnB pin duty depends on the compare match 1 0: GTIOCnB pin duty 0% 1 1: GTIOCnB pin duty 100%	R/W
26	OBDTYF	Forcible GTIOCnB Output Duty Setting 0: Not forcibly set 1: Forcibly set	R/W
27	OBDTYR	GTIOCnB Output Value Selecting after Releasing 0%/100% Duty Setting 0: The function selected by the GTIOB[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting. 1: The function selected by the GTIOB[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting.	R/W
28	OABDTYT	GTIOCnA, GTIOCnB Pin Output 0%/100% Duty Setting Reflection Timing Setting 0: The output duty-cycle setting changed at an underflow is reflected. 1: The output duty-cycle setting changed at an underflow and at the top of triangle-wave is reflected. The setting value 1 is invalid except the triangle-wave PWM mode.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

The GTUDDTYC sets the direction in which the GTCNT counts (up-counting or down-counting), and sets the duty of the GTIOCnA/GTIOCnB pin output.

The setting is invalid during the event count operation, Saw-wave PWM mode 2, Complementary PWM mode.

Count Direction:

- In saw-wave mode.
  When the UD value is set to 0 during up-counting, the count direction changes at an overflow (the timing synchronous with count clock after the GTCNT value becomes the GTPR value). When the UD value is set to 1 during down-counting, the count direction changes at an underflow (the timing synchronous with count clock after the GTCNT value becomes 0).
  When the UD value changes from 1 to 0 with the UDF bit being 0 and while counting stops, the counter starts up-counting and the count direction changes at an overflow (the timing synchronous with count clock after the GTCNT value becomes the GTPR value). When the UD value changes from 0 to 1 with the UDF bit being 0 and while counting stops, the counter starts down-counting and the count direction changes at an underflow (the timing synchronous with count clock after the GTCNT value becomes 0).
  When the UDF bit is set to 1 while counting stops, the UD bit value is reflected in the count direction when counting starts.
- In triangle-wave mode.
  When the UD value changes during counting, the count direction does not change. When the UD value changes while the UDF bit is 0 and counting stops, the change is not reflected in the count direction when counting starts.
  When the UDF bit is set to 1 while counting is stopped, the UD value is reflected in the count direction when counting starts.

UD bit (Count Direction Setting)

The UD bit sets the count direction (up-counting or down-counting) for GTCNT.

UDF bit (Forcible Count Direction Setting)

The UDF bit forcibly sets the count direction when GTCNT starts operation as the UD value. Only 0 should be written to this bit during counter operation. When 1 is written to this bit while counting stops, return this bit to 0 before counting starts.

Output duty

- In saw-wave mode.
  When the OADTY/OBDTY value changes during up-counting, the duty is reflected at an overflow (GTCNT = GTPR).
  When the OADTY/OBDTY value is changed during down-counting, the duty is reflected at an underflow (GTCNT = 0).
  When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 0 and while counting stops the output duty is not reflected at the starting counter operation. When the count direction is up, the output duty is reflected at an overflow (GTCNT = GTPR). When the count direction is down, the output duty is reflected at an underflow (GTCNT = 0).
  When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 1 and while counting stops, the output duty is reflected at starting counter operation.
- In triangle-wave mode.
  When the OADTY/OBDTY value changes during counting, the duty is reflected at an underflow.
  When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 0 and while counting stops, the output duty is not reflected at the starting counter operation. The output duty is reflected at an underflow.
  When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 1 and while counting stops, the output duty is reflected at starting counter operation.

In both saw-wave mode and triangle-wave mode, when the OADTYF/OBDTYF bit is set back to 0 and the OADTY[1:0]/OBDTY[1:0] bits are set after setting the OADTYF/OBDTYF bit to 1 and setting the OADTY[1:0]/OBDTY[1:0] bits for the duty of first cycle while count operation is stopped, these duty-cycle set during stopping count operation are reflected in the first cycle and the second cycle after starting count operation.

OmDTY[1:0] bits (GTIOCnm Output Duty Setting) (m = A, B)

The OmDTY[1:0] bits set the output duty (0%, 100% or compare match control) of the GTIOCnm pin.

OmDTYF bit (Forcible GTIOCnm Output Duty Setting) (m = A, B)

The OmDTYF bit forcibly sets the output duty cycle to the OmDTY setting. Set this bit to 0 during counter operation.

OmDTYR bit (GTIOCnm Output Value Selecting after Releasing 0%/100% Duty Setting) (m = A, B)

The OmDTYR bit selects the value that is the object of output retained or toggled at cycle end, when the control changes from 0% or 100% duty setting to compare match for the GTIOCnm pin and GTIOR.GTIOm[3:2] bits are set to 00b (output retained at cycle end) or the GTIOR.GTIOm[3:2] bits are set to 11b (output toggled at cycle end).

The GPT internally continues to perform compare match operation during duty-cycle 0% or 100% operation. When the OmDTYR bit is 1, the value after the period has elapsed due this compare match operation is target for the GTIOm[3:2] bits.

OABDTYT bit (GTIOCnm Pin Output 0%/100% Duty Setting Reflection Timing Setting) (m = A, B)

The OABDTYT bit selects the reflection timing of the 0%/100% duty setting for the GTIOCnm pin in triangle-wave mode.

If the OABDTYT bit is 0, the output duty setting set in the GTUDDTYC.OADTY[1:0] and OBDTY[1:0] bits is reflected when an underflow occurs.

If the OABDTYT bit is 1, the output duty setting set in the GTUDDTYC.OADTY[1:0] and OBDTY[1:0] bits is reflected when an underflow occurs and at the top of the triangle-wave.

20.2.14 GTIOR : General PWM Timer I/O Control Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCSB[1:0]		NFBEN	—	OBEOCD	OBDF[1:0]		OBE	OBHLD	OBDFLT	—	GTIOB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NFCSA[1:0]		NFAEN	PSYE	OAEOCD	OADF[1:0]		OAE	OAHLD	OADFLT	CPSCIR	GTIOA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	GTIOA[4:0]	GTIOCnA Pin Function Select See Table 20.4 and Table 20.5.	R/W
5	CPSCIR*1	Complementary PWM Mode Initial Output at Synchronous Clear Disable 0: Outputs the initial value set by the GTIOR.GTIOA and GTIOB bits when synchronous clear occurs in Trough section of complementary PWM mode 1: Disables the output of initial value	R/W
6	OADFLT	GTIOCnA Pin Output Value Setting at the Count Stop 0: The GTIOCnA pin outputs low when counting stops 1: The GTIOCnA pin outputs high when counting stops	R/W
7	OAHLD	GTIOCnA Pin Output Setting at the Start/Stop Count 0: The GTIOCnA pin output level at the start or stop of counting depends on the register setting 1: The GTIOCnA pin output level is retained at the start or stop of counting	R/W
8	OAE	GTIOCnA Pin Output Enable 0: Output is disabled 1: Output is enabled	R/W
10:9	OADF[1:0]	GTIOCnA Pin Disable Value Setting 0 0: None of the below options are specified 0 1: GTIOCnA pin is set to Hi-Z in response to controlling the output disable 1 0: GTIOCnA pin is set to 0 in response to controlling the output disable 1 1: GTIOCnA pin is set to 1 in response to controlling the output disable	R/W
11	OAEOCD*1	GTCCRA Compare Match Cycle End Output Invalidate 0: Validates GTIOA[3:2] setting 1: Invalidates GTIOA[3:2] setting (GTIOCnA pin output is retained)	R/W
12	PSYE	PWM Synchronous Output Enable 0: Disables GTCPPOm pin output 1: Enables GTCPPOm pin output	R/W
13	NFAEN	Noise Filter A Enable 0: The noise filter for the GTIOCnA pin is disabled 1: The noise filter for the GTIOCnA pin is enabled	R/W
15:14	NFCSA[1:0]	Noise Filter A Sampling Clock Select 0 0: GTCLK/1 0 1: GTCLK/4 1 0: GTCLK/16 1 1: GTCLK/64	R/W
20:16	GTIOB[4:0]	GTIOCnB Pin Function Select See Table 20.4 and Table 20.5.	R/W
21	—	This bit is read as 0. The write value should be 0.	R/W
22	OBDFLT	GTIOCnB Pin Output Value Setting at the Count Stop 0: The GTIOCnB pin outputs low when counting stops 1: The GTIOCnB pin outputs high when counting stops	R/W

Bit	Symbol	Function	R/W
23	OBHLD	GTIOCnB Pin Output Setting at the Start/Stop Count 0: The GTIOCnB pin output level at the start/stop of counting depends on the register setting 1: The GTIOCnB pin output level is retained at the start/stop of counting	R/W
24	OBE	GTIOCnB Pin Output Enable 0: Output is disabled 1: Output is enabled	R/W
26:25	OBDF[1:0]	GTIOCnB Pin Disable Value Setting 0 0: None of the below options are specified 0 1: GTIOCnB pin is set to Hi-Z in response to controlling the output disable 1 0: GTIOCnB pin is set to 0 in response to controlling the output disable 1 1: GTIOCnB pin is set to 1 in response to controlling the output disable	R/W
27	OBEOCD*1	GTCCRB Compare Match Cycle End Output Invalidate 0: When in saw-wave PWM mode 1, validates GTIOB[3:2] setting When in saw-wave PWM mode 2, validates GTIOA[3:2] setting 1: When in saw-wave PWM mode 1, invalidates GTIOB[3:2] setting (GTIOCnB pin output is retained) When in saw-wave PWM mode 2, invalidates GTIOA[3:2] setting (GTIOCnA pin output is retained)	R/W
28	—	This bit is read as 0. The write value should be 0.	R/W
29	NFBEN	Noise Filter B Enable 0: The noise filter for the GTIOCnB pin is disabled 1: The noise filter for the GTIOCnB pin is enabled	R/W
31:30	NFCSB[1:0]	Noise Filter B Sampling Clock Select 0 0: GTCLK/1 0 1: GTCLK/4 1 0: GTCLK/16 1 1: GTCLK/64	R/W

Note 1. This bit is only available in GPT160 to GPT162.
In GPT163, this bit is read as 0. The write value should be 0.

The GTIOR sets the functions of the GTIOCnA, GTIOCnB and GTCPPOm pins. (n = 0 to 3, m = 0 to 3)

GTIOA[4:0] bits (GTIOCnA Pin Function Select)

The GTIOA[4:0] bits select the GTIOCnA pin function. For details, see Table 20.4 and Table 20.5.

CPSCIR bit (Complementary PWM Mode Initial Output at Synchronous Clear Disable)

The CPSCIR bit selects the output waveform when synchronous clear occurs in complementary PWM mode.

The initial output is disabled by this function only when synchronous clear occurs in the trough section in complementary PWM mode. If a synchronous clear occurs at any other time, the initial value set by the GTIOA[4] and GTIOB[4] bits is output regardless of the CPSCIR bit setting. In addition, the initial value set by the GTIOA[4] and GTIOB[4] bits is output even when the synchronous clear occurs in the trough section immediately after the count starts.

OADFLT bit (GTIOCnA Pin Output Value Setting at the Count Stop)

The OADFLT bit sets whether the GTIOCnA pin outputs high or low when counting stops.

OAHLD bit (GTIOCnA Pin Output Setting at the Start/Stop Count)

The OAHLD bit specifies whether the GTIOCnA pin output level is retained or the level at the start or stop of counting depends on the register setting.

When the OAHLD bit is set to 0:

- The value specified in bit [4] of the GTIOA[4:0] bits is output when counting starts
- The value specified in the OADFLT bit is output when counting stops
- If the OADFLT bit is modified while counting stops, the new value is immediately reflected in the output.

When the OAHLD bit is set to 1:

- The output is retained when counting starts or stops.

OAE bit (GTIOCnA Pin Output Enable)

The OAE bit disables or enables the GTIOCnA pin output.

When GTCCRA register is used as the input capture register (at least one bit in the GTICASR register is set to 1), the GTIOCnA pin does not output regardless of the OAE bit value.

OADF[1:0] bits (GTIOCnA Pin Disable Value Setting)

The OADF[1:0] bits select the output value of the GTIOCnA pin in response to a request to disable output from the POEG.

OAEOCD bit (GTCCRA Compare Match Cycle End Output Invalidate)

The OAEOCD bit selects the valid/invalid setting of the GTIOA[3:2] bits if the cycle end matches the GTCCRA compare match timing in saw-wave PWM modes 1 and 2. When 1 (disabled) is set, the GTIOCnA pin holds the output when the cycle end and the GTCCRA compare match timing match.

PSYE bit (PWM Synchronous Output Enable)

The PSYE bit enables or disables the output signal from GTCPPOm pin synchronized with the PWM cycle that toggles at the crest/trough/GTCNT counter clear of complementary PWM mode and Triangle-wave mode or the end of the cycle of the Saw-wave mode.

The initial output of the GTCPPOn output pin is low, and the count start results in high.

NFAEN bit (Noise Filter A Enable)

The NFAEN bit disables or enables the noise filter for input from the GTIOCnA pin. Because changing the value of the bit might lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the GTIOR register before doing so.

NFCSA[1:0] bits (Noise Filter A Sampling Clock Select)

The NFCSA[1:0] bits set the sampling interval for the noise filter of the GTIOCnA pin. When setting these bits, wait for 2 cycles of the selected sampling interval before setting the input capture function.

GTIOB[4:0] bits (GTIOCnB Pin Function Select)

The GTIOB[4:0] bits select the GTIOCnB pin function. For details, see Table 20.4 and Table 20.5.

In Saw-wave PWM mode 2, only the GTIOB[1:0] bits are valid, and the GTIOCnA pin output is selected instead of the GTIOCnB pin by the GTCCRB register compare match.

OBDFLT bit (GTIOCnB Pin Output Value Setting at the Count Stop)

The OBDFLT bit sets whether the GTIOCnB pin outputs high or low when counting stops.

OBHLD bit (GTIOCnB Pin Output Setting at the Start/Stop Count)

The OBHLD bit specifies whether the GTIOCnB pin output level is retained or the level at the start or stop of counting depends on the register setting.

When the OBHLD bit is set to 0:

- The value specified in bit [4] of the GTIOB[4:0] bits is output when counting starts
- The value specified in the OBDFLT bit is output when counting stops
- If the OBDFLT bit is modified while counting stops, the new value is immediately reflected in the output.

When the OBHLD bit is set to 1:

- The output is retained when counting starts or stops.

OBE bit (GTIOCnB Pin Output Enable)

The OBE bit disables or enables the GTIOCnB pin output.

When GTCCRB register is used as the input capture register (at least one bit in the GTICBSR register is set to 1), the GTIOCnB pin does not output regardless of the OBE bit value.

OBDF[1:0] bits (GTIOCnB Pin Disable Value Setting)

The OBDF[1:0] bits select the output value of the GTIOCnB pin in response to a request to disable output from the POEG.

OBEOCD bit (GTCCRB Compare Match Cycle End Output Invalidate)

The OBEOCD bit selects the valid/invalid setting of the GTIOB[3:2] bits in saw-wave PWM mode 1 or the setting of the GTIOA[3:2] bits in saw-wave PWM mode 2 if the cycle end matches the GTCCRB compare match timing in saw-wave PWM modes 1 and 2. When 1 (disabled) is set, the GTIOCnB pin in saw-wave PWM mode 1 or the GTIOCnA pin in saw-wave PWM mode 2 holds the output when the cycle end and the GTCCRB compare match timing match.

NFBEN bit (Noise Filter B Enable)

The NFBEN bit disables or enables the noise filter for input from the GTIOCnB pin. Because changing the value of the bit might lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the GTIOR register before doing so.

NFCSB[1:0] bits (Noise Filter B Sampling Clock Select)

The NFCSB[1:0] bits set the sampling interval for the noise filter of the GTIOCnB pin. When setting these bits, wait for 2 cycles of the selected sampling interval before setting the input capture function.

Table 20.4 Settings of GTIOA[4:0] and GTIOB[4:0] bits (Saw-wave mode, Triangle-wave mode) (1 of 2)

GTIOA/GTIOB[4:0] bits					Function		
b4	b3	b2	b1	b0	b4*4	b3, b2*1 *2 *3 *4	b1, b0*2
0	0	0	0	0	Initial output is low	Output retained at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	0	1	0			High output at GTCCRA/GTCCRB compare match
0	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	0	1	0	0		Low output at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	1	1	0			High output at GTCCRA/GTCCRB compare match
0	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	0	0	0		High output at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	0	1	0			High output at GTCCRA/GTCCRB compare match
0	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	1	0	0		Output toggled at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	1	1	0			High output at GTCCRA/GTCCRB compare match
0	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

Table 20.4 Settings of GTIOA[4:0] and GTIOB[4:0] bits (Saw-wave mode, Triangle-wave mode) (2 of 2)

GTIOA/GTIOB[4:0] bits					Function		
b4	b3	b2	b1	b0	b4*4	b3, b2*1 *2 *3 *4	b1, b0*2
1	0	0	0	0	Initial output is high	Output retained at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	0	1	0			High output at GTCCRA/GTCCRB compare match
1	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	1	0	0		Low output at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	1	1	0			High output at GTCCRA/GTCCRB compare match
1	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	0	0	0		High output at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	0	1	0			High output at GTCCRA/GTCCRB compare match
1	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	1	0	0		Output toggled at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	1	1	0			High output at GTCCRA/GTCCRB compare match
1	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

Note 1. The cycle end means an overflow (GTCNT changes from GTPR to 0 in up-counting), an underflow (GTCNT changes from 0 to GTPR in down-counting), or counter clearing for saw-wave mode, and means a trough (GTCNT changes from 0 to 1) for triangle-wave mode.
Note 2. When the timing of a cycle end and the timing of a GTCCRA/GTCCRB compare match are the same in a compare-match operation, if the OAEOCD and OBEOCD bits are set to 0 and the end of cycle output is enabled, the b3 and b2 settings are given priority in saw-wave PWM mode, and the b1 and b0 settings are given priority in any other mode.
Note 3. In event count operation where at least one bit in GTUPSR or GTDNSR is set to 1, the setting of b3 and b2 is ignored (GPT163).
Note 4. In Saw-wave PWM mode 2, GTIOB[4:2] bits is invalid. Since only GTIOCnA pins are output pins, set GTIOA[4] bit for initial output. Set GTIOA[3:2] bits for output at the end of a cycle.

Table 20.5 Settings of GTIOA[4:0] and GTIOB[4:0] bits (Complementary PWM mode)

GTIOA/GTIOB[4:0] bits*1 *2 *3					Function		
					Initial output, Active level	Up count Compare Match output	Down count Compare Match output
b4	b3	b2	b1	b0	b4	b3, b2	b1, b0
0	0	1	1	0	Initial output is Low. Active level is High.	Low output	High output
0	1	0	0	1		High output	Low output
1	0	1	1	0	Initial output is High. Active level is Low.	Low output	High output
1	1	0	0	1		High output	Low output

Note 1. In complementary PWM mode, the only values that can be set in the GTIOA[4:0] bits are 01001b, and 10110b. Setting other values is prohibited.
Note 2. In complementary PWM mode, the only values that can be set in the GTIOB[4:0] bits are 00110b, and 11001b. Setting other values is prohibited.
Note 3. In complementary PWM mode, setting GTIOB[4:0] bits does not use compare match of GTCCRB register. The combination of counter and register that is the target of compare match depends on the operation period of complementary PWM mode. For details, see section 20.3.3.7. Complementary PWM Modes 1, 2, 3.

20.2.15 GTINTAD : General PWM Timer Interrupt Output Setting Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	GRPABL	GRPABH	GRPDTE	—	—	GRP[1:0]		—	—	—	—	ADTRBDEN	ADTRBUEN	ADTRADEN	ADTRAUEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SCFPU	SCFPO	SCFF	SCFE	SCFD	SCFC	SCFB	SCFA	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	SCFA	GTCCRA Register Compare Match/Input Capture Source Synchronous Clear Enable 0: Disables use of GTCCRA register compare match/input capture as a clear factor for other channels 1: Enables use of GTCCRA register compare match/input capture as a clear factor for other channels	R/W
9	SCFB	GTCCRB Register Compare Match/Input Capture Source Synchronous Clear Enable 0: Disables use of GTCCRB register compare match/input capture as a clear factor for other channels 1: Enables use of GTCCRB register compare match/input capture as a clear factor for other channels	R/W
10	SCFC	GTCCRC Register Compare Match Source Synchronous Clear Enable 0: Disables use of GTCCRC register compare match as a clear factor for other channels 1: Enables use of GTCCRC register compare match as a clear factor for other channels	R/W
11	SCFD	GTCCRD Register Compare Match Source Synchronous Clear Enable 0: Disables use of GTCCRD register compare match as a clear factor for other channels 1: Enables use of GTCCRD register compare match as a clear factor for other channels	R/W
12	SCFE	GTCCRE Register Compare Match Source Synchronous Clear Enable 0: Disables use of GTCCRE register compare match as a clear factor for other channels 1: Enables use of GTCCRE register compare match as a clear factor for other channels	R/W
13	SCFF	GTCCRF Register Compare Match Source Synchronous Clear Enable 0: Disables use of GTCCRF register compare match as a clear factor for other channels 1: Enables use of GTCCRF register compare match as a clear factor for other channels	R/W
14	SCFPO	Overflow Source Synchronous Clear Enable 0: Disables use of overflow as a clear factor for other channels 1: Enables use of overflow as a clear factor for other channels	R/W
15	SCFPU	Underflow Source Synchronous Clear Enable 0: Disables use of underflow as a clear factor for other channels 1: Enables use of underflow as a clear factor for other channels	R/W
16	ADTRAUEN	GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled 1: A/D conversion start request is enabled	R/W

Bit	Symbol	Function	R/W
17	ADTRADEN	GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled 1: A/D conversion start request is enabled	R/W
18	ADTRBUEN	GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled 1: A/D conversion start request is enabled	R/W
19	ADTRBDEN	GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled 1: A/D conversion start request is enabled	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
25:24	GRP[1:0]	Output Disable Source Select 0 0: Group A output disable source is selected 0 1: Group B output disable source is selected Others: Setting prohibited	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
28	GRPDTE	Dead Time Error Output Disable Request Enable*1 0: Dead time error output disable request is disabled 1: Dead time error output disable request is enabled	R/W
29	GRPABH	Same Time Output Level High Disable Request Enable 0: Same time output level high disable request disabled 1: Same time output level high disable request enabled	R/W
30	GRPABL	Same Time Output Level Low Disable Request Enable 0: Same time output level low disable request disabled 1: Same time output level low disable request enabled	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. This bit is only available in GPT160 to GPT162.
In GPT163, this bit is read as 0. The write value should be 0.

The GTINTAD enables or disables interrupt requests, A/D conversion start request, and output disable requests.

SCFA bit (GTCCRA Register Compare Match/Input Capture Source Synchronous Clear Enable)

The SCFA bit enables or disables use of GTCCRA register compare match/input capture as a clear factor for other channels.

SCFB bit (GTCCRB Register Compare Match/Input Capture Source Synchronous Clear Enable)

The SCFB bit enables or disables use of GTCCRB register compare match/input capture as a clear factor for other channels.

The setting is invalid in complementary PWM mode.

SCFC bit (GTCCRC Register Compare Match Source Synchronous Clear Enable)

The SCFC bit enables or disables use of GTCCRC register compare match as a clear factor for other channels.

The setting is invalid in complementary PWM mode.

SCFD bit (GTCCRD Register Compare Match Source Synchronous Clear Enable)

The SCFD bit enables or disables use of GTCCRD register compare match as a clear factor for other channels.

The setting is invalid in complementary PWM mode.

SCFE bit (GTCCRE Register Compare Match Source Synchronous Clear Enable)

The SCFE bit enables or disables use of GTCCRE register compare match as a clear factor for other channels.

The setting is invalid in complementary PWM mode.

SCFF bit (GTCCRF Register Compare Match Source Synchronous Clear Enable)

The SCFF bit enables or disables use of GTCCRF register compare match as a clear factor for other channels.

The setting is invalid in complementary PWM mode.

SCFPO bit (Overflow Source Synchronous Clear Enable)

The SCFPO bit enables or disables use of overflow as a clear factor for other channels.

In complementary PWM mode, it is valid only for the master channel.

SCFPU bit (Underflow Source Synchronous Clear Enable)

The SCFPU bit enables or disables use of underflow as a clear factor for other channels.

In complementary PWM mode, it is valid only for the master channel.

ADTRAUEN bit (GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Enable)

The ADTRAUEN bit enables or disables A/D conversion start requests generated by GTADTRA register compare matches during GTCNT counter up-counting.

In complementary PWM mode, it is valid only for the master channel.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

ADTRADEN bit (GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Enable)

The ADTRADEN bit enables or disables A/D conversion start requests generated by GTADTRA register compare matches during GTCNT counter down-counting.

In complementary PWM mode, it is valid only for the master channel.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

ADTRBUEN bit (GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Enable)

The ADTRBUEN bit enables or disables A/D conversion start requests generated by GTADTRB register compare matches during GTCNT counter up-counting.

In complementary PWM mode, it is valid only for the master channel.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

ADTRBDEN bit (GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Enable)

The ADTRBDEN bit enables or disables A/D conversion start requests generated by GTADTRB register compare matches during GTCNT counter down-counting.

In complementary PWM mode, it is valid only for the master channel.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

GRP[1:0] bits (Output Disable Source Select)

The GRP[1:0] bits select the group of output disable request from GPT to POEG and the group of output disable for GTIOCnA pin and GTIOCnB pin from POEG to GPT.

The output disable request to POEG is output to the group selected in the GRP[1:0] bits, with dead-time errors, simultaneous high output, and simultaneous low output factors following their respective disable request enable bits.

GTST.ODF shows the request of the output disable source group that is selected with the GRP[1:0] bits. Set the GRP[1:0] bits when both GTIOR.OAE and GTIOR.OBE bits are 0.

GRPDTE bit (Dead Time Error Output Disable Request Enable)

The GRPDTE bit enables or disables the output disable request by a dead time error.

The dead time error output disable request is not generated during the event count operation.

GRPABH bit (Same Time Output Level High Disable Request Enable)

The GRPABH bit enables or disables the output disable request when the GTIOCnA pin and GTIOCnB pin output 1 at the same time.

GRPABL bit (Same Time Output Level Low Disable Request Enable)

The GRPABL bit enables or disables the output disable request when the GTIOCnA pin and GTIOCnB pin output 0 at the same time.

20.2.16 GTST : General PWM Timer Status Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x3C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PCF	OABLF	OABHF	DTEF	—	—	—	ODF	—	—	—	—	ADTRBDF	ADTRBUF	ADTRADF	ADTRAUF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TUCF	—	—	—	—	ITCNT[2:0]			TCFPU	TCFPO	TCFF	TCFE	TCFD	TCFC	TCFB	TCFA
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCFA	Input Capture/Compare Match Flag A 0: No input capture/compare match of GTCCRA is generated 1: An input capture/compare match of GTCCRA is generated	R/W*1
1	TCFB	Input Capture/Compare Match Flag B 0: No input capture/compare match of GTCCRB is generated 1: An input capture/compare match of GTCCRB is generated	R/W*1
2	TCFC	Input Compare Match Flag C 0: No compare match of GTCCRC is generated 1: A compare match of GTCCRC is generated	R/W*1
3	TCFD	Input Compare Match Flag D 0: No compare match of GTCCRD is generated 1: A compare match of GTCCRD is generated	R/W*1
4	TCFE	Input Compare Match Flag E 0: No compare match of GTCCRE is generated 1: A compare match of GTCCRE is generated	R/W*1
5	TCFF	Input Compare Match Flag F 0: No compare match of GTCCRF is generated 1: A compare match of GTCCRF is generated	R/W*1
6	TCFPO	Overflow Flag 0: No overflow (crest) occurred 1: An overflow (crest) occurred	R/W*1
7	TCFPU	Underflow Flag 0: No underflow (trough) occurred 1: An underflow (trough) occurred	R/W*1
10:8	ITCNT[2:0]	GPTn_OVF/GPTn_UDF Interrupt Skipping Count Counter	R
14:11	—	These bits are read as 0. The write value should be 0.	R/W
15	TUCF	Count Direction Flag 0: GTCNT counter counts downward 1: GTCNT counter counts upward	R
16	ADTRAUF	GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Flag 0: No GTADTRA register compare match has occurred in up-counting 1: A GTADTRA register compare match has occurred in up-counting	R/W*1
17	ADTRADF	GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Flag 0: No GTADTRA register compare match has occurred in down-counting 1: A GTADTRA register compare match has occurred in down-counting	R/W*1

Bit	Symbol	Function	R/W
18	ADTRBUF	GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Flag 0: No GTADTRB register compare match has occurred in up-counting 1: A GTADTRB register compare match has occurred in up-counting	R/W*1
19	ADTRBDF	GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Flag 0: No GTADTRB register compare match has occurred in down-counting 1: A GTADTRB register compare match has occurred in down-counting	R/W*1
23:20	—	These bits are read as 0. The write value should be 0.	R/W
24	ODF	Output Disable Flag 0: No output disable request is generated 1: An output disable request is generated	R
27:25	—	These bits are read as 0. The write value should be 0.	R/W
28	DTEF	Dead Time Error Flag*2 0: No dead time error has occurred 1: A dead time error has occurred	R
29	OABHF	Same Time Output Level High Flag 0: No simultaneous generation of 1 both for the GTIOCA and GTIOCB pins has occurred 1: A simultaneous generation of 1 both for the GTIOCA and GTIOCB pins has occurred	R
30	OABLF	Same Time Output Level Low Flag 0: No simultaneous generation of 0 both for the GTIOCA and GTIOCB pins has occurred 1: A simultaneous generation of 0 both for the GTIOCA and GTIOCB pins has occurred	R
31	PCF	Period Count Function Finish Flag 0: No period count function finish has occurred 1: A period count function finish has occurred	R/W*1

Note 1. Only 0 can be written to this bit. Do not write 1.
When clearing the ADTRAUF, ADTRADF, ADTRBUF, or ADTRBDF flag, be sure to write 0 only to the target flag or flags for clearing and to write 1 to the other flags not for clearing.
Note 2. This bit is only available in GPT160 to GPT162.
In GPT163, this bit is read as 0. The write value should be 0.

The GTST indicates the status of the GPT.

TCFA flag (Input Capture/Compare Match Flag A)

The TCFA flag indicates the status for the input capture or compare match of GTCCRA.

[Setting conditions]

- GTCNT = GTCCRA, when the GTCCRA register functions as a compare match register
- GTCNT counter value is transferred to GTCCRA by the input capture signal when the GTCCRA register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFB flag (Input Capture/Compare Match Flag B)

The TCFB flag indicates the status for the input capture or compare match of GTCCRB.

[Setting conditions]

- GTCNT = GTCCRB, when the GTCCRB register functions as a compare match register
- GTCNT counter value is transferred to GTCCRB by the input capture signal when the GTCCRB register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

TCFC flag (Input Compare Match Flag C)

The TCFC flag indicates the status for the compare match of GTCCRC.

When GTCCRC performs buffer operation, GTCCRC does not perform compare match.

[Setting condition]

- GTCNT = GTCCRC.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[3:0] = 0x1 (saw-wave one-shot pulse mode)
- GTCR.MD[3:0] = 0x6 (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (GTCCRC performs buffer operation).

TCFD flag (Input Compare Match Flag D)

The TCFD flag indicates the status for the compare match of GTCCRD.

When GTCCRD performs buffer operation, GTCCRD does not perform compare match.

[Setting condition]

- GTCNT = GTCCRD.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[3:0] = 0x1 (saw-wave one-shot pulse mode)
- GTCR.MD[3:0] = 0x6 (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 10b, 11b (GTCCRD performs buffer operation).

TCFE flag (Input Compare Match Flag E)

The TCFE flag indicates the status for the compare match of GTCCRE.

When GTCCRE performs buffer operation, GTCCRE does not perform compare match.

[Setting condition]

- GTCNT = GTCCRE.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[3:0] = 0x1 (saw-wave one-shot pulse mode)
- GTCR.MD[3:0] = 0x6 (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (GTCCRE performs buffer operation).

TCFF flag (Input Compare Match Flag F)

The TCFF flag indicates the status for the compare match of GTCCRF.

When GTCCRF performs buffer operation, GTCCRF does not perform compare match.

[Setting condition]

- GTCNT = GTCCRF.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[3:0] = 0x1 (saw-wave one-shot pulse mode)
- GTCR.MD[3:0] = 0x6 (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 10b, 11b (GTCCRF performs buffer operation).

TCFPO flag (Overflow Flag)

The TCFPO flag indicates when an overflow or crest has occurred.

[Setting conditions]

- In saw-wave mode, an overflow (GTCNT changes from GTPR to 0 in up-counting) has occurred
- In triangle-wave mode, a crest (GTCNT changes from GTPR to GTPR - 1) has occurred
- In counting by hardware sources, an overflow (GTCNT changes from GTPR to 0 in up-counting) has occurred.

[Clearing condition]

- 0 is written to this flag.

TCFPU flag (Underflow Flag)

The TCFPU flag indicates when an underflow or trough has occurred.

[Setting conditions]

- In saw-wave mode, an underflow (GTCNT changes from 0 to GTPR in down-counting) has occurred
- In triangle-wave mode, a trough (GTCNT changes from 0 to 1) has occurred
- In counting by hardware sources, an underflow (GTCNT changes from 0 to GTPR in down-counting) has occurred.

[Clearing condition]

- 0 is written to this bit.

ITCNT[2:0] flag (GPTn_OVF/GPTn_UDF Interrupt Skipping Count Counter)

When the GPTn_OVF/GPTn_UDF interrupt skipping function is used (the GTITC.IVTC[1:0] bits are set to a value other than 00b), the counter is incremented by 1 every time the GPTn_OVF/GPTn_UDF interrupt source which is selected in GTITC.IVTC[1:0] is generated.

These bits are operated independently from the extended interrupt skipping by the GTEITC register.

[Clearing conditions]

- The GPTn_OVF/GPTn_UDF interrupt skipping function is not used (the GTITC.IVTT[2:0] bits are 000b when the IVTC[1:0] bits are 00b).
- The GPTn_OVF/GPTn_UDF interrupt skipping count matches the specified count (the ITCNT[2:0] bits match the skipping count specified by the IVTT[2:0] bits).
- When the count operation is stopped

TUCF flag (Count Direction Flag)

The TUCF flag indicates the count direction of GTCNT. In event count operation, this flag is set to 1 in up-counting and to 0 in down-counting.

ADTRAUF flag (GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Flag)

The ADTRAUF flag indicates generation of a GTADTRA register compare match in up-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRA register in up-counting.

[Clearing condition]

- 0 is written to the ADTRAUF flag.

ADTRADF flag (GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Flag)

The ADTRADF flag indicates generation of a GTADTRA register compare match in down-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRA register in down-counting.

[Clearing condition]

- 0 is written to the ADTRADF flag.

ADTRBUF flag (GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Flag)

The ADTRBUF flag indicates generation of a GTADTRB register compare match in up-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRB register in up-counting.

[Clearing condition]

- 0 is written to the ADTRBUF flag.

ADTRBDF flag (GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Flag)

The ADTRBDF flag indicates generation of a GTADTRB register compare match in down-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRB register in down-counting.

[Clearing condition]

- 0 is written to the ADTRBDF flag.

ODF flag (Output Disable Flag)

The ODF flag shows the request of the output disable source group that is selected in the GRP[1:0] bits.

When output is disabled, an output disable control is not released within the same cycle in which an output disable request is negated. It is released in the next cycle.

DTEF flag (Dead Time Error Flag)

The DTEF flag indicates that the timer output toggle point after the automatic addition of dead time has exceeded the count period.

This flag returns to 0 when the timer output toggle point after the automatic addition of dead time is back within the period.

This flag can only be read from. (writing 0 to clear the flag is not allowed.)

When the output disable request by the DTEF flag is enabled (when GTINTAD.GRPDTE bit is 1), the DTEF flag is output as the output disable request to the POEG. The GPT does not have a dead time error interrupt. Use the interrupt function in the POEG if this is necessary.

[Setting condition]

- When a change point of the waveform after the automatic setting of dead time has exceeded the count period (in the following cases).
  - Up-counting in triangle-wave mode:
    GTCCRA register – GTDVU register ≤ 0
  - Down-counting in triangle-wave mode:
    GTCCRA register – GTDVD register < 0
  - Up-counting in saw-wave one-shot pulse mode:
    GTCCRA register – GTDVU register < 0, or

GTCCRA register + GTDVD register > GTPR register

– Down-counting in saw-wave one-shot pulse mode:
GTCCRA register + GTDVU register > GTPR register, or
GTCCRA register - GTDVD register < 0

[Clearing condition]

- The timer output toggle point after the automatic addition of dead time is within the count period.

OABHF flag (Same Time Output Level High Flag)

The OABHF flag indicates that the GTIOCnA pin and GTIOCnB pin output 1 at the same time.

When the GTIOCnA or GTIOCnB pin outputs 0, this flag returns to 0. This flag is read only. Writing 0 to clear the flag is prohibited.

When the output disable request by the OABHF flag is enabled (GTINTAD.GRPABH = 1), the OABHF flag is output to POEG as an output disable request. The GPT does not have an interrupt to indicate that outputs have been simultaneous driven to the high level. Use the interrupt function in the POEG if this is necessary.

[Setting condition]

- The GTIOCnA and GTIOCnB pins output 1 at the same time when both OAE and OBE bits are set to 1.

[Clearing conditions]

- The GTIOCnA pin output value is different from the GTIOCnB pin output value when both OAE and OBE bits are set to 1
- The GTIOCnA and GTIOCnB pins output 0 at the same time when both OAE and OBE bits are set to 1
- Either the OAE bit or OBE bit is set to 0.

OABLF flag (Same Time Output Level Low Flag)

The OABLF flag indicates that the GTIOCnA and GTIOCnB pins output 0 at the same time.

When the GTIOCnA pin or GTIOCnB pin outputs 1, this flag returns to 0. This flag is read only. Writing 0 to clear the flag is prohibited.

When the output disable request by the OABLF flag is enabled (GTINTAD.GRPABL = 1), the OABLF flag is output to POEG as an output disable request. The GPT does not have an interrupt to indicate that outputs have been simultaneous driven to the low level. Use the interrupt function in the POEG if this is necessary.

[Setting condition]

- The GTIOCnA and GTIOCnB pins output 0 at the same time when both OAE and OBE bits are set to 1.

[Clearing conditions]

- The GTIOCnA pin output value is different from the GTIOCnB pin output value when both OAE and OBE bits are set to 1
- The GTIOCnA and GTIOCnB pins output 1 at the same time when both OAE and OBE bits are set to 1
- Either the OAE bit or the OBE bit is set to 0.

The compare-target signals to generate the OABHF/OABLF flag are the compare match outputs (PWM outputs) signals before they are masked by the output disable function. Even during the output disable condition, compare match operation continues internally, where the OABHF or OABLF flag is updated based on the operation results.

PCF flag (Period Count Function Finish Flag)

This bit is status flag of period count function finish.

[Setting condition]

- The GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 1 at the end of cycle.
- The GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 0 at the count clock.

[Clearing condition]

- 0 is written to this bit.

20.2.17 GTBER : General PWM Timer Buffer Enable Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ADTDB	ADTTB[1:0]		—	ADTDA	ADTTA[1:0]		—	CCRSWT	PR[1:0]		CCRB[1:0]		CCRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	DBRTECB	—	DBRTECA	—	—	—	—	BD3	BD2	BD1	BD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BD0	GTCCR Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
1	BD1	GTPR Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W*1
2	BD2	GTADTRA/GTADTRB Registers Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
3	BD3	GTDVU/GTDVD Registers Buffer Operation Disable*2 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	DBRTECA	GTCCRA Register Double Buffer Repeat Operation Enable 0: GTCCRA register double buffer repeat operation is disabled 1: GTCCRA register double buffer repeat operation is enabled	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	DBRTECB	GTCCRB Register Double Buffer Repeat Operation Enable 0: GTCCRB register double buffer repeat operation is disabled 1: GTCCRB register double buffer repeat operation is enabled	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
17:16	CCRA[1:0]	GTCCRA Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTCCRA ↔GTCCRC) Others: Double buffer operation (GTCCRA ↔ GTCCRC ↔ GTCCRD)	R/W
19:18	CCRB[1:0]	GTCCRB Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTCCRB ↔ GTCCRE) Others: Double buffer operation (GTCCRB ↔ GTCCRE ↔ GTCCRF)	R/W
21:20	PR[1:0]	GTPR Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTPBR → GTPR) Others: Double buffer operation (GTPDBR → GTPBR → GTPR)	R/W
22	CCRSWT	GTCCRA and GTCCRB Forcible Buffer Operation Writing 1 to this bit forces a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after 1 is written. This bit is read as 0.	W
23	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
25:24	ADTTA[1:0]	GTADTRA Register Buffer Transfer Timing Select 0 0: In triangle-wave or complementary PWM mode, no transfer In saw-wave mode, no transfer 0 1: In triangle-wave or complementary PWM mode, transfer at crest In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing 1 0: In triangle-wave or complementary PWM mode, transfer at trough In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing 1 1: In triangle-wave or complementary PWM mode, transfer at both crest and trough In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing	R/W
26	ADTDA	GTADTRA Register Double Buffer Operation 0: Single buffer operation (GTADTBRA → GTADTRA) 1: Double buffer operation (GTADTDBRA → GTADTBRA → GTADTRA)	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
29:28	ADTTB[1:0]	GTADTRB Register Buffer Transfer Timing Select 0 0: In triangle-wave or complementary PWM mode, no transfer In saw-wave mode, no transfer 0 1: In triangle-wave or complementary PWM mode, transfer at crest In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing 1 0: In triangle-wave or complementary PWM mode, transfer at trough In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing 1 1: In triangle-wave or complementary PWM mode, transfer at both crest and trough In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing	R/W
30	ADTDB	GTADTRB Register Double Buffer Operation 0: Single buffer operation (GTADTBRB → GTADTRB) 1: Double buffer operation (GTADTDBRB → GTADTBRB → GTADTRB)	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. In complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.
Note 2. This bit is only available in GPT160 to GPT162.
In GPT163, this bit is read as 0. The write value should be 0.

The GTBER register provides settings for the buffer operation. Set the GTBER register except the BDx (x = 0 to 3) bits while the GTCNT counter is stopped.

BD0 bit (GTCCR Buffer Operation Disable)

The BD0 bit disables the buffer operation using GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, and GTCCRF combined.

When GTDTCR.TDE is 1 and when BD0 is set to 0, GTCCRB does not perform buffer operation in Saw-wave one-shot pulse mode or Triangle-wave PWM mode. The GTCCRB register is automatically set to a compare match value for negative-phase waveform with dead time.

In complementary PWM mode, it is valid only for the buffer operation of GTCCRC register and GTCCRE register. The buffer operation of GTCCRA registers cannot be disabled. Buffer operation of GTCCRE register and GTCCRF register is enabled/disabled by CP3DB bit of GTBER2 register. No buffer transfer to GTCCRB is performed in complementary PWM mode.

A value for the BD0 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the GTSECR.SBDCE or GTSECR.SBDCD.

When the DBRTECm (m = A, B) bit is 1, setting the BD0 bit to 1 while the mode of operation is saw-wave one-shot pulse mode or triangle-wave PWM mode 3 results in transfer from the intermediate buffer to the GTCCRm register.

BD1 bit (GTPR Buffer Operation Disable)

The BD1 bit disables the buffer operation using GTPR, GTPDBR, and GTPBR combined.

A value for the BD1 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the GTSECR.SBDPE or GTSECR.SBDPD.

In complementary PWM mode, the slave channel is also controlled by setting the BD1 bit of the master channel.

BD2 bit (GTADTRA/GTADTRB Registers Buffer Operation Disable)

The BD2 bit disables buffer operation using the GTADTRA, GTADTBRA, and GTADTDBRA registers together and buffer operation using the GTADTRB, GTADTBRB, and GTADTDBRB registers together.

The setting is invalid during the event count operation, and the buffer operation using the GTADTRA and GTADTRB registers is not performed.

A value for the BD2 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDAE or SBDAD bit in the GTSECR register.

BD3 bit (GTDVU/GTDVD Registers Buffer Operation Disable)

The BD3 bit disables buffer operation using the GTDVU and GTDBU registers together and using the GTDVD and GTDBD registers together in saw-wave PWM mode 1, saw-wave one-shot pulse mode or triangle-wave PWM mode.

Even though the BD3 bit is set to 0, buffer operation in the GTDVD register is not performed if the GTDTCR.TDFER bit is set to 1. Instead, the value in the GTDVU register is set automatically.

In saw-wave PWM mode 2 or complementary PWM mode, this bit is invalid and GTDVU and GTDVD registers don’t perform buffer operation.

The setting is invalid during the event count operation, and the buffer operation using the GTDVU and GTDVD registers is not performed.

A value for the BD3 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDDE or SBDDD bit in the GTSECR register.

DBRTECm bits (GTCCRm Register Double Buffer Repeat Operation Enable) (m = A, B)

The DBRTECm bits enable the GTCCRA register double buffer repeat operation.

This setting enables the operation to repeat a transfer to the GTCCRm register from the intermediate buffer by cycle during the buffer transfer disable period when performing the double buffer operation using the GTCCRm register.

It is valid in saw-wave one-shot pulse mode and triangle-wave PWM mode 3.

The disabling period of buffer transfer indicates the period that buffer transfer is stopped by setting of the BD0 bit (CPU writing or simultaneous buffer operation control by the GTSECSR register) and the period for buffer transfer extended skipping (except the case for skipping by counting both crests and troughs) by the GTEITLB register.

When the DBRTECm bit is 1, writing by the CPU to the GTCCRm register sets the same value in temporary register x (x = C, E). The value of the GTCCRx (x = C, E) register is also transferred to the temporary register x (x = C, E) by forcible buffer transfer.

CCRA[1:0] bits (GTCCRA Buffer Operation)

The CCRA[1:0] bits set the buffer operation with GTCCRA, GTCCRC, and GTCCRD combined. When the buffer operation is restricted by the operating mode set in GTCR, the GTCR setting is given priority.

The buffer operation mode is fixed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough), or complementary PWM mode.

CCRB[1:0] bits (GTCCRB Buffer Operation)

The CCRB[1:0] bits set the buffer operation using GTCCRB, GTCCRE, and GTCCRF combined. When the buffer operation is restricted by the operating mode set in GTCR, the GTCR setting is given priority.

The buffer operation mode is fixed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough), or complementary PWM mode.

PR[1:0] bits (GTPR Buffer Operation)

The PR[1:0] bits set the buffer operation with GTPR, GTPDBR, and GTPBR combined.

The setting is invalid in complementary PWM mode. Buffer operation unique to complementary PWM mode is performed regardless of the setting value of the PR[1:0] bits.

CCRSWT bit (GTCCRA and GTCCRB Forcible Buffer Operation)

Writing 1 to the CCRSWT bit forces a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after the 1 is written. This bit is read as 0, and is valid only when counting is stopped with a compare match operation specified.

The setting is invalid in complementary PWM mode.

ADTTA[1:0] bits (GTADTRA Register Buffer Transfer Timing Select)

The ADTTA[1:0] bits set the transfer timing for buffer operation of the GTADTRA, GTADTBRA, and GTADTDBRA registers.

The setting is invalid during the event count operation.

ADTDA bit (GTADTRA Register Double Buffer Operation)

The ADTDA bit sets buffer operation with the GTADTRA, GTADTBRA, and GTADTDBRA registers combined.

The setting is invalid during the event count operation.

ADTTB[1:0] bits (GTADTRB Register Buffer Transfer Timing Select)

The ADTTB[1:0] bits set the transfer timing for buffer operation of the GTADTRB, GTADTBRB, and GTADTDBRB registers.

The setting is invalid during the event count operation.

ADTDB bit (GTADTRB Register Double Buffer Operation)

The ADTDB bit sets buffer operation with the GTADTRB, GTADTBRB, and GTADTDBRB registers combined.

The setting is invalid during the event count operation.

20.2.18 GTITC : General PWM Timer Interrupt and A/D Conversion Start Request Skipping Setting Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x44

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ADTBL	—	ADTAL	—	IVTT[2:0]			IVTC[1:0]		ITLF	ITLE	ITLD	ITLC	ITLB	ITLA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ITLA	GTCCRA Register Compare Match/Input Capture Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function	R/W
1	ITLB	GTCCRB Register Compare Match/Input Capture Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function	R/W
2	ITLC	GTCCRC Register Compare Match Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function	R/W
3	ITLD	GTCCRD Register Compare Match Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function	R/W

Bit	Symbol	Function	R/W
4	ITLE	GTCCRE Register Compare Match Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function	R/W
5	ITLF	GTCCRF Register Compare Match Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function	R/W
7:6	IVTC[1:0]	GPTn_OVF/GPTn_UDF Interrupt Skipping Function Select 0 0: Skipping is not performed 0 1: Both overflow and underflow for saw waves*1 and crest for triangle waves and complementary PWM mode are counted and skipped 1 0: Both overflow and underflow for saw waves*1 and trough for triangle waves and complementary PWM mode are counted and skipped 1 1: Both overflow and underflow for saw waves*1 and both crest and trough for triangle waves and complementary PWM mode are counted and skipped	R/W
10:8	IVTT[2:0]	GPTn_OVF/GPTn_UDF Interrupt Skipping Count Select 0 0 0: Skipping is not performed 0 0 1: Skipping count of 1 0 1 0: Skipping count of 2 0 1 1: Skipping count of 3 1 0 0: Skipping count of 4 1 0 1: Skipping count of 5 1 1 0: Skipping count of 6 1 1 1: Skipping count of 7	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	ADTAL	GTADTRA Register A/D Conversion Start Request Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
14	ADTBL	GTADTRB Register A/D Conversion Start Request Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Saw-wave PWM mode 2 is not the target of this function.

The GTITC register sets the skipping function for the GTCNT counter overflow (GTPR compare match) interrupt (GPTn_OVF) and underflow interrupt (GPTn_UDF) and also sets whether to link the other interrupts and A/D conversion start requests with the GPTn_OVF/GPTn_UDF interrupt skipping function. Note the output disable request to POEG cannot be linked with the GPTn_OVF/GPTn_UDF interrupt skipping function. Additionally, if the interrupt skipping function is performed, the change in the status flag is also skipped.

The setting is invalid during the event count operation.

The setting is operated independently from the extended interrupt skipping by the GTEITC register.

ITLA bit (GTCCRA Register Compare Match/Input Capture Interrupt Link)

The ITLA bit specifies whether to link the GTCCRA compare match/input capture interrupt (GPTn_CCMPA) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLB bit (GTCCRB Register Compare Match/Input Capture Interrupt Link)

The ITLB bit specifies whether to link the GTCCRB compare match/input capture interrupt (GPTn_CCMPB) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLC bit (GTCCRC Register Compare Match Interrupt Link)

The ITLC bit specifies whether to link the GTCCRC compare match interrupt (GPTn_CMPC) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLD bit (GTCCRD Register Compare Match Interrupt Link)

The ITLD bit specifies whether to link the GTCCRD compare match interrupt (GPTn_CMPD) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLE bit (GTCCRE Register Compare Match Interrupt Link)

The ITLE bit specifies whether to link the GTCCRE compare match interrupt (GPTn_CMPE) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

ITLF bit (GTCCRF Register Compare Match Interrupt Link)

The ITLF bit specifies whether to link the GTCCRF compare match interrupt (GPTn_CMPF) with the GPTn_OVF/GPTn_UDF interrupt skipping function.

IVTC[1:0] bits (GPTn_OVF/GPTn_UDF Interrupt Skipping Function Select)

The IVTC[1:0] bits set the skipping function for the GTPR compare match (GTCNT counter overflow) interrupt (GPTn_OVF) and GTCNT counter underflow interrupt (GPTn_UDF).

IVTT[2:0] bits (GPTn_OVF/GPTn_UDF Interrupt Skipping Count Select)

The IVTT[2:0] bits set the skipping count for the GTPR compare match (GTCNT counter overflow) interrupt (GPTn_OVF) and GTCNT counter underflow interrupt (GPTn_UDF).

When modifying the IVTT[2:0] bits, first set the IVTC[1:0] bits to 00b.

ADTAL bit (GTADTRA Register A/D Conversion Start Request Link)

The ADTAL bit specifies whether to link the GTADTRA A/D conversion start request, which is generated in response to a compare match with the GTCNT counter and the GTADTRA register, with the GPTn_OVF/GPTn_UDF interrupt skipping function.

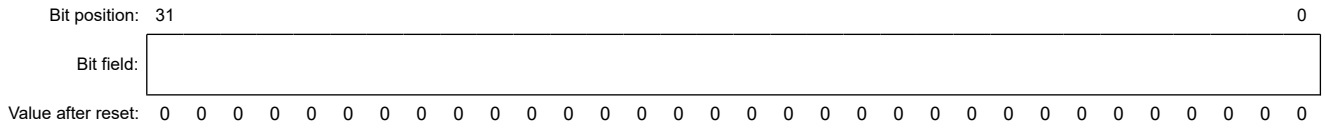
ADTBL bit (GTADTRB Register A/D Conversion Start Request Link)

The ADTBL bit specifies whether to link the GTADTRB A/D conversion start request, which is generated in response to a compare match with the GTCNT counter and the GTADTRB register, with the GPTn_OVF/GPTn_UDF interrupt skipping function.

20.2.19 GTCNT : General PWM Timer Counter

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x48

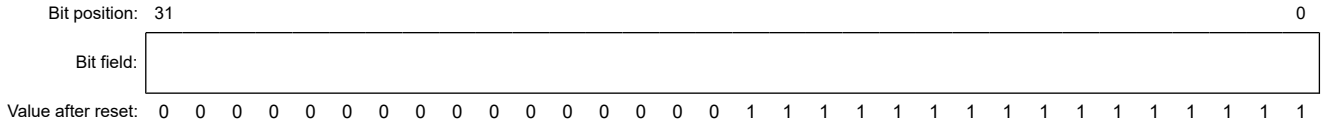


Bit	Symbol	Function	R/W
31:0	n/a	GTCNT is a 16-bit read/write counter for GPT16n (n = 0 to 3). GTCNT can only be written to after counting stops. Access in 8-bit unit to the GTCNT counter is prohibited, and it should be accessed in 16-bit or 32-bit unit. The upper 16 bits for access in 32-bit unit are always read as 0x0000, and writing to these bits is ignored. In saw waves or triangle waves, GTCNT must be set within the range of 0 ≤ GTCNT ≤ GTPR.	R/W

20.2.20 GTCCRk : General PWM Timer Compare Capture Register k (k = A to F)

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x4C (GTCCRA)
0x50 (GTCCRB)
0x54 (GTCCRC)
0x58 (GTCCRE)
0x5C (GTCCRD)
0x60 (GTCCRF)

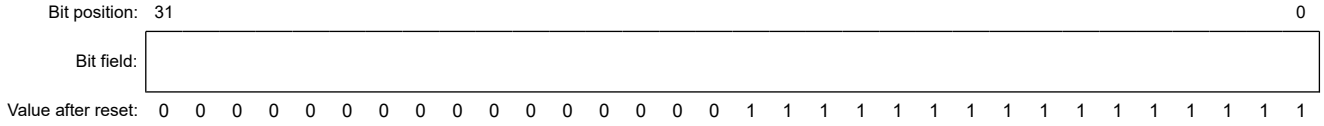


Bit	Symbol	Function	R/W
31:0	n/a	GTCCRk registers are read/write registers. The effective size of GTCCRk is the same as GTCNT (16 bits or 32 bits). Access in 8-bit unit to the GTCCRk register is prohibited, and it should be accessed in 16-bit or 32-bit unit. The upper 16 bits for access in 32-bit unit are always read as 0x0000, and writing to these bits is ignored. GTCCRA and GTCCRB are registers used for both output compare and input capture. GTCCRC and GTCCRE are compare match registers, and can also function as buffer registers for GTCCRA and GTCCRB. GTCCRD and GTCCRF are compare match registers, and can also function as buffer registers for GTCCRC and GTCCRE (double-buffer registers for GTCCRA and GTCCRB).	R/W

20.2.21 GTPR : General PWM Timer Cycle Setting Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x64

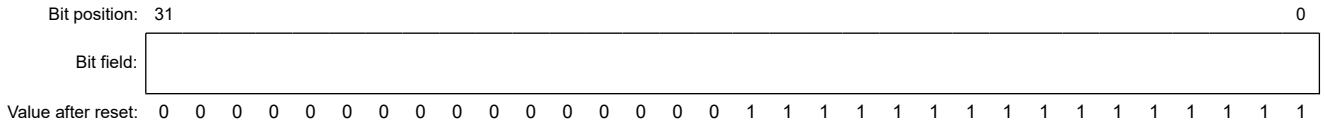


Bit	Symbol	Function	R/W
31:0	n/a	GTPR is a read/write register that sets the maximum count value of GTCNT. The effective size of GTPR is the same as GTCNT (16 bits or 32 bits). Access in 8-bit unit to the GTPR register is prohibited, and it should be accessed in 16-bit or 32-bit unit. The upper 16 bits for access in 32-bit unit are always read as 0x0000, and writing to these bits is ignored. The setting is invalid in the Saw-wave PWM mode 2. For Saw waves except the Saw-wave PWM mode 2, the value of (GTPR + 1) is the cycle. For triangle waves or complementary PWM mode, the value of (GTPR value × 2) is the cycle. In complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.	R/W

20.2.22 GTPBR : General PWM Timer Cycle Setting Buffer Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x68



Bit	Symbol	Function	R/W
31:0	n/a	GTPBR is a read/write register that functions as a buffer register for GTPR. The effective size of GTPBR is the same as GTCNT (16 bits or 32 bits). Access in 8-bit unit to the GTPBR register is prohibited, and it should be accessed in 16-bit or 32-bit unit. The upper 16 bits for access in 32-bit unit are always read as 0x0000, and writing to these bits is ignored. The setting is invalid in Saw-wave PWM mode 2. In complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.	R/W

20.2.23 GTPDBR : General PWM Timer Cycle Setting Double Buffer Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x6C

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The buffer register for the GTPBR register (double buffer register for the GTPR register). The effective size of GTPDBR is the same as GTCNT (16 bits or 32 bits). Access in 8-bit unit to the GTPDBR register is prohibited, and it should be accessed in 16-bit or 32-bit unit. The upper 16 bits for access in 32-bit unit are always read as 0x0000, and writing to these bits is ignored. The setting is invalid in Saw-wave PWM mode 2. In complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.	R/W

20.2.24 GTADTRk : A/D Conversion Start Request Timing Register k (k = A, B)

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x70 (GTADTRA)
0x7C (GTADTRB)

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	Set the timing of A/D conversion start request generation. The effective size of GTADTRk is the same as GTCNT (16 bits or 32 bits). Access in 8-bit unit to the GTADTRk register is prohibited, and it should be accessed in 16-bit or 32-bit unit. The upper 16 bits for access in 32-bit unit are always read as 0x0000, and writing to these bits is ignored. When the GTADTRk register value matches the GTCNT counter value, an A/D conversion start request is generated. In complementary PWM mode, A/D conversion start request is generated when the GTCNT counter of the master channel matches this register.	R/W

20.2.25 GTADTBRk : A/D Conversion Start Request Timing Buffer Register k (k = A, B)

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x74 (GTADTBRA)
0x80 (GTADTBRB)

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The buffer registers for the GTADTRk register The effective size of GTADTBRk is the same as GTCNT (16 bits or 32 bits). Access in 8-bit unit to the GTADTBRk register is prohibited, and it should be accessed in 16-bit or 32-bit unit. The upper 16 bits for access in 32-bit unit are always read as 0x0000, and writing to these bits is ignored.	R/W

20.2.26 GTADTDBRk : A/D Conversion Start Request Timing Double Buffer Register k (k = A, B)

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x78 (GTADTDBRA)
0x84 (GTADTDBRB)

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The buffer registers for the GTADTBRk register (double buffer registers for the GTADTRk register). The effective size of GTADTDBRk is the same as GTCNT (16 bits or 32 bits). Access in 8-bit unit to the GTADTDBRk register is prohibited, and it should be accessed in 16-bit or 32-bit unit. The upper 16 bits for access in 32-bit unit are always read as 0x0000, and writing to these bits is ignored.	R/W

20.2.27 GTDTCR : General PWM Timer Dead Time Control Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0x88

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

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

Bit	Symbol	Function	R/W
0	TDE	Negative-Phase Waveform Setting 0: GTCCRB is set without using GTDVU and GTDVD 1: GTDVU and GTDVD are used to set the compare match value for negative-phase waveform with dead time automatically in GTCCRB	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	TDBUE	GTDVU Register Buffer Operation Enable*1 0: GTDVU register buffer operation is disabled 1: GTDVU register buffer operation is enabled	R/W
5	TDBDE	GTDVD Register Buffer Operation Enable*1 0: GTDVD register buffer operation is disabled 1: GTDVD register buffer operation is enabled	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	TDFER	GTDVD Register Setting*1 0: GTDVU and GTDVD registers are set separately 1: The value written to GTDVU register is automatically set to GTDVD register	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is only available in GPT160 to GPT162.
In GPT163, this bit is read as 0. The write value should be 0.

GTDTCR enables automatic setting of a compare match value for negative-phase waveform with dead time. GPT has a dead time control function and GTDVU and GTDVD registers are used for setting dead time value.

This register is invalid in Saw-wave PWM mode 2 or complementary PWM mode.

The setting is invalid during the event count operation.

TDE bit (Negative-Phase Waveform Setting)

The TDE bit specifies whether to use GTDVU and GTDVD. When GTDVU and GTDVD are used, the compare match value for a negative-phase waveform with dead time obtained by the compare match value of a positive-phase waveform (GTCCRA) and the dead time value (GTDVU and GTDVD) is automatically set in GTCCRB.

The TDE bit setting is ignored in saw-wave PWM mode, and the GTCCRB is not automatic setting.

The GTCCRB value is automatically set and has the following upper and lower limit values. If the obtained GTCCRB value is not within the upper or lower limit, the following limit value is set in GTCCRB, and GTST.DTEF flag becomes 1. However, if the obtained GTCCRB value exceeds the upper limit in triangle-wave PWM mode, DTEF flag becomes 0.

- Triangle waves:
  Upper limit value: GTPR − 1
  Lower limit value: 1 in up-counting, 0 in down-counting
- Saw-wave one-shot pulse mode:
  Upper limit value: GTPR
  Lower limit value: 0.

TDBUE bit (GTDVU Register Buffer Operation Enable)

The TDBUE bit enables buffer operation with the GTDVU and GTDBU registers combined.

The timing of buffer transfer is at troughs in triangle-wave mode, and at overflows or underflows in saw-wave mode.

TDBDE bit (GTDVD Register Buffer Operation Enable)

The TDBDE bit enables buffer operation with the GTDVD and GTDBD registers combined.

The buffer transfer timing is the trough in triangle-wave mode, and at an overflow or underflow in saw-wave mode.

When this bit and the TDFER bit are set to 1 simultaneously, the TDFER bit setting is given priority.

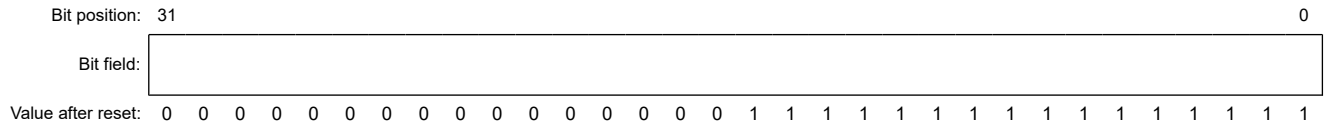
TDFER bit (GTDVD Register Setting)

The TDFER bit sets whether or not the value written to the GTDVU register is also set to the GTDVD register automatically.

20.2.28 GTDVk : General PWM Timer Dead Time Value Register k (k = U, D)

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x8C (GTDVU)
0x90 (GTDVD)

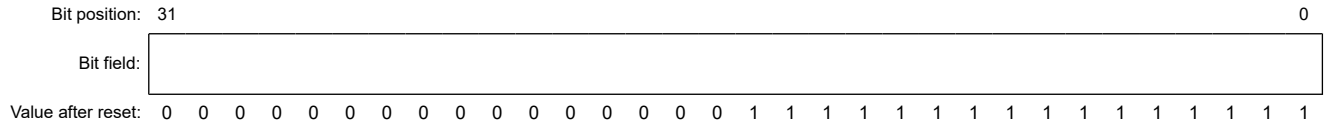


Bit	Symbol	Function	R/W
31:0	n/a	GTDVk is a read/write register that sets the dead time for generating PWM waveforms with dead time. The effective size of GTDVk is the same as GTCNT (16 bits or 32 bits). Access in 8-bit unit to the GTDVk register is prohibited, and it should be accessed in 16-bit or 32-bit unit. The upper 16 bits for access in 32-bit unit are always read as 0x0000, and writing to these bits is ignored. The setting is invalid in saw-wave PWM mode 2. In complementary PWM mode, the GTDVD register is invalid and the GTDVU register is used as the dead time value during both up-counting and down-counting. No matter which GTDVU register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time. In triangle-waves, the GTDVU register is used for up-counting and the GTDVD register is used for down-counting. In saw-waves, the GTDVU register controls the front dead time and the GTDVD register controls the rear dead time, regardless of whether the count is up or down. Setting a GTDVk value greater than or equal to GTPR is prohibited. In complementary PWM mode, set the GTDVU register to meet the following conditions: ● GTDVU > 0 ● GTDVU < GTPR/2 ● GTDVU + GTPR ≤ 0xffffffff When using the automatic dead time setting function, do not set a value that causes a change point of the waveform to exceed the count period. The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. When GTDVk is used, writing to GTCCRB is prohibited. Other than complementary PWM mode, when this register is set to 0, waveforms without dead time are output. When the GTDTCR.TDFER bit is 1, writing to the GTDVD register has no effect. At this time, when the GTDVD register is read, the value for the GTDVU register is read. While GPT is running, changing the GTDVk values is prohibited. To change GTDVk to a new value, stop the GPT with the CST bit in the GTCR register.	R/W

20.2.29 GTDBk : General PWM Timer Dead Time Buffer Register k (k = U, D)

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x94 (GTDBU)
0x98 (GTDBD)



Bit	Symbol	Function	R/W
31:0	n/a	The buffer register for the GTDVk register. The effective size of GTDBk is the same as GTCNT (16 bits or 32 bits). Access in 8-bit unit to the GTDBk register is prohibited, and it should be accessed in 16-bit or 32-bit unit. The upper 16 bits for access in 32-bit unit are always read as 0x0000, and writing to these bits is ignored.	R/W

Note: GTDBD is only available in GPT160 to GPT162.
In GPT163, GTDBD is read as 0x0000FFFF. The write value should be 0x0000FFFF.

20.2.30 GTSOS : General PWM Timer Output Protection Function Status Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)

Offset address: 0x9C

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

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

Bit	Symbol	Function	R/W
1:0	SOS[1:0]	Output Protection Function Status 0 0: Normal operation 0 1: Protected state (GTCCRA = 0 is set during transfer at trough or crest) 1 0: Protected state (GTCCRA ≥ GTPR is set during transfer at trough) 1 1: Protected state (GTCCRA ≥ GTPR is set during transfer at crest)	R
7:2	—	These bits are read as 0.	R
9:8	—	The read value is undefined.	R
31:10	—	These bits are read as 0.	R

The GTSOS register is a status register that indicates the status of the output protection function. The output protection function is enabled only when the dead time is automatically set (GTDTCR.TDE bit = 1) in triangle-wave mode.

SOS[1:0] bits (Output Protection Function Status)

The SOS[1:0] bits indicate the status of the output protection function in triangle-wave PWM mode. For details of the output protection function, see section 20.8.4. Output Protection Function for GTIOCnm Pin Output (n = 0 to 2, m = A, B).

20.2.31 GTSOTR : General PWM Timer Output Protection Function Temporary Release Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)

Offset address: 0xA0

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

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

Bit	Symbol	Function	R/W
0	SOTR	Output Protection Function Temporary Release 0: Protected state is not released 1: Protected state is released	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The GTSOTR register temporarily releases the protected state of GTIOCnB pin output (n = 0 to 3) when output protection has been set.

The protected state can be released only for the case of GTSOS.SOS[1:0] bits are 10b (protected state in which GTCCRA register ≥ GTPR register has occurred during transfer at trough). The protected state cannot be released for any other case.

SOTR bit (Output Protection Function Temporary Release)

The SOTR bit sets whether to temporarily release the protected state of the GTIOCnB pin output in an output protected state.

After the SOTR bit is set to 1, the output protection function is canceled from the first trough. After the SOTR bit is set to 0, output protection is resumed from the first trough.

20.2.32 GTADSMR : General PWM Timer A/D Conversion Start Request Signal Monitoring Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0xA4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	ADSMEN1	—	—	—	—	—	—	ADSMS1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADSMEN0	—	—	—	—	—	—	ADSMS0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ADSMS0[1:0]	A/D Conversion Start Request Signal Monitor 0 Selection 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	ADSMEN0	A/D Conversion Start Request Signal Monitor 0 Output Enabling 0: Output of A/D conversion start request signal monitor 0 is disabled 1: Output of A/D conversion start request signal monitor 0 is enabled	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
17:16	ADSMS1[1:0]	A/D Conversion Start Request Signal Monitor 1 Selection 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
24	ADSMEN1	A/D Conversion Start Request Signal Monitor 1 Output Enabling 0: Output of A/D conversion start request signal monitor 1 is disabled 1: Output of A/D conversion start request signal monitor 1 is enabled	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The GTADSMR register is used to control monitors for the A/D conversion start request signal that is synchronized with a frame period.

ADSMSk[1:0] bits (A/D Conversion Start Request Signal Monitor k Selection) (k = 0, 1)

The ADSMSk[1:0] bits are used to select A/D conversion start request signal synchronized with a frame period which is monitored by the GTASMk pin.

In triangle-wave PWM mode or complementary PWM mode, the following settings are prohibited:

- Set ADSMSk[1:0] bits to 00b (A/D conversion start request during up-counting) when GTADTRA = 0
- Set ADSMSk[1:0] bits to 10b (A/D conversion start request during up-counting) when GTADTRB = 0
- Set ADSMSk[1:0] bits to 01b (A/D conversion start request during down-counting) when GTADTRA = GTPR
- Set ADSMSk[1:0] bits to 11b (A/D conversion start request during down-counting) when GTADTRB = GTPR

ADSMENk bit (A/D Conversion Start Request Signal Monitor k Output Enabling) (k = 0, 1)

The ADSMENk bit enables or disables the monitor output to the GTADSMk pin.

When the output is disabled, the GTADSMk pin goes to the low level.

When the bit is 1, the signal on the GTADSMk pin goes to the high level on assertion of the signal to request to the start of A/D conversion selected by the ADSMSk[1:0] bits and returns to the low level at the end of the current cycle of the timer for the channel that generated the given signal to request the start of A/D conversion. When the counter stops, the value when the counter stopped is retained for output. Set the ADSMENk bit to 0 to output the low level.

When a signal to request the start of A/D conversion is generated at the end of a timer period, the generation of this signal has priority in terms of monitoring output and the output remains at the high level till the end of the next period.

When the output of the same A/D conversion start request signal monitoring output is enabled for multiple channels, ORed signals will be output from the GPT.

20.2.33 GTEITC : General PWM Timer Extended Interrupt Skipping Counter Control Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0xA8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EITCNT2[3:0]				EITCNT2IV[3:0]				EIVTT2[3:0]				—	—	EIVTC2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EITCNT1[3:0]				—	—	—	—	EIVTT1[3:0]				—	—	EIVTC1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	EIVTC1[1:0]	Extended Interrupt Skipping Counter 1 Count Source Select 0 0: Not counted (not skipped) 0 1: Counting both at overflow or underflow in saw-wave mode, and counting crests in triangle-wave mode or complementary PWM mode 1 0: Counting both at overflow or underflow in saw-wave mode, and counting troughs in triangle-wave mode or complementary PWM mode 1 1: Counting both at overflow or underflow in saw-wave mode, and counting both crests and troughs in triangle-wave mode or complementary PWM mode	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	EIVTT1[3:0]	Extended Interrupt Skipping 1 Skipping Count Setting Skipping count for the extended interrupt skipping 1	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
15:12	EITCNT1[3:0]	Extended Interrupt Skipping Counter 1	R

Bit	Symbol	Function	R/W
17:16	EIVTC2[1:0]	Extended Interrupt Skipping Counter 2 Count Source select 0 0: Not counted (not skipped) 0 1: Counting both at overflow or underflow in saw-wave mode, and counting crests in triangle-wave mode or complementary PWM mode 1 0: Counting both at overflow or underflow in saw-wave mode, and counting troughs in triangle-wave mode or complementary PWM mode 1 1: Counting both at overflow or underflow in saw-wave mode, and counting both crests and troughs in triangle-wave mode or complementary PWM mode	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
23:20	EIVTT2[3:0]	Extended Interrupt Skipping 2 Skipping Count Setting Skipping count for the extended interrupt skipping 2	R/W
27:24	EITCNT2IV[3:0]	Extended Interrupt Skipping Counter 2 Initial Value	R/W*1
31:28	EITCNT2[3:0]	Extended Interrupt Skipping Counter 2	R

Note 1. The EITCNT2IV[3:0] bits are only writable when the value other than 00b is written to the EIVTC2[1:0] bits that have been 00b.

GTEITC register sets the extended interrupt skipping function to skip the interrupts, A/D conversion start requests, and buffer transfers independently by counting at overflow and underflow in the GTCNT counter.

The setting is operated independently from the interrupt skipping by the GTITC register or the GTADCMSC register.

The setting is invalid during the event count operation.

Access in 8-bit unit to GTEITC is prohibited.

EIVTCk[1:0] bits (Extended Interrupt Skipping Counter k Count Source Select) (k = 1, 2)

The EIVTCk[1:0] bits select the way of counting for the extended interrupt skipping counter k.

Setting only with these bits does not skip the interrupts, A/D conversion start requests, and buffer transfers. Skipping function for the interrupt, A/D conversion start request, and buffer transfer, all of which are a target of skipping, is set individually with the GTEITLI1, GTEITLI2, and GTEITLB registers.

EIVTTk[3:0] bits (Extended Interrupt Skipping k Skipping Count Setting) (k = 1, 2)

The EIVTTk[3:0] bits set the skipping count for the extended interrupt skipping k.

A count for the period with continuous skipping is set as a skipping count, where a period is from a generation of a count source selected by the EIVTCk[1:0] bits to the next generation of the count source.

When the count source is generated while the EIVTTk[3:0] bits match the EITCNTk[3:0] bits, the EITCNTk[3:0] bits are cleared.

When these bits are 0x0, skipping is not performed.

EITCNT1[3:0] bits (Extended Interrupt Skipping Counter 1)

The EITCNT1[3:0] bits are the extended interrupt skipping counter 1.

The counting is incremented by 1 every time a count source (overflow/underflow/crest/trough) selected by the EIVTC1[1:0] bits is generated.

Counting is performed periodically within the range between 0 and the EIVTT1[3:0] bits.

Even if the GTCNT counter is stopped, the value is not cleared, and value at stop of the GTCNT counter is retained.

[Clearing conditions]

- 00b is written to the EIVTC1[1:0] bits.
- 0x0 is written to the EIVTT1[3:0] bits.
- A count source (overflow/underflow/crest/trough) selected by the EIVTC1[1:0] bits is generated when the extended interrupt skipping 1 skipping count set by the EIVTT1[3:0] bits match the value for the EITCNT1[3:0]

EITCNT2IV[3:0] bits (Extended Interrupt Skipping Counter 2 Initial Value)

The EITCNT2IV[3:0] bits are the value of the initial value for the extended interrupt skipping counter 2.

Writing to the EITCNT2IV[3:0] bits is performed only when the writing value to the EIVTC2[1:0] bits are other than 00b and when the GTEITC register is written by the access of upper 16 bits or 32 bits while the EITCNT2[3:0] bits are set not to count (EIVTC2[1:0] bits are 00b). When the EITCNT2IV[3:0] bits are written, the value written to the EITCNT2IV[3:0] bits is written to the EITCNT2[3:0] bits simultaneously.

The writing to the EITCNT2IV[3:0] bits are ignored when the EITCNT2[3:0] bits are set to count (EIVTC2[1:0] bits are other than 00b) or perform the setting of not to count (00b is written to the EIVTC2[1:0] bits).

The EITCNT2IV[3:0] bits are not reset by the writing 00b to the EIVTC2[1:0] bits.

EITCNT2[3:0] bits (Extended Interrupt Skipping Counter 2)

The EITCNT2[3:0] bits are the extended interrupt skipping counter 2.

The counting is incremented by 1 every time a count source (overflow/underflow/crest/trough) selected by the EIVTC2[1:0] bits is generated.

Counting is performed periodically within the range between 0 and the EIVTT2[3:0] bits.

Even if the GTCNT counter is stopped, the value is not cleared, and value at stop of the GTCNT counter is retained.

Setting of the initial value for the EITCNT2[3:0] bits are performed only when the writing value to the EIVTC2[1:0] bits are other than 00b and when the GTEITC register is written by the access of upper 16 bits or 32 bits while the extended interrupt skipping counter 2 is set as not to count (EIVTC2[1:0] bits are 00b).

When the initial value is set, the written value to the EITCNT2IV[3:0] bits is written to the EITCNT2[3:0] bits as the initial value.

[Clearing condition]

- 00b is written to the EIVTC2[1:0] bits.
- 0x0 is written to the EIVTT2[3:0] bits.
- The value other than 00b is written to the EIVTC2[1:0] bits and the 0x0 is written to the EITCNT2IV[3:0] bits simultaneously while 00b is set to the EIVTC2[1:0] bits.
- A count source (overflow/underflow/crest/trough) selected by the EIVTC2[1:0] bits is generated when the extended interrupt skipping 2 skipping count set by the EIVTT2[3:0] bits match the value for the EITCNT2[3:0].

20.2.34 GTEITLI1 : General PWM Timer Extended Interrupt Skipping Setting Register 1

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0xAC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	EITLU[2:0]			—	EITLV[2:0]			—	EITLF[2:0]			—	EITLE[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	EITLA[2:0]	GTCCRA Register Compare Match/Input Capture Interrupt Extended Skipping Function Select See Table 20.6.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	EITLB[2:0]	GTCCRB Register Compare Match/Input Capture Interrupt Extended Skipping Function Select See Table 20.6.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
10:8	EITLC[2:0]	GTCCRC Register Compare Match Interrupt Extended Skipping Function Select See Table 20.6.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
14:12	EITLD[2:0]	GTCCRD Register Compare Match Interrupt Extended Skipping Function Select See Table 20.6.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
18:16	EITLE[2:0]	GTCCRE Register Compare Match Interrupt Extended Skipping Function Select See Table 20.6.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	EITLF[2:0]	GTCCRF Register Compare Match Interrupt Extended Skipping Function Select See Table 20.6.	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
26:24	EITLV[2:0]	Overflow Interrupt Extended Skipping Function Select See Table 20.6.	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	EITLU[2:0]	Underflow Interrupt Extended Skipping Function Select See Table 20.6.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The GTEITLI1 register sets the extended skipping function for interrupts such as compare match/input capture, overflow and underflow.

Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.

The setting is operated independently from the interrupt skipping by the GTITC register or the GTADCMSC register.

The setting is invalid during the event count operation.

EITLm[2:0] bits (GTCCRm Register Compare Match/Input Capture Interrupt Extended Skipping Function Select) (m = A, B)

The EITLm[2:0] bits select the extended interrupt skipping function to skip the compare match/input capture interrupt (GPTn_CCMPm) in the GTCCRm register. See Table 20.6.

EITLx[2:0] bits (GTCCRx Register Compare Match Interrupt Extended Skipping Function Select) (x = C, D, E, F)

The EITLx[2:0] bits select the extended interrupt skipping function to skip the compare match interrupt (GPTn_CMPx) in the GTCCRx register. See Table 20.6.

EITLV[2:0] bits (Overflow Interrupt Extended Skipping Function Select)

The EITLV[2:0] bits select the extended interrupt skipping function to skip the interrupt at overflow(GPTn_OVF). See Table 20.6.

EITLU[2:0] bits (Underflow Interrupt Extended Skipping Function Select)

The EITLU[2:0] bits select the extended interrupt skipping function to skip the interrupt at underflow (GPTn_UDF). See Table 20.6.

Table 20.6 Setting the Function select for the GTEITLI1 (1 of 2)

EITLy[2:0]	Function
0 0 0	Do not perform an extended interrupt skipping
0 0 1	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 is other than 0 (An interrupt is output in the period of the EITCNT1[3:0] bits = 0)
0 1 0	Skip an interrupt in the period when the value for the extended interrupt skipping counter 2 is other than 0 (An interrupt is output in the period of the EITCNT2[3:0] bits = 0)

Table 20.6 Setting the Function select for the GTEITLI1 (2 of 2)

EITLy[2:0]	Function
0 1 1	Skip an interrupt in the period the value for the extended interrupt skipping counter 1 or 2 is other than 0 (An interrupt is output in the period of the EITCNT1[3:0] bits = 0 and the EITCNT2[3:0] bits = 0)
1 0 0	Setting prohibited
1 0 1	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count (An interrupt is output in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits)
1 1 0	Skip an interrupt in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count (An interrupt is output in the period of the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)
1 1 1	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count (An interrupt is output in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits and the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)

Note:
- y = A, B, C, D, E, F, V, U
- When the intended skipping counter is set as not to count (the EIVTCk[1:0] bits = 00b or the EIVTTk[3:0] bits = 0x0), skipping is not performed.(k = 1, 2)
- When the EITLy[2:0] bits are set to 011b or 111b, and when one of the skipping counter 1 or 2 is set as not to count, skipping is not performed.

20.2.35 GTEITLI2 : General PWM Timer Extended Interrupt Skipping Setting Register 2

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0xB0

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

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

Bit	Symbol	Function	R/W
2:0	EADTAL[2:0]	GTADTRA Register A/D Conversion Start Request Extended Skipping Function Select See Table 20.7.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	EADTBL[2:0]	GTADTRB Register A/D Conversion Start Request Extended Skipping Function Select See Table 20.7.	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

The GTEITLI2 register sets the extended skipping function for A/D conversion start requests.

Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.

The setting is operated independently from the interrupt skipping by the GTITC register or the GTADCMSC register.

The setting is invalid during the event count operation.

EADTmL[2:0] bits (GTADTRm Register A/D Conversion Start Request Extended Skipping Function Select) (m = A, B)

The EADTmL[2:0] bits select the extended skipping function to skip A/D conversion start requests for the compare match in the GTADTRm register. See Table 20.7.

Table 20.7 Setting the function select for the GTEITLI2 Register

EADTmL[2:0]	Function
0 0 0	Do not perform an extended interrupt skipping
0 0 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 is other than 0 (An A/D conversion start request is output in the period of the EITCNT1[3:0] bits = 0)
0 1 0	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 2 is other than 0 (An A/D conversion start request is output in the period of the EITCNT2[3:0] bits = 0)
0 1 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 or 2 is other than 0 (An A/D conversion start request is output in the period of the EITCNT1[3:0] bits = 0 and the EITCNT2[3:0] bits = 0)
1 0 0	Setting prohibited
1 0 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count (An A/D conversion start request is output in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits)
1 1 0	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count (An A/D conversion start request is output in the period of the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)
1 1 1	Skip an A/D conversion start request in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count (An A/D conversion start request is output in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits and the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)

Note:
- m = A, B
- When the intended skipping counter is set as not to count (the EIVTCk[1:0] bits = 00b or the EIVTTk[3:0] bits = 0x0), skipping is not performed. (k = 1, 2)
- When the EADTmL[2:0] bits are set to 011b or 111b, and when one of the skipping counter 1 or 2 is set as not to count, skipping is not performed.

20.2.36 GTEITLB : General PWM Timer Extended Buffer Transfer Skipping Setting Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0xB4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	EBTLDVD[2:0]			—	EBTLDVU[2:0]			—	EBTLADB[2:0]			—	EBTLADA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	EBTLCA[2:0]	GTCCRA Register Buffer Transfer Extended Skipping Function Select See Table 20.8.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	EBTLCB[2:0]	GTCCRB Register Buffer Transfer Extended Skipping Function Select See Table 20.8.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	EBTLPR[2:0]	GTPR Register Buffer Transfer Extended Skipping Function Select See Table 20.8.	R/W*1
15:11	—	These bits are read as 0. The write value should be 0.	R/W
18:16	EBTLADA[2:0]	GTADTRA Register Buffer Transfer Extended Skipping Function Select See Table 20.8.	R/W

Bit	Symbol	Function	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	EBTLADB[2:0]	GTADTRB Register Buffer Transfer Extended Skipping Function Select See Table 20.8.	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
26:24	EBTLDVU[2:0]	GTDVU Register Buffer Transfer Extended Skipping Function Select*2 See Table 20.8.	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	EBTLDVD[2:0]	GTDVD Register Buffer Transfer Extended Skipping Function Select*2 See Table 20.8.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. In complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.
Note 2. This bit is only available in GPT160 to GPT162.
In GPT163, this bit is read as 0. The write value should be 0.

The GTEITLB register sets the extended skipping function for buffer transfers.

Setting only this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.

The setting is operated independently from the interrupt skipping by the GTITC register.

The buffer transfer from the GTOLBR register to the GTIOR.GTIOA[4:0], GTIOB[4:0] bits is not target of the extended buffer transfer skipping function.

The setting is invalid during the event count operation.

EBTLCA[2:0] bits (GTCCRA Register Buffer Transfer Extended Skipping Function Select)

The EBTLCA[2:0] bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTCCRA, GTCCRC, GTCCRD registers and a temporary register A) in the GTCCRA register. See Table 20.8.

An extended skipping of buffer transfers in the GTCCRA register is valid for forcible buffer transfers by the GTBER.CCRSWT bit while the count operation is stopped. Forcible buffer transfers in the GTCCRA register should be performed in the condition of not performing the extended buffer transfer skipping.

The buffer transfer between the GTCCRC, GTCCRE, and GTCCRA in complementary PWM mode cannot be skipped.

EBTLCB[2:0] bits (GTCCRB Register Buffer Transfer Extended Skipping Function Select)

The EBTLCB[2:0] bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTCCRB, GTCCRE, GTCCRF registers and a temporary register B) in the GTCCRB register. See Table 20.8.

An extended skipping of buffer transfers in the GTCCRB register is valid for forcible buffer transfers by the GTBER.CCRSWT bit while the count operation is stopped. Forcible buffer transfers in the GTCCRB register should be performed in the condition of not performing the extended buffer transfer skipping.

EBTLPR[2:0] bits (GTPR Register Buffer Transfer Extended Skipping Function Select)

The EBTLPR[2:0] bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTPR, GTPBR, and GTPDBR registers) in the GTPR register. See Table 20.8.

If the buffer transfer of GTPR is skipped in complementary PWM mode, GTEITC setting of slave channel should be matched to the master channel so that the buffer transfer timing of the slave channel matches the master channel.

EBTLADm[2:0] bits (GTADTRm Register Buffer Transfer Extended Skipping Function Select) (m = A, B)

The EBTLADm[2:0] bits select the extended buffer transfer skipping function to skip buffer transfers (transfers among the GTADTRm, GTADTBRm, and GTADTDBRm registers) in the GTADTRm register. See Table 20.8.

EBTLDVm[2:0] bits (GTDVm Register Buffer Transfer Extended Skipping Function Select) (m = U, D)

The EBTLDVm[2:0] bits select the extended buffer transfer skipping function to skip buffer transfers (transfers between the GTDVm and GTDBm registers) in the GTDVm register. See Table 20.8.

Table 20.8 Setting the Function Select for the GTEITLB Register

EBTLx[2:0] bits	Function
0 0 0	Do not perform an extended interrupt skipping
0 0 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 is other than 0 (Buffer is transferred in the period of the EITCNT1[3:0] bits = 0)
0 1 0	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 2 is other than 0 (Buffer is transferred in the period of the EITCNT2[3:0] bits = 0)
0 1 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 or 2 is other than 0 (Buffer is transferred in the period of the EITCNT1[3:0] bits = 0 and the EITCNT2[3:0] bits = 0)
1 0 0	Setting prohibited
1 0 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count (Buffer is transferred in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits)
1 1 0	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count (Buffer is transferred in the period of the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)
1 1 1	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count (Buffer is transferred in the period of the EITCNT1[3:0] bits = the EIVTT1[3:0] bits and the EITCNT2[3:0] bits = the EIVTT2[3:0] bits)

Note:
- x = CA, CB, PR, ADA, ADB, DVU, DVD
- When the intended skipping counter is set as not to count (EIVTCk[1:0] bits = 00b or EIVTTk[3:0] bits = 0x0), skipping is not performed (k = 1, 2).
- When the EBTLx[2:0] bits are set to 011b or 111b, and when one of the skipping counter 1 or 2 is set as not to count, skipping is not performed.

20.2.37 GTPC : General PWM Timer Period Count Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0xBC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	PCNT[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
0	PCEN	Period Count Function Enable 0: Period count function is disabled 1: Period count function is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	ASTP	Automatic Stop Function Enable 0: Automatic stop function is disabled 1: Automatic stop function is enabled	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
27:16	PCNT[11:0]	Period Counter Counter for the number of period	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The GTPC register counts the number of period.

PCEN bit (Period Count Function Enable)

The PCEN bit enables or disables period count function.

Writing is available when counting is both in progress and stopped.

When 1 is written to either the GTSECR.SPCE bit or the GTSECR.SPCD bit, the value is simultaneously set to the PCEN bit in the channels set to 1 by the GTSECSR register.

ASTP bit (Automatic Stop Function Enable)

The ASTP bit enables or disables the GTCNT counter automatic stopping after finishing counting the number of period.

When the PCEN bis is 0, writing is available.

When the PCEN bit is 1, writing is disabled.

When the PCEN bit is 1, the ASTP bit is 1, and the PCNT counter is stopped at PCNT = 0, the GTCNT counter is also stopped. When the ASTP bit is 0, the GTCNT counter continues to count.

PCNT[11:0] bits (Period Counter)

The PCNT[11:0] counter count the number of period.

When the PCEN bit is 0, writing the number of period is available.

When the PCEN bit is 1, writing is disabled, and down-counting is performed at the end of period. In saw-wave mode, the end of period refers to overflow, underflow, or counter clearing. In triangle-wave mode or complementary PWM mode, it refers to trough.

When the PCNT[11:0] counter is 1 at the end of period, it becomes 0 and counting is stopped.

When the GTCNT counter is stopped while period count function is enabled, the PCNT[11:0] counter keeps its value. When the GTCNT counter restarts counting and the PCEN bit is 1, the PCNT[11:0] counter restarts down-counting from the hold value.

When the PCEN bit is changed from 0 to 1 while the PCNT[11:0] counter is 0 and the ASTP bit is 1, the GTCNT counter is stopped at the count clock immediately after that.

20.2.38 GTADCMSC : General PWM Timer A/D Conversion Start Request Compare Match Skipping Control Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0xC0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ADCMSCNT2[3:0]				ADCMSCNT2IV[3:0]				ADCMST2[3:0]				—	—	ADCMSC2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADCMSCNT1[3:0]				ADCMSCNT1IV[3:0]				ADCMST1[3:0]				—	—	ADCMSC1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ADCMSC1[1:0]	A/D Conversion Start Request Compare Match Skipping Counter 1 Count Source Select 0 0: Not counted (not skipped) 0 1: Counting GTADTRA register compare match 1 0: Counting GTADTRB register compare match 1 1: Counting both GTADTRA register compare match and GTADTRB register compare match	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	ADCMST1[3:0]	A/D Conversion Start Request Compare Match Skipping 1 Skipping Count Setting Skipping count for the A/D conversion start request compare match skipping 1.	R/W

Bit	Symbol	Function	R/W
11:8	ADCMSCNT1IV[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 1 Initial Value	R/W*1
15:12	ADCMSCNT1[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 1	R
17:16	ADCMSC2[1:0]	A/D Conversion Start Request Compare Match Skipping Counter 2 Count Source Select 0 0: Not counted (not skipped) 0 1: Counting GTADTRA register compare match 1 0: Counting GTADTRB register compare match 1 1: Counting both GTADTRA register compare match and GTADTRB register compare match	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
23:20	ADCMST2[3:0]	A/D Conversion Start Request Compare Match Skipping 2 Skipping Count Setting Skipping count for the A/D conversion start request compare match skipping 2	R/W
27:24	ADCMSCNT2IV[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 2 Initial Value	R/W*2
31:28	ADCMSCNT2[3:0]	A/D Conversion Start Request Compare Match Skipping Counter 2	R

Note 1. Writing is possible only when the ADCMSC1[1:0] bits are 00b and a value other than 00b is written to the ADCMSC1[1:0] bits.
Note 2. Writing is possible only when the ADCMSC2[1:0] bits are 00b and a value other than 00b is written to the ADCMSC2[1:0] bits.

The GTADCMSC register is a register that controls the skipping counter of the A/D conversion start request compare match skipping function that counts compare matches of GTADTRA register and GTADTRB register, and skipping A/D conversion start request and buffer transfer independently.

This register setting is operated independently from the interrupt skipping by the GTITC register or GTEITC register.

Access in 8-bit unit to GTADCMSC is prohibited.

ADCMSCk[1:0] bits (A/D Conversion Start Request Compare Match Skipping Counter k Count Source Select) (k = 1, 2)

The ADCMSCk[1:0] bits select the way of counting for the A/D conversion start request compare match skipping counter k.

Setting only with these bits does not skip the A/D conversion start requests and buffer transfers. Skipping function for the A/D conversion start request and buffer transfer which are a target of skipping is set individually with the GTADCMSS register.

ADCMSTk[3:0] bits (A/D Conversion Start Request Compare Match Skipping k Skipping Count Setting) (k = 1, 2)

The ADCMST2[3:0] bits set the skipping count for the A/D conversion start request compare match skipping k.

A count for the period with continuous skipping is set as a skipping count, where a period is from a generation of a count source selected by the ADCMSCk[1:0] bits to the next generation of the count source.

When the count source is generated while the ADCMSTk[3:0] bits match the ADCMSCNTk[3:0] bits, the ADCMSCNTk[3:0] bits are cleared.

When these bits are 0x0, skipping is not performed.

ADCMSCNTkIV[3:0] bits (A/D Conversion Start Request Compare Match Skipping Counter k Initial Value) (k = 1, 2)

The ADCMSCNTkIV[3:0] bits are the value of the initial value for the A/D conversion start request compare match skipping counter k.

Writing to the ADCMSCNTkIV[3:0] is performed only when the writing value to the ADCMSCk[1:0] bits are other than 00b and when the GTADCMSC register is written by the access of 16 bits or 32 bits while the ADCMSCNTk[3:0] bits are set not to count (ADCMSCk[1:0] bits are 00b). When the ADCMSCNTkIV[3:0] bits are written, the value written to the ADCMSCNTkIV[3:0] bits is written to the ADCMSCNTk[3:0] bits simultaneously.

The writing to the ADCMSCNTkIV[3:0] bits are ignored when the ADCMSCNTk[3:0] bits are set to count (ADCMSCk[1:0] bits are other than 00b) or perform the setting of not to count (00b is written to the ADCMSCk[1:0] bits).

The ADCMSCNTkIV[3:0] bits are not reset by the writing 00b to the ADCMSCk[1:0] bits.

ADCMSCNTk[3:0] bits (A/D Conversion Start Request Compare Match Skipping Counter k) (k = 1, 2)

The ADCMSCNT1[3:0] bits are the A/D conversion start request compare match skipping counter k.

The counting is incremented by 1 every time a count source selected by the ADCMSCk[1:0] bits is generated.

Counting is performed periodically within the range between 0 and the ADCMSTk[3:0] bits.

Even if the GTCNT counter is stopped, the value is not cleared, and value at stop of the GTCNT counter is retained.

Setting of the initial value for the ADCMSCNTk[3:0] bits are performed only when the writing value to the ADCMSCk[1:0] bits are other than 00b and when the GTADCMSC register is written by the access of 16 bits or 32 bits while the A/D conversion start request compare match skipping counter k is set as not to count (ADCMSCk[1:0] bits are 00b).

When the initial value is set, the written value to the ADCMSCNTkIV[3:0] bits is written to the ADCMSCNTk[3:0] bits as the initial value.

[Clearing condition]

- 00b is written to the ADCMSCk[1:0] bits.
- 0x0 is written to the ADCMSTk[3:0] bits.
- The value other than 00b is written to the ADCMSCk[1:0] bits and the 0x0 is written to the ADCMSCNTkIV[3:0] bits simultaneously while 00b is set to the ADCMSCk[1:0] bits.
- A count source selected by the ADCMSCk[1:0] bits is generated when the extended interrupt skipping 2 skipping count set by the ADCMSTk[3:0] bits match the value for the ADCMSCNTk[3:0].

20.2.39 GTADCMSS : General PWM Timer A/D Conversion Start Request Compare Match Skipping Setting Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0xC4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	ADCMBSB[2:0]			—	ADCMBSA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
2:0	ADCMSAL[2:0]	GTADTRA Register A/D Conversion Start Request Compare Match Skipping Function Select See Table 20.9.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	ADCMSBL[2:0]	GTADTRB Register A/D Conversion Start Request Compare Match Skipping Function Select See Table 20.9.	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W
18:16	ADCMBSA[2:0]	GTADTRA Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select See Table 20.10.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	ADCMBSB[2:0]	GTADTRB Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select See Table 20.10.	R/W

Bit	Symbol	Function	R/W
31:23	—	These bits are read as 0. The write value should be 0.	R/W

The GTADCMSS register is a register that selects the A/D conversion start request compare match skipping function or the GTADTRm (m = A, B) register buffer transfer by A/D conversion start request compare match skipping function.

Setting only with this register does not perform skipping. The GTADCMSC register should be set so that a corresponding A/D conversion start request compare match skipping counter operates the counting.

This register setting is operated independently from the interrupt skipping by the GTITC register or GTEITC register.

ADCMSmL[2:0] bits (GTADTRm Register A/D Conversion Start Request Compare Match Skipping Function Select) (m = A, B)

The ADCMSmL[2:0] bits select the A/D conversion start request compare match skipping function of the GTADTRm (m = A, B) register.Table 20.9.

ADCMBSm[2:0] bits (GTADTRm Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select) (m = A, B)

The ADCMBSm[2:0] bits select the GTADTRm (m = A, B) register buffer transfer by A/D conversion start request compare match skipping function which skips the buffer transfer of GTADTRm (m = A, B) register(Transfer between GTADTRm register, GTADTBRm register, GTADTDBRm register). See Table 20.10.

Table 20.9 Setting of the GTADTRm Register A/D Conversion Start Request Compare Match Skipping Function Select bit (m = A ,B)

ADCMSmL[2:0]	Function
0 0 0	Do not perform an A/D conversion start request compare match skipping
0 0 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than 0 (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = 0)
0 1 0	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than 0 (An A/D conversion start request is output in the period of the ADCMSCNT2[3:0] bits = 0)
0 1 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than 0 (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = 0 and the ADCMSCNT2[3:0] bits = 0)
1 0 0	Setting prohibited
1 0 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than the skipping count (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits)
1 1 0	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than the skipping count (An A/D conversion start request is output in the period of the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)
1 1 1	Skip an A/D conversion start request in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than the skipping count (An A/D conversion start request is output in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits and the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)

Note:
- m = A, B
- When the intended skipping counter is set as not to count (the ADCMSCk[1:0] bits = 00b or the ADCMSTk[3:0] bits = 0x0), skipping is not performed (k = 1, 2).
- When the ADCMSmL[2:0] bits are set to 011b or 111b, and when one of the skipping counters is set as not to count, skipping is not performed.

Table 20.10 Setting of the GTADTRm Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select bit (m = A, B) (1 of 2)

ADCMBSm[2:0]	Function
0 0 0	Do not perform an GTADTRm register buffer transfer by A/D conversion start request compare match skipping
0 0 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than 0 (Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = 0)

Table 20.10 Setting of the GTADTRm Register Buffer Transfer by A/D Conversion Start Request Compare Match Skipping Function Select bit (m = A, B) (2 of 2)

ADCMBSm[2:0]	Function
0 1 0	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than 0 (Buffer is transferred in the period of the ADCMSCNT2[3:0] bits = 0)
0 1 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than 0 (Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = 0 and the ADCMSCNT2[3:0] bits = 0)
1 0 0	Setting prohibited
1 0 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 is other than the skipping count (Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits)
1 1 0	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 2 is other than the skipping count (Buffer is transferred in the period of the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)
1 1 1	Skip a buffer transfer in the period when the value for the A/D conversion start request compare match skipping counter 1 or 2 is other than the skipping count (A Buffer is transferred in the period of the ADCMSCNT1[3:0] bits = the ADCMST1[3:0] bits and the ADCMSCNT2[3:0] bits = the ADCMST2[3:0] bits)

Note:
- m = A, B
- When the intended skipping counter is set as not to count (the ADCMSCk[1:0] bits = 00b or the ADCMSTk[3:0] bits = 0x0), skipping is not performed. (k = 1, 2)
- When the ADCMSmL[2:0] bits are set to 011b or 111b, and when one of the skipping counters is set as not to count, skipping is not performed.

20.2.40 GTSECSR : General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0xD0

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	SECSEL3	SECSEL2	SECSEL1	SECSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SECSEL0	Channel 0 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
1	SECSEL1	Channel 1 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
2	SECSEL2	Channel 2 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
3	SECSEL3	Channel 3 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The GTSECSR register selects an intended channel n (n = 0 to 3) for updating an operation enable bit by the GTSECR register. A bit position for the GTSECSR register indicates a channel number. The GTSECSR register of each channel is a common register, and writing 1 to a bit in the GTSECSR register in any channel and updating it changes a channel, related to the position of the bit written with 1 by the GTSECSR register, to be simultaneously controlled of the operation enable bit by the GTSECR register.

Access in 8-bit or 16-bit unit to GTSECSR is prohibited, and it should be accessed in 32-bit unit.

SECSELn bit (Operation Enable Bit Simultaneous Control Channel Select) (n = 0 to 3)

The SECSELn bit enables or disables the simultaneous control of operation enable in channel n.

When this bit is set to 1, simultaneous control is enabled, and is disabled when the bit is 0.

20.2.41 GTSECR : General PWM Timer Operation Enable Bit Simultaneous Control Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0xD4

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

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	SBDDD	SBDAD	SBDPD	SBDCD	—	—	—	—	SBDDE	SBDAE	SBDPE	SBDCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SBDCE	GTCCR Register Buffer Operation Simultaneous Enable 0: Disables simultaneous enabling GTCCR buffer operations 1: Enables GTCCR register buffer operations simultaneously	R/W
1	SBDPE	GTPR Register Buffer Operation Simultaneous Enable 0: Disables simultaneous enabling GTPR buffer operations 1: Enables GTPR register buffer operations simultaneously	R/W*1
2	SBDAE	GTADTR Register Buffer Operation Simultaneous Enable 0: Disables simultaneous enabling GTADTR buffer operations 1: Enables GTADTR register buffer operations simultaneously	R/W
3	SBDDE	GTDV Register Buffer Operation Simultaneous Enable*2 0: Disables simultaneous enabling GTDV buffer operations 1: Enables GTDV register buffer operations simultaneously	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	SBDCD	GTCCR Register Buffer Operation Simultaneous Disable 0: Disables simultaneous disabling GTCCR buffer operations 1: Disables GTCCR register buffer operations simultaneously	R/W
9	SBDPD	GTPR Register Buffer Operation Simultaneous Disable 0: Disables simultaneous disabling GTPR buffer operations 1: Disables GTPR register buffer operations simultaneously	R/W*1
10	SBDAD	GTADTR Register Buffer Operation Simultaneous Disable 0: Disables simultaneous disabling GTADTR buffer operations 1: Disables GTADTR register buffer operations simultaneously	R/W
11	SBDDD	GTDV Register Buffer Operation Simultaneous Disable*2 0: Disables simultaneous disabling GTDV buffer operations 1: Disables GTDV register buffer operations simultaneously	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	SPCE	Period Count Function Simultaneous Enable 0: Disables simultaneous enabling period count function 1: Enables period count function simultaneously	R/W
17	SSCE	Synchronous Set/Clear Simultaneous Enable 0: Disables simultaneous enabling synchronous set/clear 1: Enables synchronous set/clear simultaneously	R/W*1
23:18	—	These bits are read as 0. The write value should be 0.	R/W
24	SPCD	Period Count Function Simultaneous Disable 0: Disables simultaneous disabling period count function 1: Disables period count function simultaneously	R/W
25	SSCD	Synchronous Set/Clear Simultaneous Disable 0: Disables simultaneous disabling synchronous set/clear 1: Disables synchronous set/clear simultaneously	R/W*1
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In complementary PWM mode, no matter which register of master channel/slave channel 1/slave channel 2 is written, three channels are written at the same time.
Note 2. This bit is only available in GPT160 to GPT162.
In GPT163, this bit is read as 0. The write value should be 0.

The GTSECR register simultaneously updates the value for operation enable bits of a channel set by the GTSECSR register.

Writing 1 to a bit in the GTSECR register in any channel and updating it updates an operation enable bit for all channels, related to the position of the bit written with 1 by the all GTSECSR registers.

Setting enable and disable bits for the same operation enable bit to 1 in the GTSECR is prohibited.

A bit written to 1 is automatically cleared. When the GTSECR is read, 0 is read.

Access in 8-bit or 16-bit unit to the GTSECR register is prohibited, and it should be accessed in 32-bit unit.

SBDCE bit (GTCCR Register Buffer Operation Simultaneous Enable)

The SBDCE bit simultaneously enables GTCCR register buffer operation.

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[0] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTCCRA, GTCCRC, and GTCCRD registers and using the GTCCRB, GTCCRE, and GTCCRF registers are enabled.

Simultaneous setting of SBDCE and SBDCD bits to 1 is prohibited.

SBDPE bit (GTPR Register Buffer Operation Simultaneous Enable)

The SBDPE bit simultaneously enables GTPR register buffer operation.

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[1] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTPR, GTPDBR, and GTPBR registers are enabled.

Simultaneous setting of SBDPE and SBDPD bits to 1 is prohibited.

SBDAE bit (GTADTR Register Buffer Operation Simultaneous Enable)

The SBDAE bit simultaneously enables GTADTR register buffer operation.

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[2] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTADTRA, GTADTBRA, and GTADTDBRA registers and using the GTADTRB, GTADTBRB, and GTADTDBRB registers are enabled.

Simultaneous setting of SBDAE and SBDAD bits to 1 is prohibited.

SBDDE bit (GTDV Register Buffer Operation Simultaneous Enable)

The SBDDE bit simultaneously enables GTDV register buffer operation.

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[3] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTDVU and GTDBU registers and using the GTDVD and GTDBD registers are enabled.

Simultaneous setting of SBDDE and SBDDD bits to 1 is prohibited.

SBDCD bit (GTCCR Register Buffer Operation Simultaneous Disable)

The SBDCD bit simultaneously disables GTCCR register buffer operation.

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[0] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTCCRA, GTCCRC, and GTCCRD registers and using the GTCCRB, GTCCRE, and GTCCRF registers are disabled.

Simultaneous setting of SBDCE and SBDCD bits to 1 is prohibited.

SBDPD bit (GTPR Register Buffer Operation Simultaneous Disable)

The SBDPD bit simultaneously disables GTPR register buffer operation.

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[1] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTPR, GTPDBR, and GTPBR registers are disabled.

Simultaneous setting of SBDPE and SBDPD bits to 1 is prohibited.

SBDAD bit (GTADTR Register Buffer Operation Simultaneous Disable)

The SBDAD bit simultaneously disables GTADTR register buffer operation.

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[2] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTADTRA, GTADTBRA, and GTADTDBRA registers and using the GTADTRB, GTADTBRB, and GTADTDBRB registers are disabled.

Simultaneous setting of SBDAE and SBDAD bits to 1 is prohibited.

SBDDD bit (GTDV Register Buffer Operation Simultaneous Disable)

The SBDDD bit simultaneously disables GTDV register buffer operation.

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[3] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTDVU and GTDBU registers and using the GTDVD and GTDBD registers are disabled.

Simultaneous setting of SBDDE and SBDDD bits to 1 is prohibited.

SPCE bit (Period Count Function Simultaneous Enable)

The SPCE bit simultaneously enables period count function.

When 1 is written to this bit, 1 is simultaneously set to GTPC.PCEN bit in the channels set to 1 by the GTSECSR register, and period count function is enabled.

Simultaneous setting of SPCE and SPCD bits to 1 is prohibited.

SSCE bit (Synchronous Set/Clear Simultaneous Enable)

The SSCE bit simultaneously enables synchronous set/clear function.

When 1 is written to this bit, 1 is simultaneously set to GTCR.SSCEN bit in the channels set to 1 by the GTSECSR register and Enable synchronous set/clear function.

Simultaneous setting of SSCE and SSCD bits to 1 is prohibited.

SPCD bit (Period Count Function Simultaneous Disable)

The SPCD bit simultaneously disables period count function.

When 1 is written to this bit, 0 is simultaneously set to GTPC.PCEN bit in the channels set to 1 by the GTSECSR register, and period count function is disabled.

Simultaneous setting of SPCE and SPCD bits to 1 is prohibited.

SSCD bit (Synchronous Set/Clear Simultaneous Disable)

The SSCD bit simultaneously disables synchronous set/clear function.

When 1 is written to this bit, 0 is simultaneously set to GTCR.SSCEN bit in the channels set to 1 by the GTSECSR register and Disable synchronous set/clear function.

Simultaneous setting of SSCE and SSCD bits to 1 is prohibited.

20.2.42 GTBER2 : General PWM Timer Buffer Enable Register 2

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0xE0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	OLTTB[1:0]		OLTTA[1:0]		CPBTD	CP3DB	—		CPTDV	CPTADB	CPTADA	CPTPR	CPTCB	CPTCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CMTADB	CMTADA	—	CMTCB[1:0]		CMTCA[1:0]		—	—	CCTDV	CCTADB	CCTADA	CCTPR	CCTCB	CCTCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CCTCA	Counter Clear Source GTCCRA Register Buffer Transfer Disable 0: Enables GTCCRA register buffer transfer by counter clear 1: Disables GTCCRA register buffer transfer by counter clear	R/W
1	CCTCB	Counter Clear Source GTCCRB Register Buffer Transfer Disable 0: Enables GTCCRB register buffer transfer by counter clear 1: Disables GTCCRB register buffer transfer by counter clear	R/W
2	CCTPR	Counter Clear Source GTPR Register Buffer Transfer Disable 0: Enables GTPR register buffer transfer by counter clear 1: Disables GTPR register buffer transfer by counter clear	R/W
3	CCTADA	Counter Clear Source GTADTRA Register Buffer Transfer Disable 0: Enables GTADTRA register buffer transfer by counter clear 1: Disables GTADTRA register buffer transfer by counter clear	R/W
4	CCTADB	Counter Clear Source GTADTRB Register Buffer Transfer Disable 0: Enables GTADTRB register buffer transfer by counter clear 1: Disables GTADTRB register buffer transfer by counter clear	R/W
5	CCTDV	Counter Clear Source GTDVU/GTDVD Register Buffer Transfer Disable 0: Enables GTDVU/GTDVD register buffer transfer by counter clear 1: Disables GTDVU/GTDVD register buffer transfer by counter clear	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
9:8	CMTCA[1:0]	Compare Match Source GTCCRA Register Buffer Transfer Enable 0 0: Disables GTCCRA register Buffer Transfer by compare match of GTCCRA register and GTCCRB register 0 1: Enables GTCCRA register Buffer Transfer by compare match of GTCCRA register 1 0: Enables GTCCRA register Buffer Transfer by compare match of GTCCRB register 1 1: Enables GTCCRA register Buffer Transfer by compare match of GTCCRA register and GTCCRB register	R/W
11:10	CMTCB[1:0]	Compare Match Source GTCCRB Register Buffer Transfer Enable 0 0: Disables GTCCRB register Buffer Transfer by compare match of GTCCRA register and GTCCRB register 0 1: Enables GTCCRB register Buffer Transfer by compare match of GTCCRA register 1 0: Enables GTCCRB register Buffer Transfer by compare match of GTCCRB register 1 1: Enables GTCCRB register Buffer Transfer by compare match of GTCCRA register and GTCCRB register	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	CMTADA	Compare Match Source GTADTRA Register Buffer Transfer Enable 0: Disables GTADTRA register buffer transfer by compare match of GTADTRA register 1: Enables GTADTRA register buffer transfer by compare match of GTADTRA register	R/W

Bit	Symbol	Function	R/W
14	CMTADB	Compare Match Source GTADTRB Register Buffer Transfer Enable 0: Disables GTADTRB register buffer transfer by compare match of GTADTRB register 1: Enables GTADTRB register buffer transfer by compare match of GTADTRB register	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
16	CPTCA	Overflow/Underflow Source GTCCRA Register Buffer Transfer Disable 0: Enables GTCCRA register buffer transfer by overflow/underflow 1: Disables GTCCRA register buffer transfer by overflow/underflow	R/W
17	CPTCB	Overflow/Underflow Source GTCCRB Register Buffer Transfer Disable 0: Enables GTCCRB register buffer transfer by overflow/underflow 1: Disables GTCCRB register buffer transfer by overflow/underflow	R/W
18	CPTPR	Overflow/Underflow Source GTPR Register Buffer Transfer Disable 0: Enables GTPR register buffer transfer by overflow/underflow 1: Disables GTPR register buffer transfer by overflow/underflow	R/W
19	CPTADA	Overflow/Underflow Source GTADTRA Register Buffer Transfer Disable 0: Enables GTADTRA register buffer transfer by overflow/underflow 1: Disables GTADTRA register buffer transfer by overflow/underflow	R/W
20	CPTADB	Overflow/Underflow Source GTADTRB Register Buffer Transfer Disable 0: Enables GTADTRB register buffer transfer by overflow/underflow 1: Disables GTADTRB register buffer transfer by overflow/underflow	R/W
21	CPTDV	Overflow/Underflow Source GTDVU/GTDVD Register Buffer Transfer Disable 0: Enables GTDVU/GTDVD register buffer transfer by overflow/underflow 1: Disables GTDVU/GTDVD register buffer transfer by overflow/underflow	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	CP3DB	Complementary PWM mode 3,4 Double Buffer select*1 0: Disables double buffer function in complementary PWM mode 3, 4 1: Enables double buffer function in complementary PWM mode 3, 4	R/W
25	CPBTD	Complementary PWM mode Buffer Transfer Disable*1 0: Enables buffer transfer from temporary register to GTCCRC and GTPBR register 1: Disables buffer transfer from temporary register to GTCCRC and GTPBR register	R/W
27:26	OLTTA[1:0]	GTIOCnA Output Level Buffer Transfer Timing Select*1 0 0: No transfer 0 1: Triangle-waves, complementary PWM mode: Transfer at crest Saw-waves: Transfer at the end of period 1 0: Triangle-waves, complementary PWM mode: Transfer at trough Saw-waves: Transfer by compare match of GTCCRA register 1 1: Triangle-waves, complementary PWM mode: Transfer at both crest and trough Saw-waves: Setting prohibited	R/W
29:28	OLTTB[1:0]	GTIOCnB Output Level Buffer Transfer Timing Select*1 0 0: No transfer 0 1: Triangle-waves, complementary PWM mode: Transfer at crest Saw-waves: Transfer at the end of period 1 0: Triangle-waves, complementary PWM mode: Transfer at trough Saw-waves: Transfer by compare match of GTCCRB register 1 1: Triangle-waves, complementary PWM mode: Transfer at both crest and trough Saw-waves: Setting prohibited	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is only available in GPT160 to GPT162.
In GPT163, this bit is read as 0. The write value should be 0.

The GTBER2 register makes settings for buffer operation.

Set the CP3DB bit and OLTTm[1:0] (m = A, B) bits when the GTCNT counter is stopped.

CCTCA bit (Counter Clear Source GTCCRA Register Buffer Transfer Disable)

The CCTCA bit disables buffer transfer by counter clear using the GTCCRA, GTCCRC, and GTCCRD registers together.

This bit is effective when the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRA[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTCA bit is 0, buffer transfer is not performed by counter clear.

If there is a conflict with the CMTCA bit setting, the CCTCA bit setting has priority.

The setting is invalid during the event count operation.

CCTCB bit (Counter Clear Source GTCCRB Register Buffer Transfer Disable)

The CCTCB bit disables buffer transfer by counter clear using the GTCCRB, GTCCRE, and GTCCRF registers together.

This bit is effective when the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRB[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTCB bit is 0, buffer transfer is not performed by counter clear.

If there is a conflict with the CMTCB bit setting, the CCTCB bit setting has priority.

The setting is invalid during the event count operation.

CCTPR bit (Counter Clear Source GTPR Register Buffer Transfer Disable)

The CCTPR bit disables buffer transfer by counter clear using the GTPR, GTPBR, and GTPDBR registers together.

This bit is effective when the GTBER.BD[1] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.PR[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTPR bit is 0, buffer transfer is not performed by counter clear.

The setting is invalid during the event count operation.

CCTADm bit (Counter Clear Source GTADTRm Register Buffer Transfer Disable) (m = A, B)

The CCTADm bit disables buffer transfer by counter clear using the GTADTRm, GTADTBRm and GTADTDBRm registers together.

This bit is effective when the GTBER.BD[2] bit is 0 (buffer operation enabled) and the buffer operation is selected by the ADTTm[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTADm bit is 0, buffer transfer is not performed by counter clear.

If there is a conflict with the CMTADm bit setting, the CCTADm bit setting has priority.

The setting is invalid during the event count operation.

CCTDV bit (Counter Clear Source GTDVU/GTDVD Register Buffer Transfer Disable)

The CCTDV bit disables buffer transfer by counter clear using the GTDVU and GTDBU or GTDVD and GTDBD registers together.

This bit is effective when the GTBER.BD[3] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTDTCR.TDBUE or GTDTCR.TDBDE bit with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode. Even if the CCTDV bit is 0, buffer transfer is not performed by counter clear.

The setting is invalid during the event count operation.

CMTCA[1:0] bits (Compare Match Source GTCCRA Register Buffer Transfer Enable)

The CMTCA[1:0] bits enable buffer transfer by compare match of GTCCRA using the GTCCRA, GTCCRC, and GTCCRD registers together.

This bit is effective when the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRA[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

If there is a conflict with the CCTCA bit setting, the CCTCA bit setting has priority.

The setting is invalid during the event count operation.

CMTCB[1:0] bits (Compare Match Source GTCCRB Register Buffer Transfer Enable)

The CMTCB[1:0] bits enable buffer transfer by compare match of GTCCRB using the GTCCRB, GTCCRE, and GTCCRF registers together.

This bit is effective when the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRB[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

If there is a conflict with the CCTCB bit setting, the CCTCB bit setting has priority.

The setting is invalid during the event count operation.

CMTADm bit (Compare Match Source GTADTRm Register Buffer Transfer Enable) (m = A, B)

The CMTADm bit enables buffer transfer by compare match of GTADTRm using the GTADTRm, GTADTBRm and GTADTDBRm registers together.

This bit is effective when the GTBER.BD[2] bit is 0 (buffer operation enabled) and the buffer operation is selected by the ADTTm[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

If there is a conflict with the CCTADm bit setting, the CCTADm bit setting has priority.

The setting is invalid during the event count operation.

CPTCA bit (Overflow/Underflow Source GTCCRA Register Buffer Transfer Disable)

The CPTCA bit disables buffer transfer by overflow/underflow in saw-waves using the GTCCRA, GTCCRC, and GTCCRD registers together.

This bit is effective when the CCTCA bit is 0 (GTCCRA register buffer transfer by counter clear is enabled), the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRA[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CPTCB bit (Overflow/Underflow Source GTCCRB Register Buffer Transfer Disable)

The CPTCB bit disables buffer transfer by overflow/underflow in saw-waves using the GTCCRB, GTCCRE and GTCCRF registers together.

This bit is effective when the CCTCB bit is 0 (GTCCRB register buffer transfer by counter clear is enabled), the GTBER.BD[0] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.CCRB[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CPTPR bit (Overflow/Underflow Source GTPR Register Buffer Transfer Disable)

The CPTPR bit disables buffer transfer by overflow/underflow in saw-waves using the GTPR, GTPBR, and GTPDBR registers together.

This bit is effective when the CCTPR bit is 0 (GTPR register buffer transfer by counter clear is enabled), the GTBER.BD[1] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTBER.PR[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CPTADm bit (Overflow/Underflow Source GTADTRm Register Buffer Transfer Disable) (m = A, B)

The CPTADm bit disables buffer transfer by overflow/underflow in saw-waves using the GTADTRm, GTADTBRm and GTADTDBRm registers together.

This bit is effective when the CCTADm bit is 0 (GTADTRm register buffer transfer by counter clear is enabled), the GTBER.BD[2] bit is 0 (buffer operation enabled) and the buffer operation is selected by the ADTTm[1:0] bits with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CPTDV bit (Overflow/Underflow Source GTDVU/GTDVD Register Buffer Transfer Disable)

The CPTDV bit disables buffer transfer by overflow/underflow in saw-waves using the GTDVU and GTDBU or GTDVD and GTDBD registers together.

This bit is effective when the CCTDV bit is 0 (GTDVU/GTDVD register buffer transfer by counter clear is enabled), the GTBER.BD[3] bit is 0 (buffer operation enabled) and the buffer operation is selected by the GTDTCR.TDBUE or GTDTCR.TDBDE bit with saw-waves.

The setting is invalid in triangle waves or complementary PWM mode.

The setting is invalid during the event count operation.

CP3DB bit (Complementary PWM mode 3,4 Double Buffer select)

The CP3DB bit enables buffer transfer using the GTCCRA, GTCCRE and GTCCRF registers together in complementary PWM mode 3 and 4.

CPBTD bit (Complementary PWM mode Buffer Transfer Disable)

The CPBTD bit disables buffer transfer from temporary register(temporary register A and temporary register P) to the GTCCRC and GTPBR registers during timer counting in complementary PWM mode 1, 2, and 3. When CP3DB bit is 1, the buffer transfer from temporary register B to GTCCRE register is also disabled. The setting is invalid in complementary PWM mode 4.

OLTTm[1:0] bits (GTIOCnA Output Level Buffer Transfer Timing Select) (m = A, B)

The OLTTm[1:0] bits set the timing of buffer transfer from GTOLBR.GTIOmB[4:0] bits to GTIOR.GTIOm[4:0] bits.

20.2.43 GTOLBR : General PWM Timer Output Level Buffer Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)

Offset address: 0xE4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	GTIOBB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Function	R/W
4:0	GTIOAB[4:0]	GTIOA buffer bits	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W
20:16	GTIOBB[4:0]	GTIOB buffer bits	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The GTOLBR register is a buffer register for the GTIOR.GTIOA[4:0] bits and GTIOR.GTIOB[4:0] bits.

GTIOmB[4:0] bits (GTIOm buffer bits) (m = A, B)

The GTIOmB[4:0] bits are buffer bits of GTIOR.GTIOm[4:0] bits.

These bits are transferred to the GTIOR.GTIOm[4:0] bits at the transfer timing selected by the GTBER2.OLTTm[1:0] (m = A, B) bits.

20.2.44 GTICCR : General PWM Timer Inter Channel Cooperation Input Capture Control Register

Base address: GPT16n = 0x4008_9000 + 0x0100 × n (n = 0 to 2)
GPT163 = 0x4008_9300

Offset address: 0xEC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ICBGRP[1:0]		—	—	—	—	—	ICBCLK	ICBFPU	ICBFPO	ICBFF	ICBFE	ICBFD	ICBFC	ICBFB	ICBFA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ICAGRP[1:0]		—	—	—	—	—	ICACLK	ICAFPU	ICAFPO	ICAFF	ICAFE	ICAFD	ICAFC	ICAFB	ICAFA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ICAFA	Forwarding GTCCRA register Compare Match/Input Capture to Other Channel GTCCRA Input Capture Source Enable 0: Disables forwarding GTCCRA register compare match/input capture to GTCCRA input capture source of other channels 1: Enables forwarding GTCCRA register compare match/input capture to GTCCRA input capture source of other channels	R/W
1	ICAFB	Forwarding GTCCRB register Compare Match/Input Capture to Other Channel GTCCRA Input Capture Source Enable 0: Disables forwarding GTCCRB register compare match/input capture to GTCCRA input capture source of other channels 1: Enables forwarding GTCCRB register compare match/input capture to GTCCRA input capture source of other channels	R/W
2	ICAFC	Forwarding GTCCRC register Compare Match to Other Channel GTCCRA Input Capture Source Enable 0: Disables forwarding GTCCRC register compare match to GTCCRA input capture source of other channels 1: Enables forwarding GTCCRC register compare match to GTCCRA input capture source of other channels	R/W
3	ICAFD	Forwarding GTCCRD register Compare Match to Other Channel GTCCRA Input Capture Source Enable 0: Disables forwarding GTCCRD register compare match to GTCCRA input capture source of other channels 1: Enables forwarding GTCCRD register compare match to GTCCRA input capture source of other channels	R/W
4	ICAFE	Forwarding GTCCRE register Compare Match to Other Channel GTCCRA Input Capture Source Enable 0: Disables forwarding GTCCRE register compare match to GTCCRA input capture source of other channels 1: Enables forwarding GTCCRE register compare match to GTCCRA input capture source of other channels	R/W
5	ICAFF	Forwarding GTCCRF register Compare Match to Other Channel GTCCRA Input Capture Source Enable 0: Disables forwarding GTCCRF register compare match to GTCCRA input capture source of other channels 1: Enables forwarding GTCCRF register compare match to GTCCRA input capture source of other channels	R/W

Bit	Symbol	Function	R/W
6	ICAFPO	Forwarding Overflow to Other Channel GTCCRA Input Capture Source Enable 0: Disables forwarding overflow in saw-waves or the crest in triangle-waves or complementary PWM mode to GTCCRA input capture source of other channels 1: Enables forwarding overflow in saw-waves or the crest in triangle-waves or complementary PWM mode to GTCCRA input capture source of other channels	R/W
7	ICAFPU	Forwarding Underflow to Other Channel GTCCRA Input Capture Source Enable 0: Disables forwarding underflow in saw-waves or the trough in triangle-waves or complementary PWM mode to GTCCRA input capture source of other channels 1: Enables forwarding underflow in saw-waves or the trough in triangle-waves or complementary PWM mode to GTCCRA input capture source of other channels	R/W
8	ICACLK	Forwarding Count Clock to Other Channel GTCCRA Input Capture Source Enable 0: Disables forwarding count clock to GTCCRA input capture source of other channels 1: Enables forwarding count clock to GTCCRA input capture source of other channels	R/W
13:9	—	These bits are read as 0. The write value should be 0.	R/W
15:14	ICAGRP[1:0]	GTCCRA Input Capture Group Select 0 0: Selects group A 0 1: Selects group B 1 0: Selects group C 1 1: Selects group D	R/W
16	ICBFA	Forwarding GTCCRA register Compare Match/Input Capture to Other Channel GTCCRB Input Capture Source Enable 0: Disables forwarding GTCCRA register compare match/input capture to GTCCRB input capture source of other channels 1: Enables forwarding GTCCRA register compare match/input capture to GTCCRB input capture source of other channels	R/W
17	ICBFB	Forwarding GTCCRB register Compare Match/Input Capture to Other Channel GTCCRB Input Capture Source Enable 0: Disables forwarding GTCCRB register compare match/input capture to GTCCRB input capture source of other channels 1: Enables forwarding GTCCRB register compare match/input capture to GTCCRB input capture source of other channels	R/W
18	ICBFC	Forwarding GTCCRC register Compare Match to Other Channel GTCCRB Input Source Capture Enable 0: Disables forwarding GTCCRD register compare match to GTCCRB input capture source of other channels 1: Enables forwarding GTCCRD register compare match to GTCCRB input capture source of other channels	R/W
19	ICBFD	Forwarding GTCCRD register Compare Match to Other Channel GTCCRB Input Capture Source Enable 0: Disables forwarding GTCCRD register compare match to GTCCRB input capture source of other channels 1: Enables forwarding GTCCRD register compare match to GTCCRB input capture source of other channels	R/W
20	ICBFE	Forwarding GTCCRE register Compare Match to Other Channel GTCCRB Input Capture Source Enable 0: Disables forwarding GTCCRE register compare match to GTCCRB input capture source of other channels 1: Enables forwarding GTCCRE register compare match to GTCCRB input capture source of other channels	R/W
21	ICBFF	Forwarding GTCCRF register Compare Match to Other Channel GTCCRB Input Capture Source Enable 0: Disables forwarding GTCCRF register compare match to GTCCRB input capture source of other channels 1: Enables forwarding GTCCRF register compare match to GTCCRB input capture source of other channels	R/W

Bit	Symbol	Function	R/W
22	ICBFPO	Forwarding Overflow to Other Channel GTCCRB Input Capture Source Enable 0: Disables forwarding overflow in saw-waves or the crest in triangle-waves or complementary PWM mode to GTCCRB input capture source of other channels 1: Enables forwarding overflow in saw-waves or the crest in triangle-waves or complementary PWM mode to GTCCRB input capture source of other channels	R/W
23	ICBFPU	Forwarding Underflow to Other Channel GTCCRB Input Capture Source Enable 0: Disables forwarding underflow in saw-waves or the trough in triangle-waves or complementary PWM mode to GTCCRB input capture source of other channels 1: Enables forwarding underflow in saw-waves or the trough in triangle-waves or complementary PWM mode to GTCCRB input capture source of other channels	R/W
24	ICBCLK	Forwarding Count Clock to Other Channel GTCCRB Input Capture Source Enable 0: Disables forwarding count clock to GTCCRB input capture source of other channels 1: Enables forwarding count clock to GTCCRB input capture source of other channels	R/W
29:25	—	These bits are read as 0. The write value should be 0.	R/W
31:30	ICBGRP[1:0]	GTCCRB Input Capture Group Select 0 0: Selects group A 0 1: Selects group B 1 0: Selects group C 1 1: Selects group D	R/W

The GTICCR register is a register that controls input capture by inter channel cooperation.

For channels that perform input capture by inter channel cooperation, the forwarding enable bit of the input capture source corresponding to the GTCCRA register or GTCCRB register where the input capture occurs is invalid.

ICAFm bit (Forwarding GTCCRm register Compare Match/Input Capture to Other Channel GTCCRA Input Capture Source Enable) (m = A, B)

The ICAFm bit enables or disables the use of compare match/input capture of GTCCRm register as the input capture source of other channel's GTCCRA register.

ICAFx bit (Forwarding GTCCRx register Compare Match to Other Channel GTCCRA Input Capture Source Enable) (x = C, D, E, F)

The ICAFx bit enables or disables the use of compare match of GTCCRx register as the input capture source of other channel's GTCCRA register.

ICAFPO bit (Forwarding Overflow to Other Channel GTCCRA Input Capture Source Enable)

The ICAFPO bit enables or disables the use of overflow of saw-waves, the crest of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRA register.

ICAFPU bit (Forwarding Underflow to Other Channel GTCCRA Input Capture Source Enable)

The ICAFPU bit enables or disables the use of underflow of saw-waves, the trough of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRA register.

ICACLK bit (Forwarding Count Clock to Other Channel GTCCRA Input Capture Source Enable)

The ICACLK bit enables or disables the use of count clock as the input capture source of other channel's GTCCRA register.

ICAGRP[1:0] bits (GTCCRA Input Capture Group Select)

The ICAGRP[1:0] bits select the group of input capture by inter channel cooperation for GTCCRA register.

For channels that accept input capture of the GTCCRA register due to input capture sources from other channels, set the GTICASR.ASOC bit to 1 and select the group of inter channel cooperation with the ICAGRP[1:0] bits.

ICBFm bit (Forwarding GTCCRm register Compare Match/Input Capture to Other Channel GTCCRB Input Capture Source Enable) (m = A, B)

The ICBFm bit enables or disables the use of compare match/input capture of GTCCRm register as the input capture source of other channel's GTCCRB register.

ICBFx bit (Forwarding GTCCRx register Compare Match to Other Channel GTCCRB Input Capture Source Enable) (x = C, D, E, F)

The ICBFx bit enables or disables the use of compare match of GTCCRx register as the input capture source of other channel's GTCCRB register.

ICBFPO bit (Forwarding Overflow to Other Channel GTCCRB Input Capture Source Enable)

The ICBFPO bit enables or disables the use of overflow of saw-waves, the crest of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRB register.

ICBFPU bit (Forwarding Underflow to Other Channel GTCCRB Input Capture Source Enable)

The ICBFPU bit enables or disables the use of underflow of saw-waves, the trough of triangle-waves or complementary PWM mode as the input capture source of other channel's GTCCRB register.

ICBCLK bit (Forwarding Count Clock to Other Channel GTCCRB Input Capture Source Enable)

The ICBCLK bit enables or disables the use of count clock as the input capture source of other channel's GTCCRB register.

ICBGRP[1:0] bits (GTCCRB Input Capture Group Select)

The ICBGRP[1:0] bits select the group of input capture by inter channel cooperation for GTCCRB register.

For channels that accept input capture of the GTCCRB register due to input capture sources from other channels, set the GTICBSR.BSOC bit to 1 and select the group of inter channel cooperation with the ICBGRP[1:0] bits.

20.2.45 OPSCR : Output Phase Switching Control Register

Base address: GPT_OPS = 0x4008_9A00

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCS[1:0]		NFEN	—	—	GODF	GRP[1:0]		—	—	ALIGN	RV	INV	N	P	FB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	EN	—	W	V	U	—	WF	VF	UF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	UF	Input Phase Soft Setting These bits set the input phase from software settings. Setting these bits is valid when OPSCR.FB = 1.	R/W
1	VF		R/W
2	WF		R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	U	Input U-Phase Monitor This bit monitors the state of the input phase. OPSCR.FB = 0 : External input that are synchronized by GTCLK OPSCR.FB = 1 : Software settings (UF)	R
5	V	Input V-Phase Monitor This bit monitors the state of the input phase. OPSCR.FB = 0 : External input that are synchronized by GTCLK OPSCR.FB = 1 : Software settings (VF)	R
6	W	Input W-Phase Monitor This bit monitors the state of the input phase. OPSCR.FB = 0 : External input that are synchronized by GTCLK OPSCR.FB = 1 : Software settings (WF)	R
7	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	EN	Output Phase Enable 0: Does not output (Hi-Z external pin) 1: Outputs*1	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	FB	External Feedback Signal Enable This bit selects the input phase from software settings and external input. 0: Selects the external input 1: Selects the soft setting (OPSCR.UF, VF, WF)	R/W
17	P	Positive-Phase Output (P) Control 0: Level signal output 1: PWM signal output	R/W
18	N	Negative-Phase Output (N) Control 0: Level signal output 1: PWM signal output	R/W
19	INV	Output Phase Invert Control 0: Positive logic (active-high) output 1: Negative logic (active-low) output	R/W
20	RV	Output Phase Rotation Direction Reversal Control 0: Positive rotation 1: Reverse rotation	R/W
21	ALIGN	Input Phase Alignment 0: Input phase aligned to GTCLK 1: Input phase aligned to the falling edge of PWM	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
25:24	GRP[1:0]	Output Disabled Source Selection 0 0: Selects group A output disable source 0 1: Selects group B output disable source 1 0: Selects group C output disable source 1 1: Selects group D output disable source	R/W
26	GODF	Group Output Disable Function 0: This bit function is ignored 1: Group disable clears the OPSCR.EN bit*1	R/W
28:27	—	These bits are read as 0. The write value should be 0.	R/W
29	NFEN	External Input Noise Filter Enable 0: Does not use a noise filter on the external input 1: Uses a noise filter on the external input	R/W
31:30	NFCS[1:0]	External Input Noise Filter Clock Selection Noise filter sampling clock setting of the external input. 0 0: GTCLK/1 0 1: GTCLK/4 1 0: GTCLK/16 1 1: GTCLK/64	R/W

Note 1. When OPSCR.GODF = 1 and the signal value selected by the OPSCR.GRP[1:0] bits is high, the OPSCR.EN bit is set to 0.

The OPSCR register sets the output of the signal waveform required for brushless DC motor control.

UF, VF, WF bits (Input Phase Soft Setting)

The UF, VF, WF bits set the input phase from the software settings. When OPSCR.FB bit is 1, these bits are valid. The set value of the UF /VF /WF takes the place of the U/V/W external input.

U, V, W bits (Input Phase Monitor)

The U, V, W bits indicate the state of the input U, V, W phase.

When the OPSCR.FB bit is 0, external inputs that are synchronized by GTCLK are monitored by these bits. When the OPSCR.FB bit is 1, the OPSCR.U, OPSCR.V, and OPSCR.W bits can read the OPSCR.UF, OPSCR.VF, and OPSCR.WF bits.

EN bit (Output Phase Enable)

The EN bit controls the output enable signal of output phase (positive phase/negative phase).

When the OPSCR.EN bit is 1, the signal waveform is output.

When the OPSCR.EN bit is 0, first set OPSCR.FB, OPSCR.UF /VF /WF (software setting is selected), OPSCR.P/N, OPSCR.INV, OPSCR.RV, OPSCR.ALIGN, OPSCR.GRP[1:0], OPSCR.GODF, OPSCR.NFEN, OPSCR.NFCS. Then, set the EN bit to 1. The EN bit should be set when output disable request does not occur from POEG. Also when OPSCR.GODF is 1 and the signal value selected in the OPSCR.GRP[1:0] bits is high, the OPSCR.EN bit is set to 0. Even if 1 is written by software, the EN bit remains at 0.

For the return, after clearing the Output Disable Request by software, set the EN bit to 1.

Priority order of the EN bit is as follows (when the conflict occurs).

When writing 1 by software and clearing to 0 by the Output Disable Request conflict for the EN bit, clearing to 0 by the Output Disable Request is enabled.

FB bit (External Feedback Signal Enable)

The FB bit selects the input phase from the software settings (OPSCR.UF, VF, WF) and external input such as a Hall element.

P bit (Positive-Phase Output (P) Control)

The P bit selects one of the level signal output or PWM signal output for the positive-phase output (GTOUUP, GTOVUP and GTOWUP pins).

N bit (Negative-Phase Output (N) Control)

The N bit selects one of the level signal output or PWM signal output for the negative-phase output (GTOULO, GTOVLO and GTOWLO pins).

INV bit (Output Phase Invert Control)

The INV bit selects one of the positive logic (active-high) output or negative logic (active-low) output for the output phase.

RV bit (Output Phase Rotation Direction Reversal Control)

The RV bit reverses the direction of rotation of the motor by inverting the input phase.

ALIGN bit (Input Phase Alignment)

The ALIGN bit selects the GTCLK or PWM for the sampling of the input phase (input phase is specified in the OPSCR.FB bit).

When OPSCR.ALIGN bit is 0, input phase is aligned to GTCLK.

Note: When the chopping is performed, there are cases where the PWM width of output is shorter than the width of the PWM used to chop just before or after switching of output phase, depending on the phase difference between the phase output switch timing and the phase of PWM.

When OPSCR.ALIGN bit is 1, input phase is aligned with the falling edge of PWM.

GRP[1:0] bits (Output Disabled Source Selection)

The GRP[1:0] bits select the output disable source.

The GRP bits should be set when GODF bit is 0. If GRP bits select a POEG except for the connected groups, the status of output pin never change to disable.

GODF bit (Group Output Disable Function)

The GODF bit clears the OPSCR.EN bit.

When OPSCR.GODF is 1 and the signal value selected by the OPSCR.GRP[1:0] bits is high, the OPSCR.EN bit is set to 0.

When OPSCR.GODF bit is 0, this bit is ignored.

The GODF bit should be set when output disable request does not occur from POEG.

NFEN bit (External Input Noise Filter Enable)

The NFEN bit selects the noise filter for external input. When OPSCR.NFEN bit is 0, a noise filter for the external input is not used.

Note: Set this bit during the EN bit is 0 to avoid generation of unintentional internal edge caused by switching this bit.

NFCS[1:0] bits (External Input Noise Filter Clock Selection)

The NFCS[1:0] bits select the clock for the external input noise filter. When the OPSCR.NFEN bit is 1, noise filter sampling clock setting of the external input is enabled.

1. Set the NFCS[1:0].
2. Wait for 2 cycles.
3. Set the OPSCR.EN bit to 1.

20.3 Operation

20.3.1 Basic Operation

Each channel has a 16-bit timer that performs a periodic count operation using the count clock and hardware sources. The count function provides both up-counting and down-counting. The GTPR or GTCCRm (m = A to F) controls the count cycle.

When the GTCNT counter value matches the value in GTCCRA or GTCCRB, the output from the associated GTIOCnA or GTIOCnB can be changed (n = 0 to 3). GTCCRA or GTCCRB can be used as an input capture register with hardware resources.

GTCCRC and GTCCRD can function as buffer registers for GTCCRA. GTCCRE and GTCCRF can function as buffer registers for GTCCRB.

20.3.1.1 Counter Operation

(1) Counter start and stop

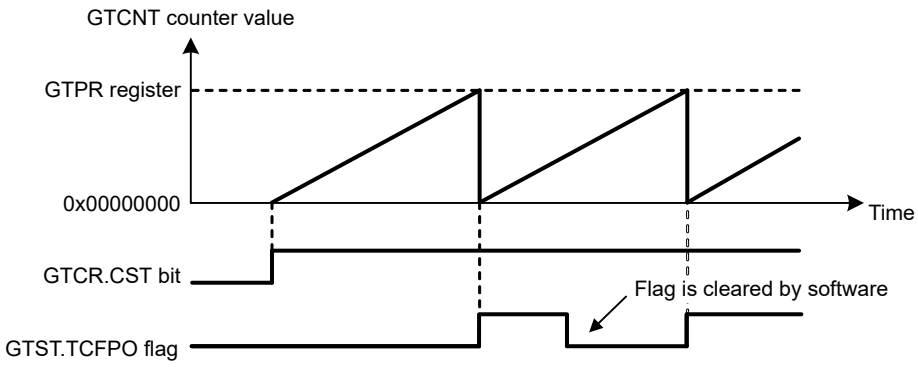
The counter of each channel starts the count operation when GTCR.CST is set to 1, and stops counting when the bit is set to 0. The GTCR.CST bit value is changed by the following sources:

- Writing to GTCR register
- Writing 1 to the bit in GTSTR associated with the GPT channel number when the GTSSR.CSTRT bit set to 1
- Writing 1 to the bit in GTSTP associated with the GPT channel number when the GTPSR.CSTOP bit set to 1
- The hardware source selected in the GTSSR register
- The hardware source selected in the GTPSR register
- Completion of the period count function while the GTPC.ASTP bit is 1

(2) Periodic count operation in up-counting by count clock

The GTCNT counter in each channel starts up-counting when the associated GTCR.CST bit is set to 1 with GTUPSR and GTDNSR registers set to 0x00000000. When the GTCNT value changes from the GTPR value to 0 (overflow) or the GTCCRm (m = A to F) value selected by the GTCSR.CSCMSC[2:0] bits to 0 in Saw-wave PWM mode 2, the GTST.TCFPO flag is set to 1, and the overflow interrupt (GPTn_OVF) is also generated. After GTCNT overflows, up-counting resumes from 0x00000000.

Figure 20.4 shows an example of a periodic count operation in up-counting by the count clock.



Note: The base address of registers are the same.

Figure 20.4 Example of periodic count operation in up-counting by the count clock

Table 20.11 shows an example for setting periodic count operation in up-counting by the count clock.

Table 20.11 Example for setting a periodic count operation in up-counting by the count clock

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.4, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.4, after 11b is set in the GTUDDTYC.UDF and UD bits, 01b is set in the UDF and UD bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	In modes other than saw-wave PWM mode 2, set the cycle with the GTPR register. In saw-wave PWM mode 2, select the GTCCRx (x = A to F) with the GTCSR.CSCMSC[2:0] bits and set the cycle with the GTPR register.
5	Set initial value for counter	Set the initial value to the GTCNT counter. In Figure 20.4, 0x00000000 is set.
6	Start count operation	Set 1 to the GTCR.CST bit and the corresponding channel starts count operation.

(3) Periodic count operation in down-counting by count clock

The GTCNT counter in each channel can perform down-counting by setting GTUDDTYC.UD with GTUPSR and GTDNSR registers set to 0x00000000. When GTCNT changes from 0 to the GTPR value (underflow), GTST.TCFPU is set to 1, and the underflow interrupt(GPTn_UDF) is also generated. After the GTCNT counter underflows, down-counting resumes from the GTPR value.

Figure 20.5 shows an example of periodic count operation in down-counting by the count clock.

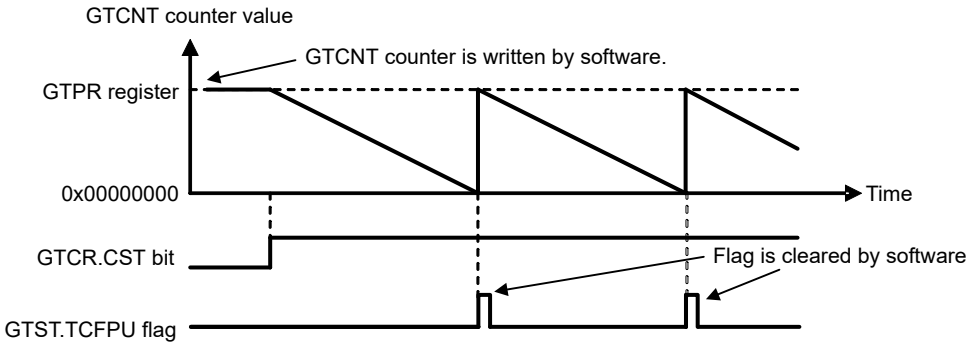


Figure 20.5 Example of periodic count operation in down-counting by the count clock

Table 20.12 shows an example for setting periodic count operation in down-counting by the count clock.

Table 20.12 Example for setting periodic count operation in down-counting by count clock

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.5, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction with the GTUDDTYC register. In Figure 20.5, after 10b is set in the GTUDDTYC.UDF and UD bits, 00b is set in the GTUDDTYC.UDF and UD bits (down-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 20.5, the GTPR register value is set.
6	Start count operation	Set the GTCR.CST bit to 1 to start count operation. In Figure 20.5, 1 is set in the CST bit.

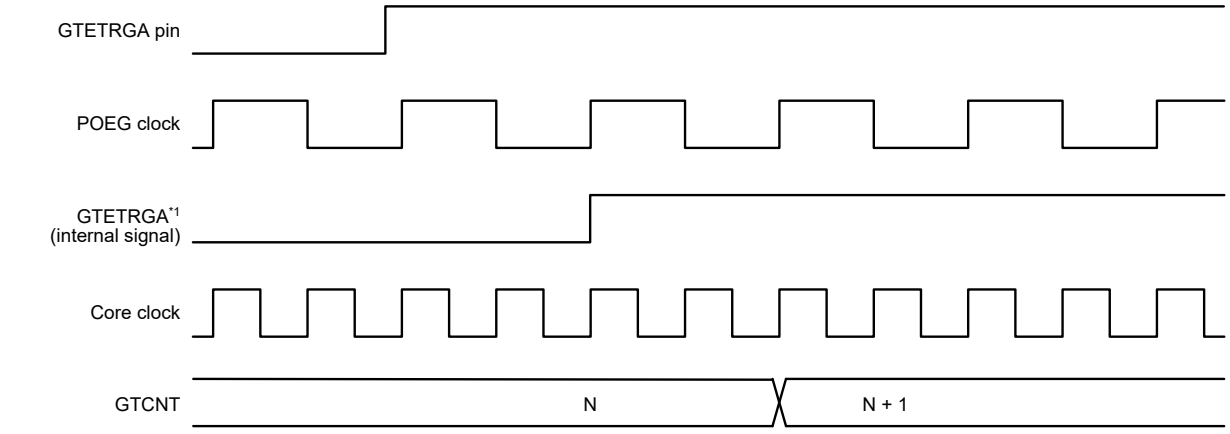
(4) Event count operation in up-counting using hardware sources

The GTCNT counter in each channel can perform up-counting using hardware sources as set in GTUPSR.

When GTUPSR is set to enable, the count clock selected in GTCR.TPCS[3:0] and the count direction selected in GTUDDTYC.UD are ignored. If up-counting and down-counting using hardware sources occur at the same time, the GTCNT counter value does not change. The overflow behavior when up-counting using hardware sources is the same as when up-counting by the count clock.

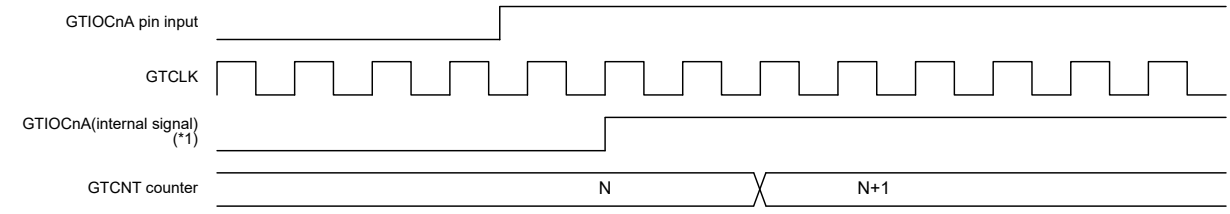
If you are using a hardware source to count up, set the GTCR.CST bit to 1 to enable the counting operation. After GTCR.CST is set to 1, the counter cannot count up for 1 clock cycle as specified in GTCR.TPCS[3:0] because the count operation is synchronized by the count clock selected in GTCR.TPCS[3:0]. Set GTCR.TPCS[3:0] to 0x0 to count up with a 1 GTCLK delay after GTCR.CST is set to 1.

Figure 20.6 and Figure 20.7 show an example of an event count operation in up-counting by a hardware resource (the rising edge of GTETRGA pin input and the rising edge of GTIOCnA pin input).



Note: POEG clock: PCLKB, Core clock: PCLKD
Note 1. The GTETRGA internal signal reflects the signal being input to the GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 20.6 Example of event count operation in up-counting using hardware sources



Note 1. This is the case when the digital noise filter is not in use.

Figure 20.7 Example of event count operation (up-counting of rising edges of the input on the GTIOCnA pin)

Figure 20.8 shows an example of event count operation by ELC_GPTA event input.

This is an example of event count operation of GPT161.GTCNT counter. An event signal is output to the ELC after matches in comparison with the GPT160.GTCCRA register. This is selected as a trigger for output to the GPT161 by the ELC as ELC_GPTA.

The GPT160 compare match A signal is synchronized with GTCLK. The ELC receives the signal in synchronization with PCLKB, and outputs the GPT16 event source A signal after 1 cycle of PCLKB has elapsed.

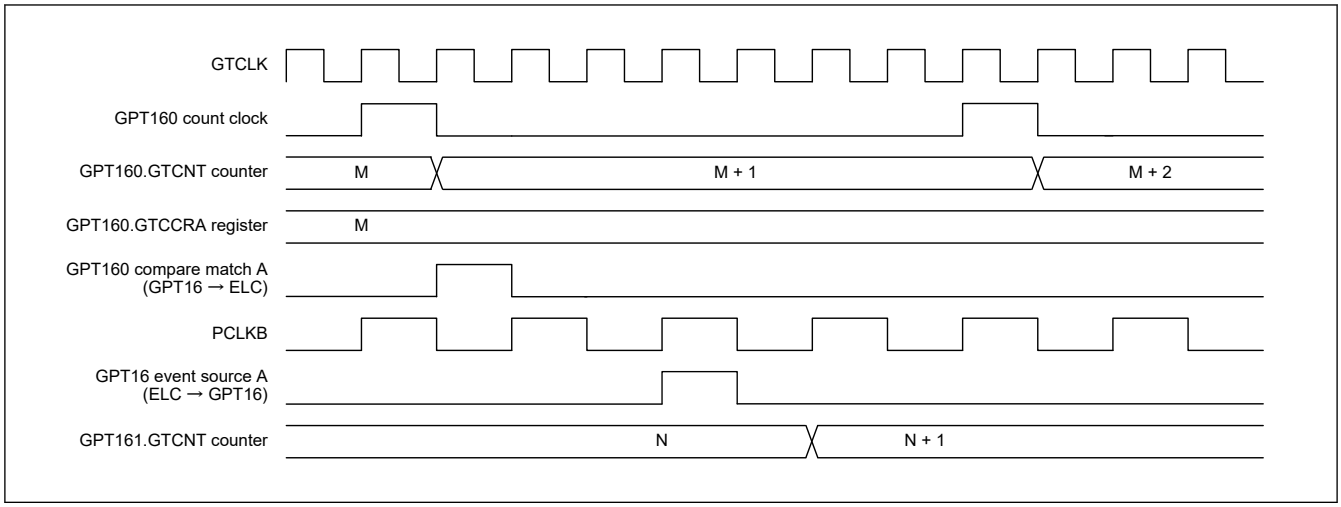


Figure 20.8 Example of event count operation (up-counting the number of event signals input from ELC_GPTA)

Table 20.13 shows an example for setting event count operation in up-counting by a hardware source.

Table 20.13 Example for setting an event count operation in up-counting using hardware sources

No.	Step Name	Description
1	Set count source	Select the counting-up hardware source with the GTUPSR register.
2	Set cycle	Set the cycle in the GTPR register.
3	Set initial value for counter	Set the initial value in the GTCNT counter.
4	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

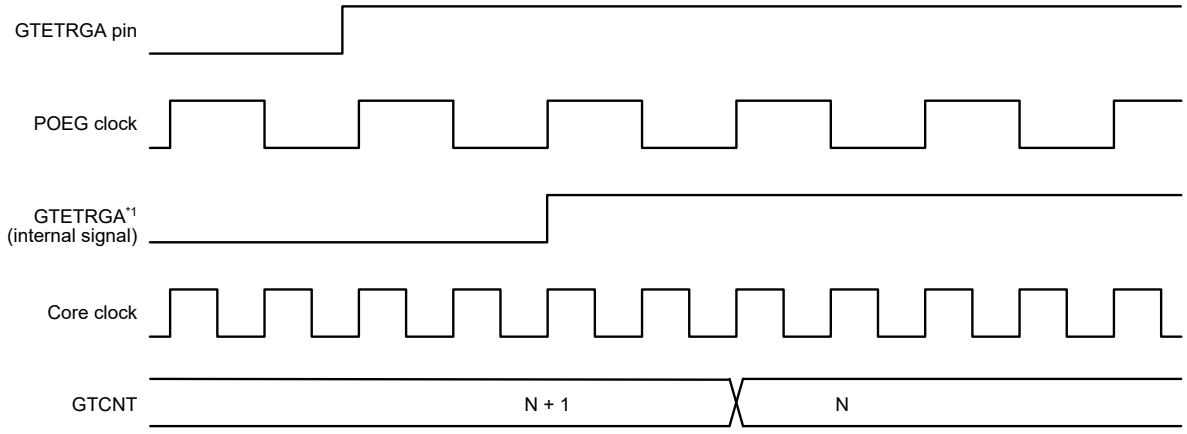
(5) Event count operation in down-counting using hardware sources

The GTCNT counter in each channel can perform down-counting using hardware sources set in the GTDNSR.

When GTDNSR is set to enable, the count clock selected in GTCR.TPCS[3:0] and the count direction selected in GTUDDTYC.UD are ignored. If up-counting and down-counting using hardware sources occur at the same time, the GTCNT counter value does not change. The underflow behavior when down-counting using hardware sources is the same as when down-counting by the count clock.

When GTCR.CST bit is set to 1 to count down using hardware sources, the count operation is enabled. After GTCR.CST is set to 1, the counter cannot count down for 1 clock cycle as specified in GTCR.TPCS[3:0] because the count operation is synchronized with the count clock selected in GTCR.TPCS[3:0]. Set GTCR.TPCS[3:0] to 0x0 to count down with a 1 GTCLK delay after GTCR.CST is set to 1.

Figure 20.9 shows an example of a event count operation in down-counting by a hardware resource (rising edge of GTETRGA pin).



Note: POEG clock: PCLKB, Core clock: PCLKD
Note 1. The GTETRGA internal signal reflects the signal being input to the GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 20.9 Example of event count operation in down-counting using hardware sources

Table 20.14 shows an example for setting a periodic count operation in down-counting using a hardware source.

Table 20.14 Example for setting an event count operation in down-counting using hardware sources

No.	Step Name	Description
1	Set count source	Select the counting-down hardware source with the GTDNSR register.
2	Set cycle	Set the cycle in the GTPR register.
3	Set initial value for counter	Set the initial value in the GTCNT counter.
4	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

(6) Counter clear operation

The counter of each channel is cleared by following sources:

- Writing 0 to GTCNT register
- Writing 1 to the bit in GTCLR associated with the GPT channel number when the GTCSR.CCLR bit set to 1
- The hardware source selected in GTCSR register.

Writing to the GTCNT register is prohibited during count operation. Write access during counting (when CST = 1) is disabled. The GTCNT counter can be cleared both by writing 1 to the GTCLR and by the clear request of hardware sources, whether GTCNT is counting (GTCR.CST is 1) or not (GTCR.CST is 0).

When the count direction flag is set as decrement (GTST.TCUF flag = 0) in saw-wave PWM mode 1 and saw-wave one-shot pulse mode selected with the GTCR.MD[3:0] bits, the GTCNT register is set to the value of the GTPR register when writing 1 to the GTCLR register and when clearing by hardware sources are performed.

In modes other than saw-wave PWM mode 1, saw-wave one-shot pulse mode, and down-counting, the GTCNT register is set to 0 when writing 1 to the GTCLR register and when clearing by hardware sources are performed.

In event count operation when at least 1 bit in the GTUPSR or GTDNSR is set to 1, after clear sources occur, both writing to GTCLR register and clearing by hardware sources are performed immediately to synchronize with GTCLK. If other settings are used, clear is synchronized with the counter clock selected in GTCR.TPCS[3:0].

20.3.1.2 Waveform Output by Compare Match

Compare match means that the GTCNT counter value matches the value of GTCCRA or GTCCRB. When a compare match occurs, the compare match flag is generated synchronously with the count clock, including the event count. At the same time, the GPT can output low, high, or toggled output from the associated GTIOCnA or GTIOCnB output pin (n = 0 to 3). In addition, the GTIOCnA or GTIOCnB pin output can be low, high, or toggled at the cycle end which is determined by

GTPR or GTCCRm selected as a counter clear source by the GTCSR.CSCMSC[2:0] bits in saw-wave PWM mode 2 (n = 0 to 3, m = A to F).

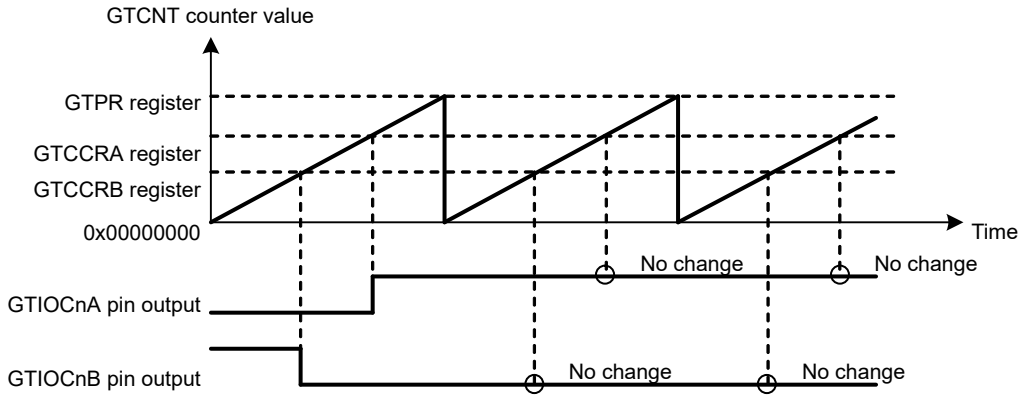
The cycle end is:

- For saw waves (except saw-wave PWM mode 2) in up-counting – when GTCNT changes from the GTPR value to 0 (overflow)
- For saw waves (except saw-wave PWM mode 2) in down-counting – when GTCNT changes from 0 to GTPR value (underflow)
- For saw-wave PWM mode 2 and GTCCRm register (m = A to F) selected as a counter clear source by the GTCSR.CSCMSC[2:0] bits – When the GTCNT counter value changes from the GTCCRm register value to 0x0000 0000
- For saw waves – when the GTCNT counter is cleared
- For triangle waves or complementary PWM mode – when the GTCNT changes from 0 to 1 (trough).

(1) Low output and high output

Figure 20.10 shows an example of low output and high output operation by a compare match of GTCCRA and GTCCRB.

In this example, the GTCNT counter performs up-counting, and settings are made so that high is output from the GTIOCnA pin by a GTCCRA compare match, and low is output from the GTIOCnB pin by a GTCCRB compare match. The pin level does not change when the specified level and pin level match.



[Setting examples]
GTIOR.GTIOA[4:0] bits: Initial output is low, high output at compare match, output retained at cycle end
GTIOR.GTIOB[4:0] bits: Initial output is high, low output at compare match, output retained at cycle end

Note: The base address of registers are the same.

Figure 20.10 Example of low output and high output operation

Table 20.15 shows an example for setting low output and high output operation.

Table 20.15 Example for setting low output and high output operation (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.10, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.10, after 11b is set in the GTUDDTYC.UDF and UD bits, 01b is set in the GTUDDTYC.UDF and UD bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode 2, set the cycle in the GTPR register. In the saw-wave PWM mode 2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.

Table 20.15 Example for setting low output and high output operation (2 of 2)

No.	Step Name	Description
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 20.10, GTIOA[4:0] = 00010b, GTIOB[4:0] = 10001b.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set compare match value*1	Set compare match values in the GTCCRA and GTCCRB registers.
9	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

Note: n: 0 to 3
m: A, B

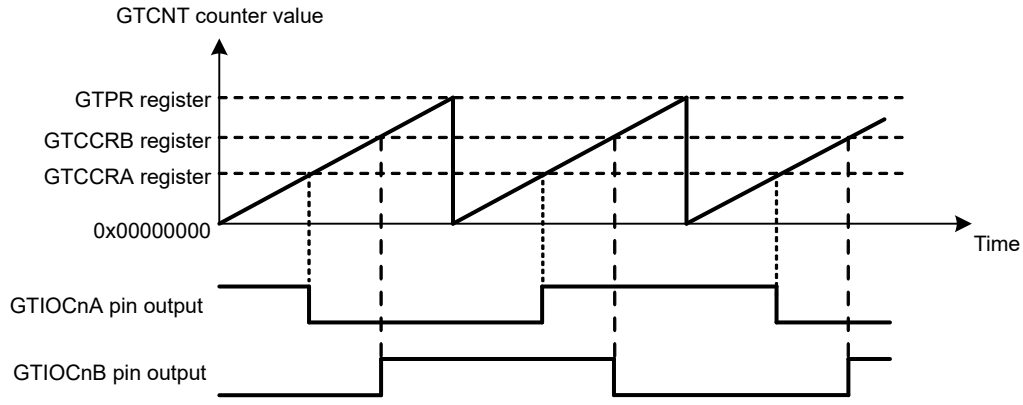
Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

(2) Toggled output

Figure 20.11 and Figure 20.12 show examples of toggled output operation by compare matches of GTCCRA and GTCCRB.

In Figure 20.11, the GTCNT counter performs up-counting, and settings are made so that the GTIOCnA pin output by a GTCCRA compare match and GTIOCnB pin output by a GTCCRB compare match are toggled.

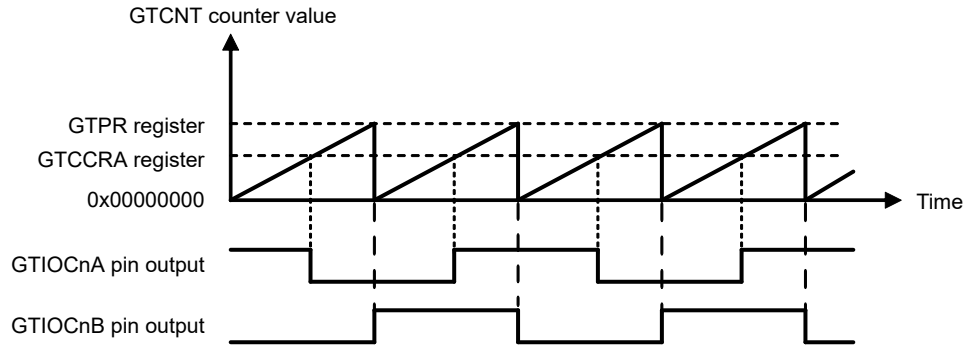
In Figure 20.12, the GTCNT counter performs up-counting, and settings are made so that a GTCCRA compare match toggles the GTIOCnA pin output level and a cycle end toggles the GTIOCnB pin output level.



[Setting examples]
GTIOR.GTIOA[4:0] bits: Initial output is high, output toggled at compare match, output retained at cycle end
GTIOR.GTIOB[4:0] bits: Initial output is low, output toggled at compare match, output retained at cycle end

Note: The base address of registers are the same.

Figure 20.11 Example of toggled output operation (1)



[Setting examples]
GTIOR.GTIOA[4:0] bits: Initial output is high, output toggled at compare match, output retained at cycle end
GTIOR.GTIOB[4:0] bits: Initial output is low, output retained at compare match, output toggled at cycle end

Note: The base address of registers are the same.

Figure 20.12 Example of toggled output operation (2)

Table 20.16 shows an example for setting toggled output operation.

Table 20.16 Example for setting toggled output operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.11 and Figure 20.12, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.11 and Figure 20.12, after 11b is set in the GTUDDTYC.UDF and UD bits, 01b is set in the GTUDDTYC.UDF and UD bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode 2, set the cycle in the GTPR register. In the saw-wave PWM mode 2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 20.11, GTIOA[4:0] = 10011b, GTIOB[4:0] = 00011b, and in Figure 20.12, GTIOA[4:0] = 10011b, GTIOB[4:0] = 01100b.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set compare match value*1	Set compare match values in the GTCCRA and GTCCRB registers.
9	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

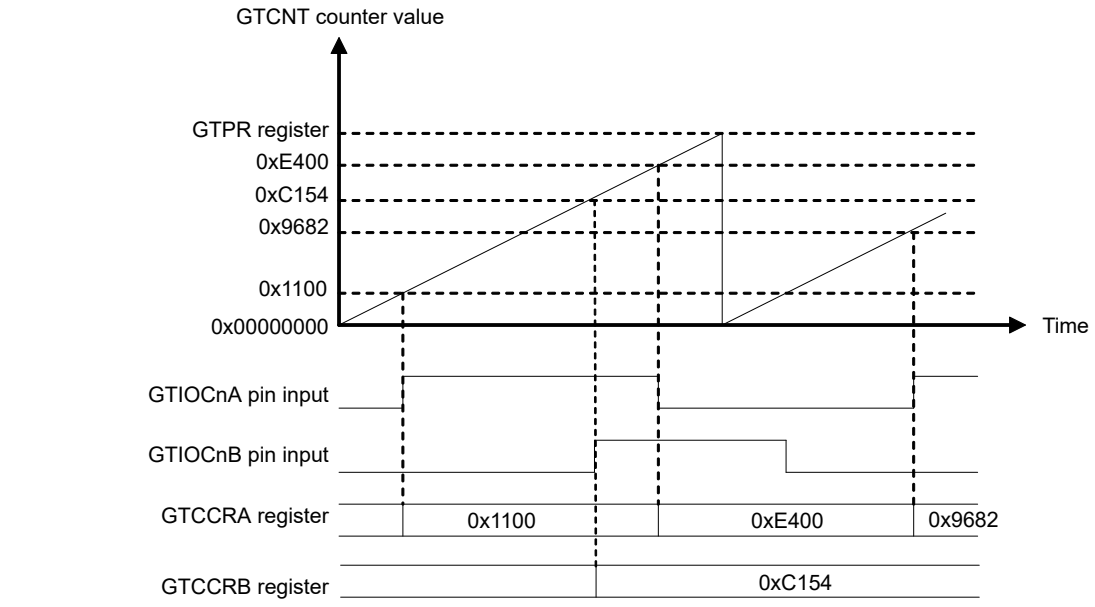
Note: n: 0 to 3
m: A, B
Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

20.3.1.3 Input Capture Function

The GTCNT counter value can be transferred to either GTCCRA or GTCCRB on detection of the hardware source that is set in GTICASR and GTICBSR. In complementary PWM mode, GTCCRA and GTCCRB registers do not function as input capture registers.

Figure 20.13 shows an example of the input capture function.

In this example, the GTCNT counter performs up-counting by the count clock, and settings are made so that an input capture is performed to GTCCRA at both edges of the GTIOCnA input pin and to GTCCRB on the rising edge of the GTIOCnB input pin.



Note: The base address of registers are the same.
Note: [Setting examples]
GTICASR setting: input capture at both edges
GTICBSR setting: input capture at the rising edge

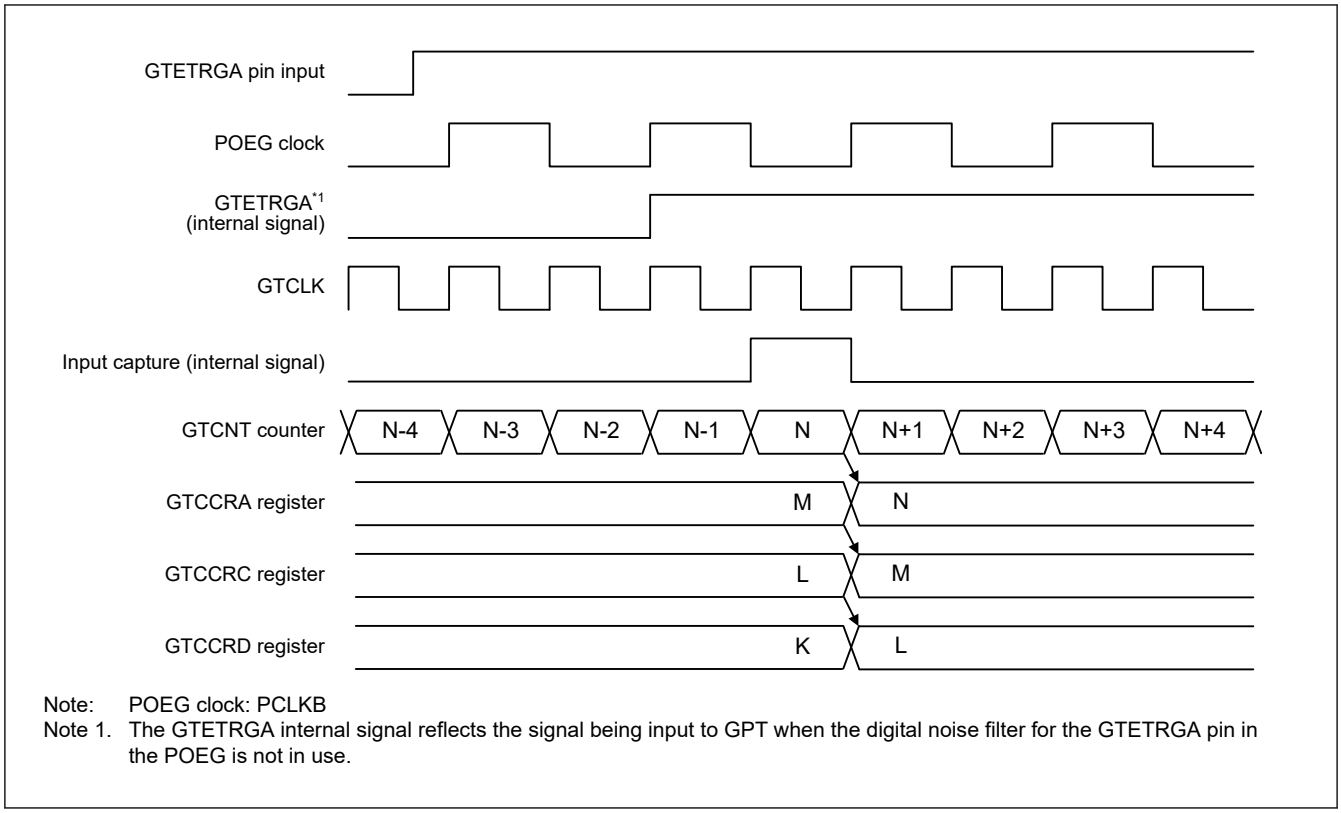
Figure 20.13 Example of input capture operation

Table 20.17 and Table 20.21 show the example for setting an input capture operation with count operation by the count clock.

Table 20.17 Example for setting input capture operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.13, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.13, after 11b is set in the GTUDDTYC.UDF and UD bits, 01b is set in the GTUDDTYC.UDF and UD bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode 2, set the cycle in the GTPR register. In the saw-wave PWM mode 2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Select input capture source	Select the input capture source in the GTICASR and GTICBSR registers. In Figure 20.13, GTICASR = 0x00000F00, GTICBSR = 0x00003000. When input capture by other channel sources is used, set the GTICmSR.mSOC bit (m = A or B) to 1 and allow input capture by other channel sources, and select the group that cooperates with other channel sources by the GTICCR register.
7	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

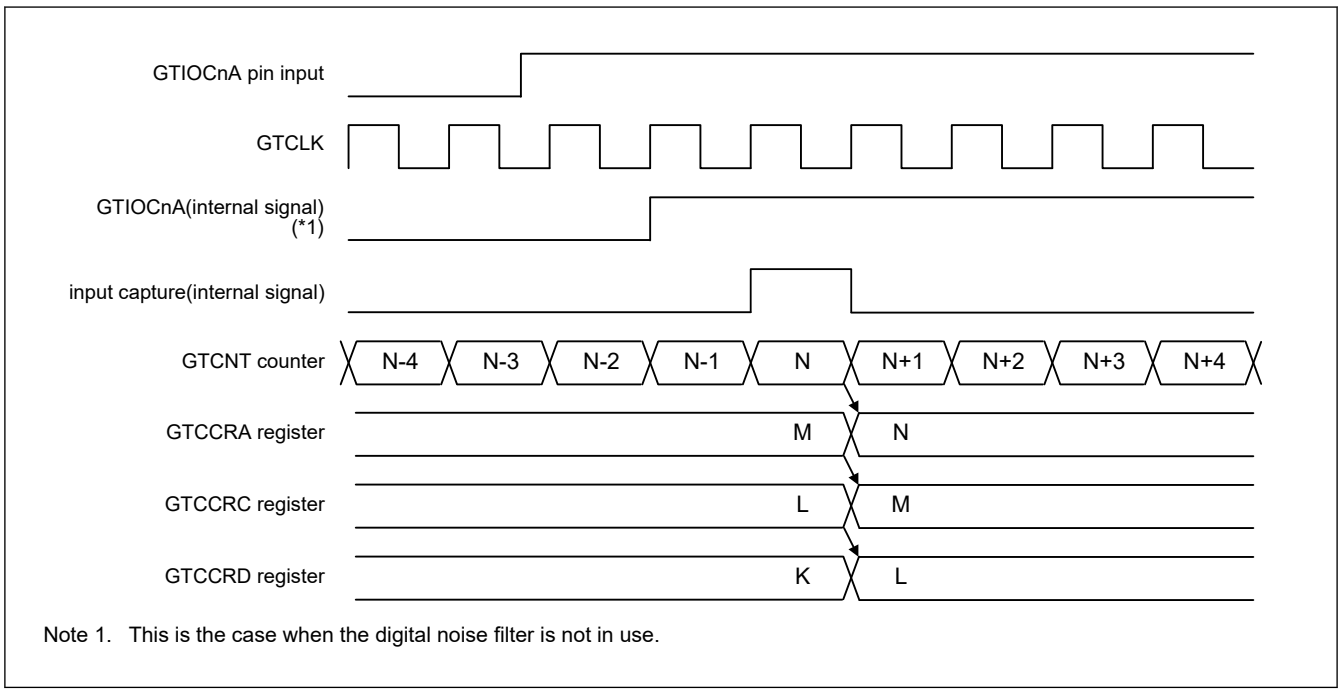
Figure 20.14 shows an example of the timing of input capture operation in response to a rising edge of the input on the GTETRGA pin.



Note: POEG clock: PCLKB
Note 1. The GTETRGA internal signal reflects the signal being input to GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 20.14 Example of the timing of input capture operation in response to a rising edge of the input on the GTETRGA pin

Figure 20.15 shows an example of the timing of input capture operation in response to a rising edge of the input on the GTIOCnA pin.



Note 1. This is the case when the digital noise filter is not in use.

Figure 20.15 Example of the timing of input capture operation in response to a rising edge of the input on the GTIOCnA pin

Figure 20.16 shows an example of the timing of input capture operation in response to ELC_GPTA event input.

This is an example of capture of the counter value by the GPT161.GTCCRA register in response to an input signal. An event signal is output to the ELC after matches in comparison with the GPT160.GTCCRA register. This is selected as a trigger for output to the GPT161 by the ELC as ELC_GPTA.

The GPT160 compare match A signal is synchronized with GTCLK. The ELC receives the signal in synchronization with PCLKB, and outputs the GPT16 event source A signal after 1 cycle of PCLKB has elapsed.

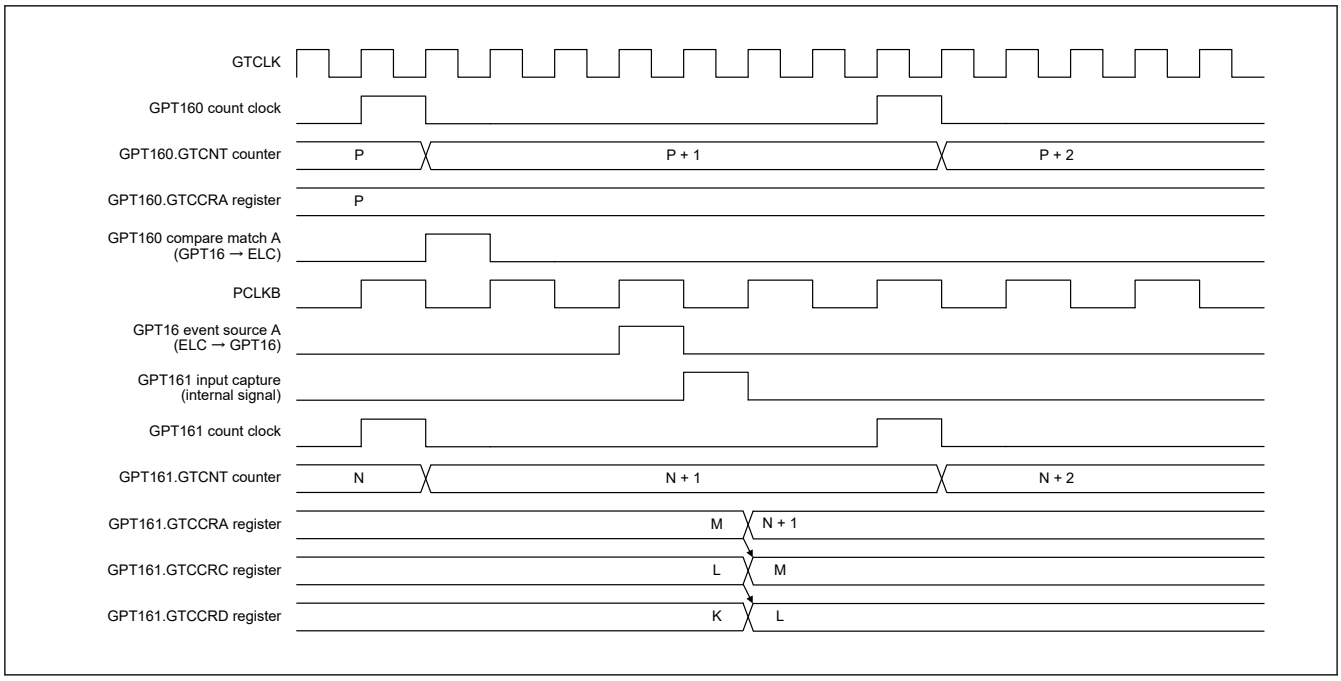


Figure 20.16 Example of the timing of input capture operation in response to ELC_GPTA event input

Figure 20.17 shows an example of the timing of input capture operation in response to count clock from other channel.

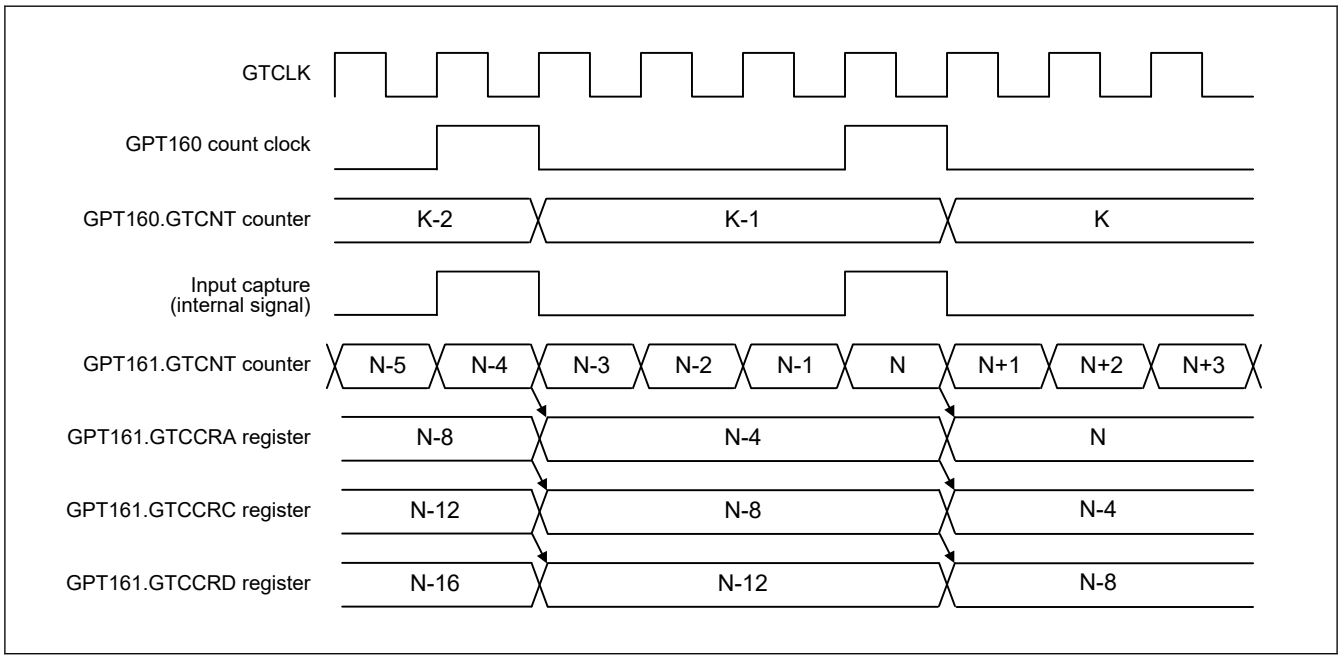


Figure 20.17 Example of the timing of input capture operation in response to count clock from other channel

20.3.2 Buffer Operation

The following buffer operations can be set with GTBER:

- GTPR, GTPBR, and GTPDBR
- GTCCRA, GTCCRC, and GTCCRD
- GTCCRB, GTCCRE, and GTCCRF
- GTADTRA, GTADTBRA, and GTADTDBRA
- GTADTRB, GTADTBRB, and GTADTDBRB

The following buffer operation is enabled by setting GTDTCR:

- GTDVU and GTDBU
- GTDVD and GTDBD

The following buffer operation is enabled by setting GTBER2:

- GTCCRA, GTCCRE, and GTCCRF (in complementary PWM mode 3, 4)
- GTOLBR.GTIOAB[4:0] bits and GTIOR.GTIOA[4:0] bits
- GTOLBR.GTIOBB[4:0] bits and GTIOR.GTIOB[4:0] bits

20.3.2.1 GTPR Register Buffer Operation

GTPBR can function as a buffer register for GTPR.

GTPDBR register can function as a buffer register for the GTPBR register (double-buffer register for the GTPR register).

In complementary PWM mode, the buffer transfer from the GTPDBR register to temporary register P is performed only in the master channel (GPT16n). The temporary register P is transferred to each GTPBR register of master channel, slave channel 1 (GPT16n+1), and slave channel 2 (GPT16n+2). Transfer from the GTPBR register to the GTPR register is made concurrently in three channels. Therefore, the same value is stored in the same register of the three channels. The GTPR register of the master channel represents the GTCNT counter's (GTCNTn) period of the master channel. In the slave channels, periods are controlled using the GTPR register value and the GTDVU register value.

The setting is invalid in saw-wave PWM mode 2.

The buffer transfer is performed at an overflow (during up-counting) or an underflow (during down-counting) in saw-wave mode or in event count, and at a trough in triangle-wave mode.

In saw-wave mode or in event count, the buffer transfer is performed when the following counter clear operations occur during counting:

- Clear by hardware sources (the clear source is selected in GTCSR register)
- Clear by software (when GTCSR.CCLR bit is 1 and GTCLR.CCLRn bit is set to 1, n = 0 to 3).

For saw waves, the buffer transfer by clearing counter can be prohibited by GTBER2.CCTPR bit.

Table 20.18 shows the buffer transfer timing in the complementary PWM mode.

Table 20.18 GTPR buffer transfer timing in complementary PWM mode (1 of 2)

	Complementary PWM mode 1	Complementary PWM mode 2	Complementary PWM modes 3, 4
GTPDBR ↓ Temporary register P	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT16n+2)	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT16n+2)	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT16n+2)
Temporary register P ↓ GTPBR	(1) When data is transferred to temporary register P during up-counting middle section: After one GTCLK cycle from data transfer to temporary register P (2) When data is transferred to temporary register P during a section other than up-counting middle section: At the end of trough section	(1) When data is transferred to temporary register P during down-counting middle section: After one GTCLK cycle from data transfer to temporary register P (2) When data is transferred to temporary register P during a section other than down-counting middle section: At the end of crest section	(1) When data is transferred to temporary register P during middle section: After one GTCLK cycle from data transfer to temporary register P (2) When data is transferred to temporary register P during a section other than middle section: At the end of crest/trough sections

Table 20.18 GTPR buffer transfer timing in complementary PWM mode (2 of 2)

	Complementary PWM mode 1	Complementary PWM mode 2	Complementary PWM modes 3, 4
GTPBR ↓ GTPR	At the end of crest section Counter clear in up-counting middle section and crest section (including counter clear by setting GTCSR.CP1CCE)	At the end of trough section Counter clear in down-counting middle section and trough section	At the end of crest section At the end of trough section Counter clear

To set GTPR to function as a buffer, set the GTBER.PR bit to 1. To set GTPR not to function as a buffer, set the GTBER.PR bit to 0.

To set the GTPR register to function as double buffer, set the GTBER.PR[1:0] bits to 10b or 11b. For single buffer operation, set 01b. Not to function as buffer, set 00b.

In complementary PWM mode, complementary PWM mode-specific buffer operation is performed regardless of the GTBER.PR[1:0] bits setting.

Figure 20.18 to Figure 20.23 show examples of GTPR buffer operation and Table 20.19 shows an example for setting GTPR buffer operation. For details of operation settings in complementary PWM mode, see section 20.3.3.7. Complementary PWM Modes 1, 2, 3.

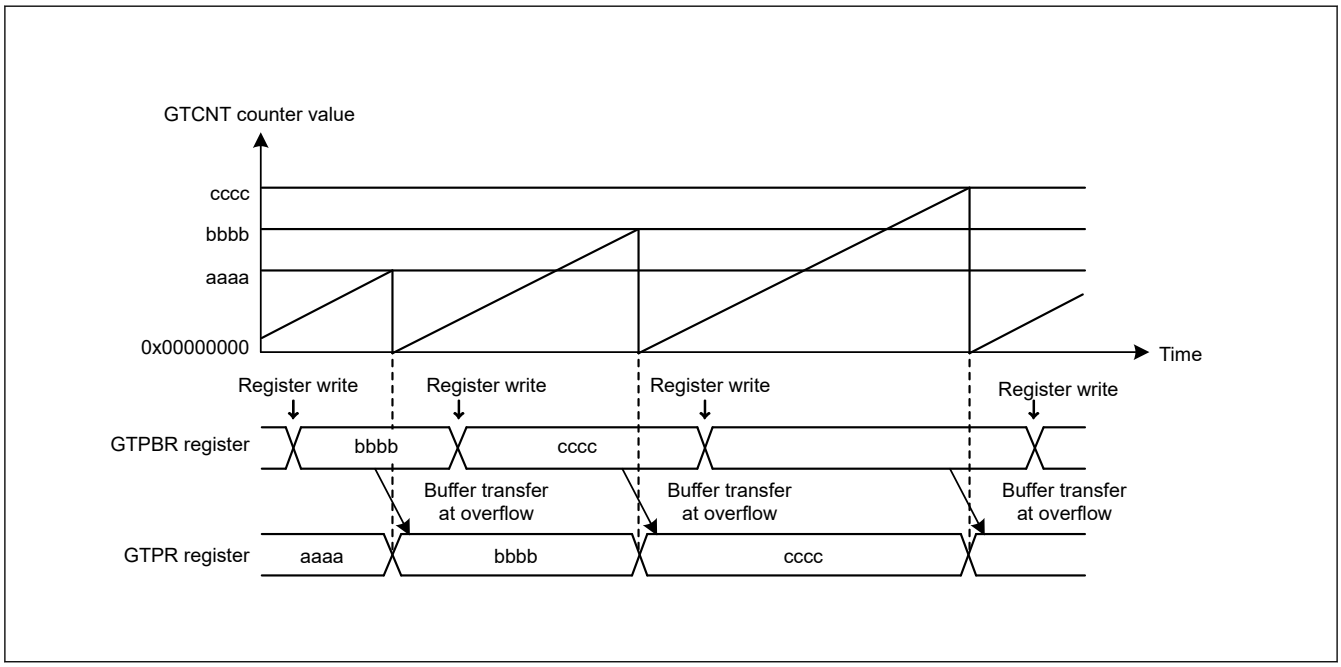


Figure 20.18 Example of GTPR buffer operation with saw waves in up-counting

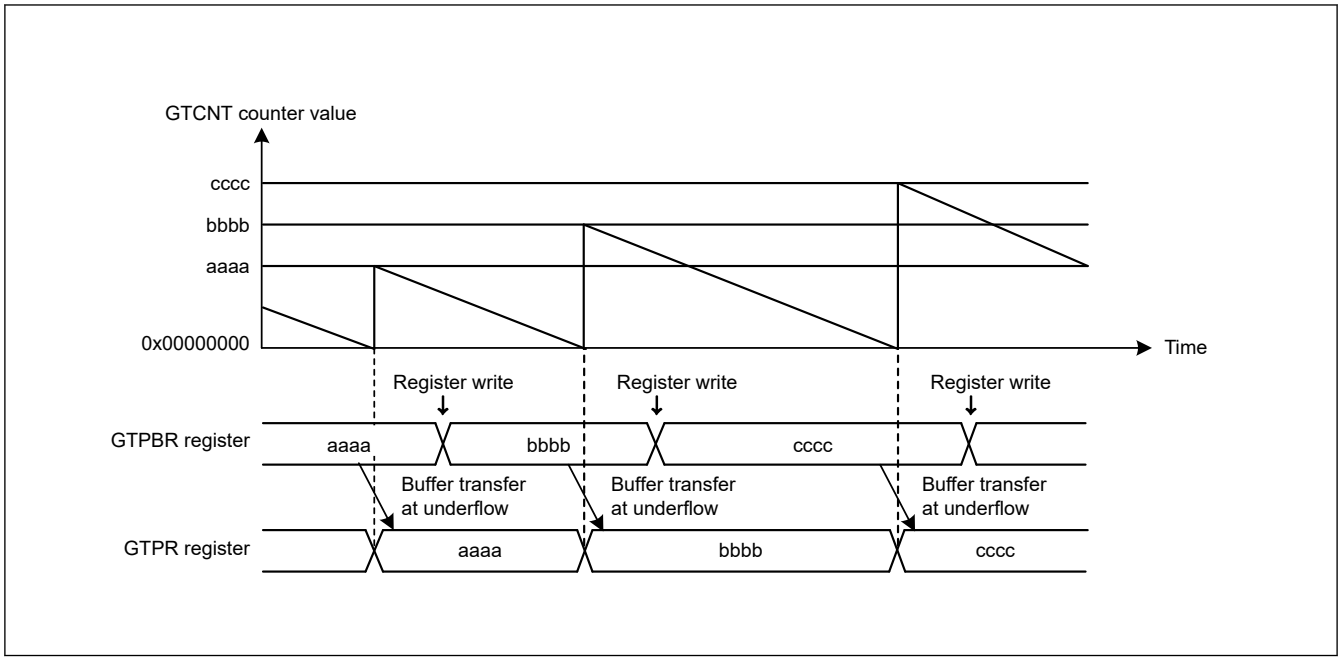


Figure 20.19 Example of GTPR buffer operation with saw waves in down-counting

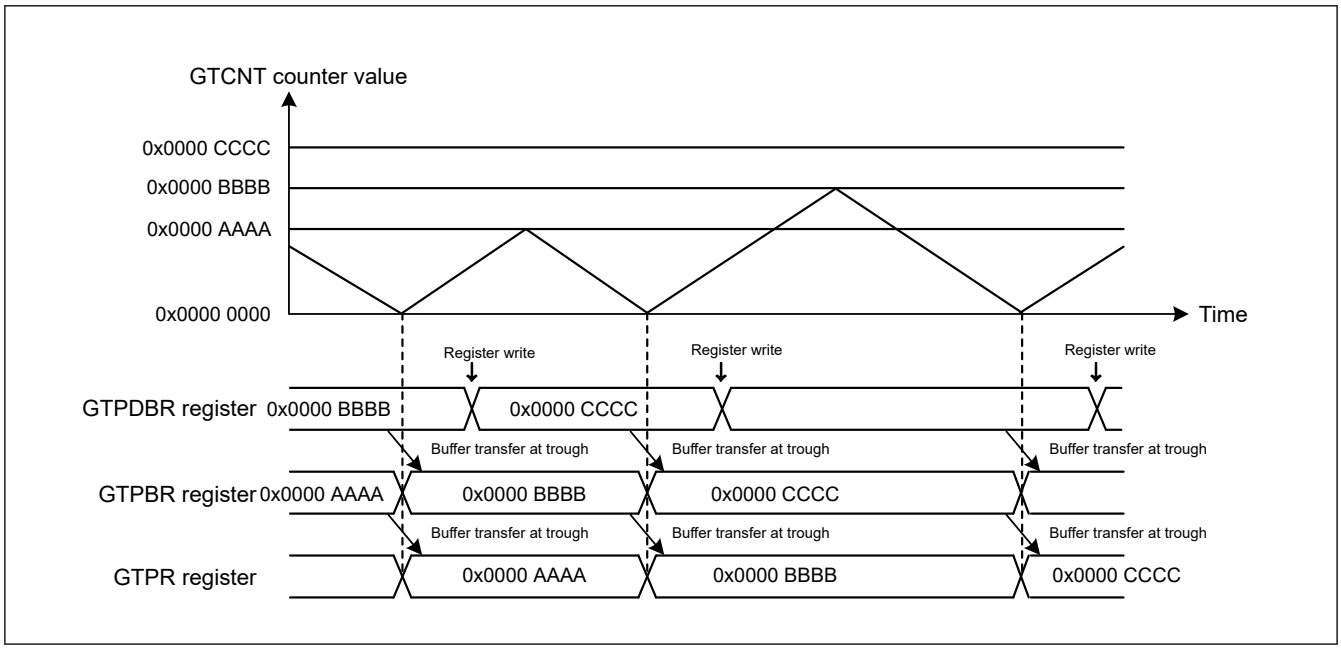


Figure 20.20 Example of GTPR double buffer operation with triangle waves

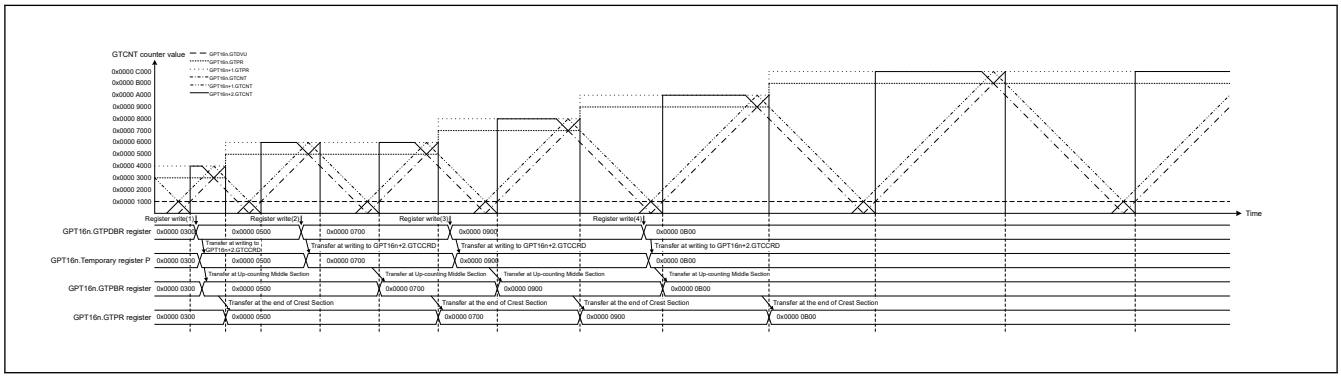


Figure 20.21 Example of GTPR double buffer operation with complementary PWM mode 1

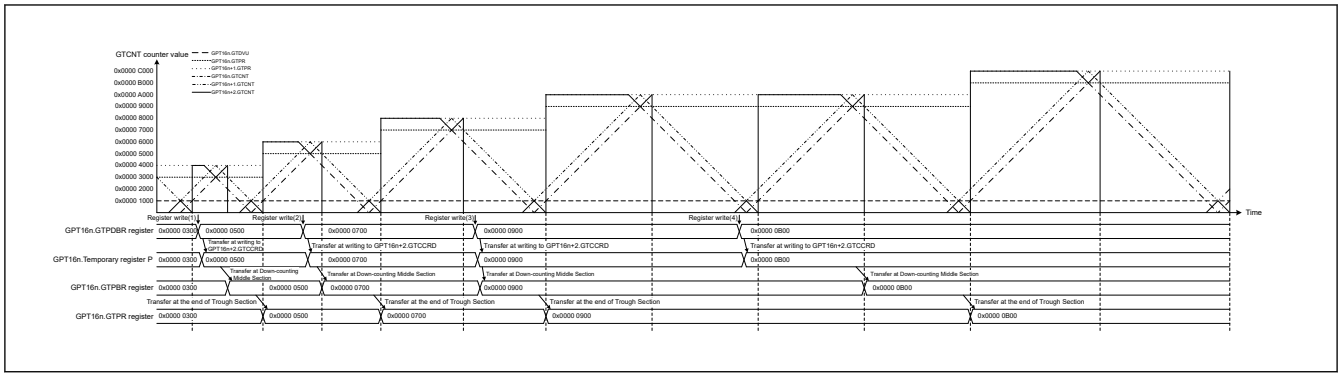


Figure 20.22 Example of GTPR double buffer operation with complementary PWM mode 2

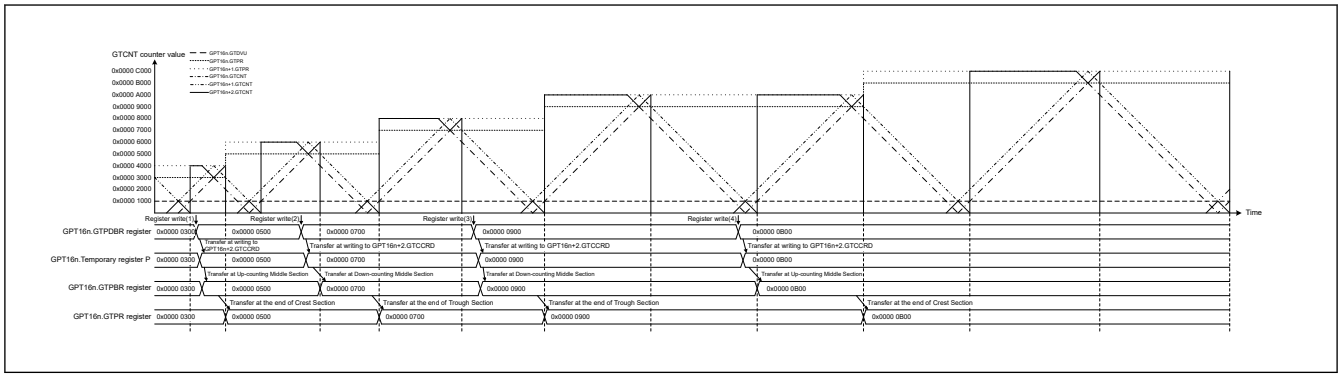


Figure 20.23 Example of GTPR double buffer operation with complementary PWM mode 3, 4

Table 20.19 Example for setting GTPR register buffer operation (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.18 and Figure 20.19, 000b or 0000b (saw-wave PWM mode 1) is set, and in Figure 20.20, 100b or 0100b (triangle-wave PWM mode 1) is set. In Figure 20.21 to Figure 20.23, 11xxb (complementary PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.18, after 11b is set in the GTUDDTYC.UDF and UD bits, 01b is set in the UDF and UD bits (up-counting). In Figure 20.19, after 10b is set in the UDF and UD bits, 00b is set in the UDF and UD bits (down-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle to the GTPR register.
5	Set initial value for counter	Set the initial value to the GTCNT counter.
6	Set buffer operation	Set the buffer operation with the GTBER.PR[1:0] bits. In Figure 20.18 and Figure 20.19, PR[1:0] = 01b. In Figure 20.20, PR[1:0] = 1xb. In Figure 20.21 to Figure 20.23, setting the PR[1:0] bits is not required.
7	Set buffer value	For buffer operation, set the cycle value after one cycle of the current cycle to the GTPBR register. For double buffer operation, also set the cycle value after the next cycle to the GTPDBR register.
8	Start count operation	Set 1 to the GTCR.CST bit and the corresponding channel starts count operation.

Table 20.19 Example for setting GTPR register buffer operation (2 of 2)

No.	Step Name	Description
9	Set buffer value for each cycle	For buffer operation, set a value in one cycle after the current cycle in the GTPBR register. For double buffer operation, also set a value of the period for the cycle after the next cycle in the GTPDBR register. In complementary PWM mode 1 (crest transfer), when the GTPDBR register is set in the up-counting trough or middle section, a value of the period for the next cycle is set. In other sections, a value of the period for the cycle after the next cycle is set. In complementary PWM mode 2 (trough transfer), when the GTPDBR register is set in the up-counting section or the down-counting crest or middle sections, a value of the period for the next cycle is set. In other sections, a value of the period for the cycle after the next cycle is set. In complementary PWM mode 3 or 4 (crest/trough immediate transfer), when the GTPDBR register is set in a section other than the down-counting trough section, a value of the period for the next cycle is set. In the down-counting trough section, a value of the period for the cycle after the next cycle is set.

20.3.2.2 Buffer Operation for GTCCRA and GTCCRB Registers

GTCCRC can function as the GTCCRA buffer register and GTCCRD can function as the GTCCRC buffer register (double-buffer register for GTCCRA). Similarly, GTCCRE can function as the GTCCRB buffer register and GTCCRF can function as the GTCCRE buffer register (double-buffer register for GTCCRB).

To set GTCCRA or GTCCRB to function as a double buffer, set GTBER.CCRA[1:0] or GTBER.CCRB[1:0] to 10b or 11b. For single buffer operation, set 01b. To set GTCCRA or GTCCRB to not function as a buffer, set 00b.

In saw-wave one-shot pulse mode, triangle-wave PWM mode 3, complementary PWM mode, the buffer operations that specific each PWM output operation mode are performed regardless of the setting of GTBER.CCRA [1:0] bits and GTBER.CCRB [1:0] bits.

(1) When GTCCRA or GTCCRB Functions as Output Compare Register

In saw-wave one-shot pulse mode, triangle-wave PWM mode 3, complementary PWM mode, the buffer operations that specific each PWM output operation mode are performed regardless of the setting of GTBER.CCRA [1:0] bits and GTBER.CCRB [1:0] bits. For details, see section 20.3.3. PWM Output Operating Mode. Other than above PWM output operation mode, Buffer transfer occurs in the following situations:

- Buffer transfer by overflow or underflow
  Buffer transfer is performed at an overflow (during up-counting) or an underflow (during down-counting) in saw-wave mode or in event count operation. In triangle-wave mode, buffer transfer is performed at a trough (triangle-wave PWM mode 1) or a crest and trough (triangle-wave PWM mode 2).
- Buffer transfer by counter clear
  In saw-wave mode or in event count operation, during counting, buffer transfer (which is the same as an overflow during up-counting or an underflow during down-counting) is performed by the counter clear sources similar to the case shown in section 20.3.2.1. GTPR Register Buffer Operation.
  In saw-wave, the buffer transfer of GTCCRm register by counter clearing is possible to be disabled with GTBER2.CCTCm bit (m = A, B).
  In triangle-wave mode, buffer transfer is not performed by the counter clear.
- Buffer transfer by compare match
  In saw-wave, the buffer transfer by the compare match of GTCCRm register enabled by GTBER2.CMTCm (m = A, B) bit is performed.
- Forcible buffer transfer
  When GTBER.CCRSWT bit is set to 1 while the count operation is stopped, the GTCCRA and the GTCCRB register buffer transfer are performed forcibly in saw-wave mode, in event count operation and in triangle-wave mode. Additionally buffer transfer from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B are performed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3.

When the setting of the GTBER.DBRTECm (m = A, B) bit is 1 in saw-wave one-shot pulse mode or triangle-wave PWM mode 3, transfer from the intermediate buffers to the GTCCRm (m = A, B) registers is repeated on a cyclic basis even while buffer transfer is disabled by the setting of the GTBER.BD[0] bit or buffer transfer extended skipping function (function for repeated double-buffer operation while buffer transfer is disabled). For details, see section 20.8.2.2. Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.

Figure 20.24 to Figure 20.27 show examples of GTCCRA and GTCCRB buffer operation and Table 20.20 shows an example for setting GTCCRA and GTCCRB buffer operation.

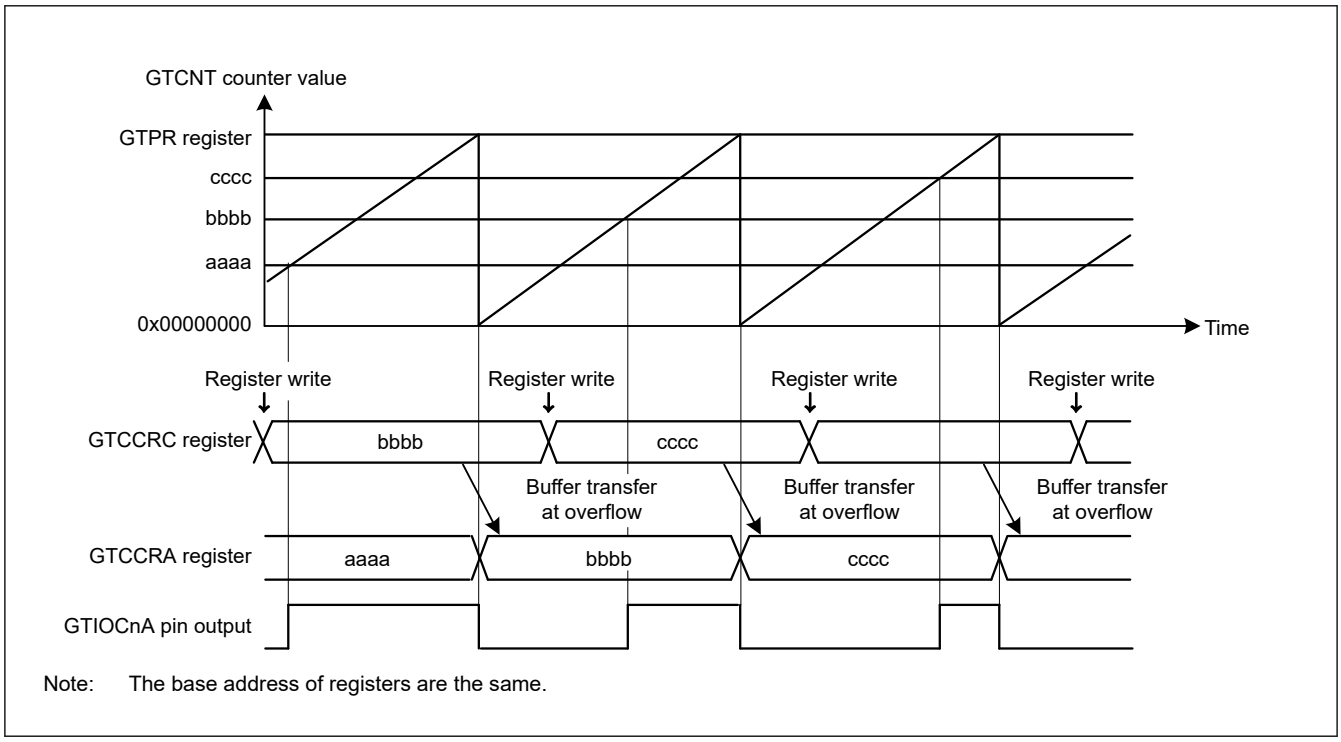


Figure 20.24 Example of GTCCRA and GTCCRB buffer operation with output compare, saw-wave PWM mode 1 in up-counting, high output at GTCCRA compare match, and low output at cycle end

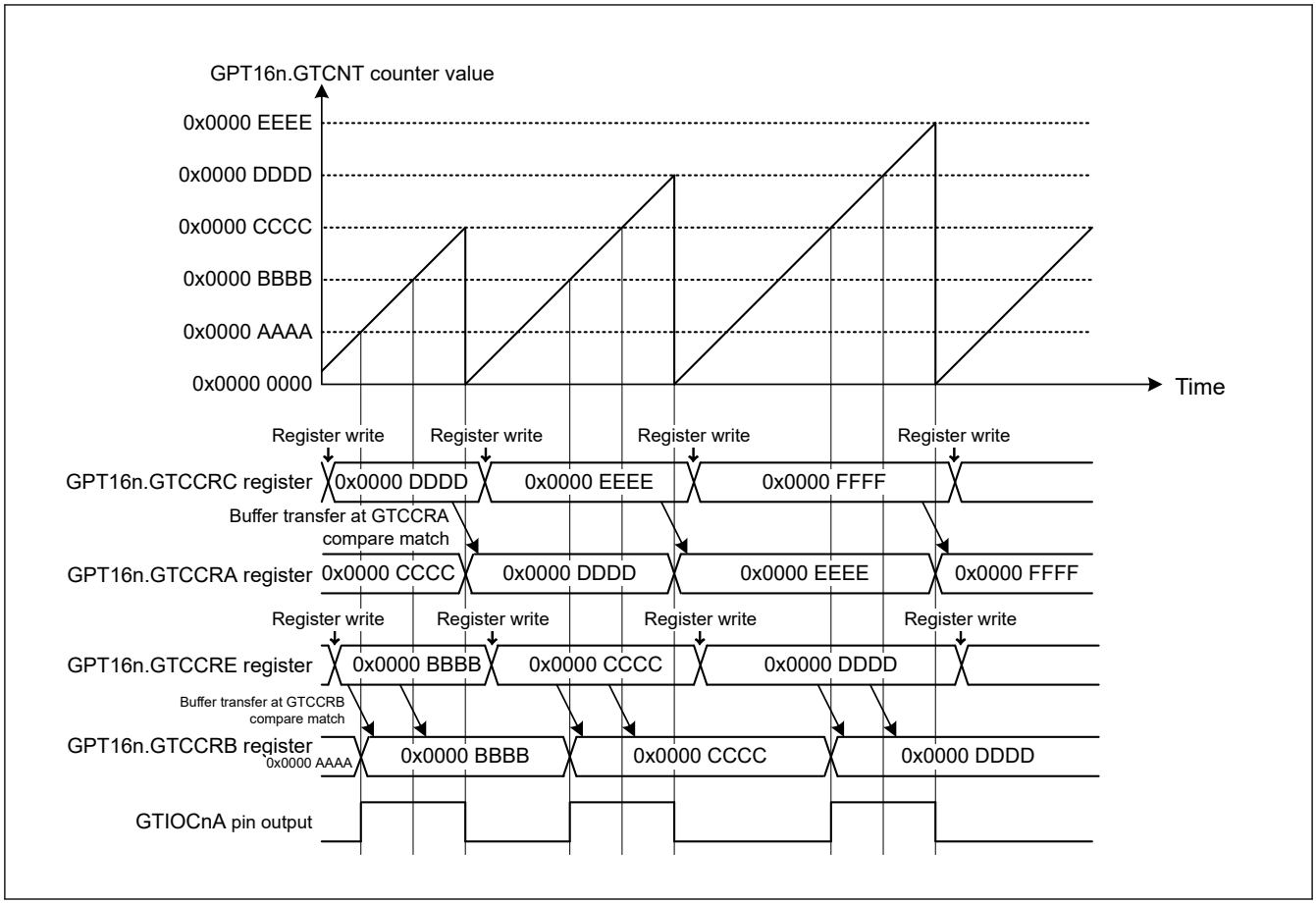
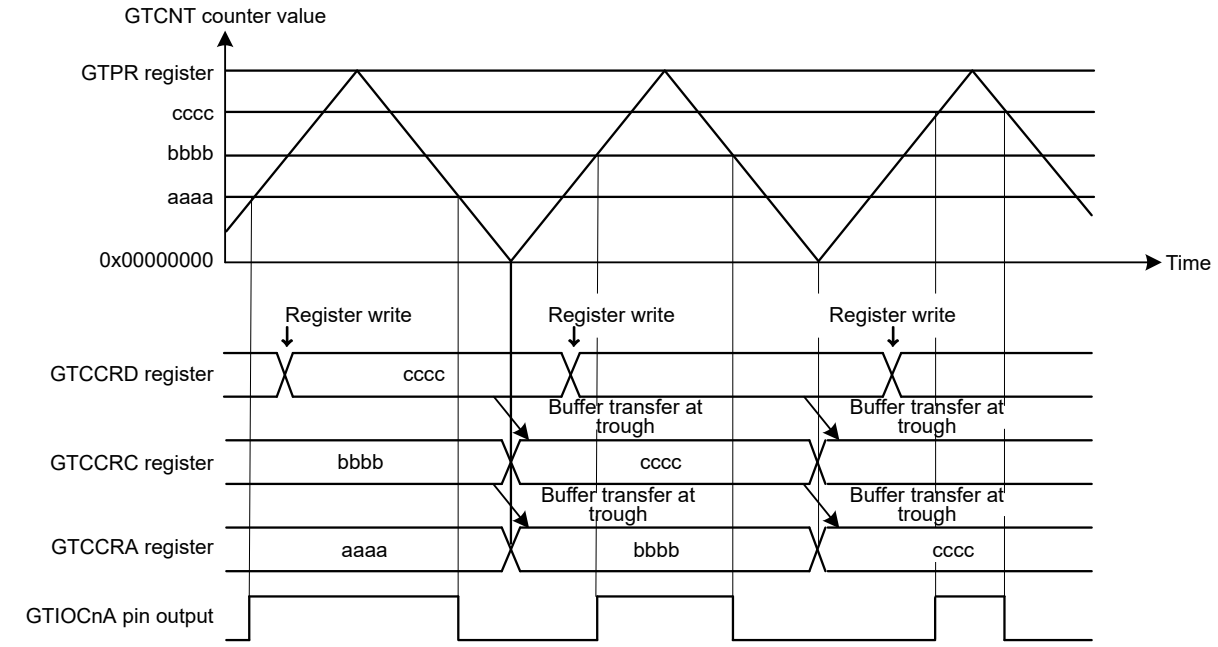
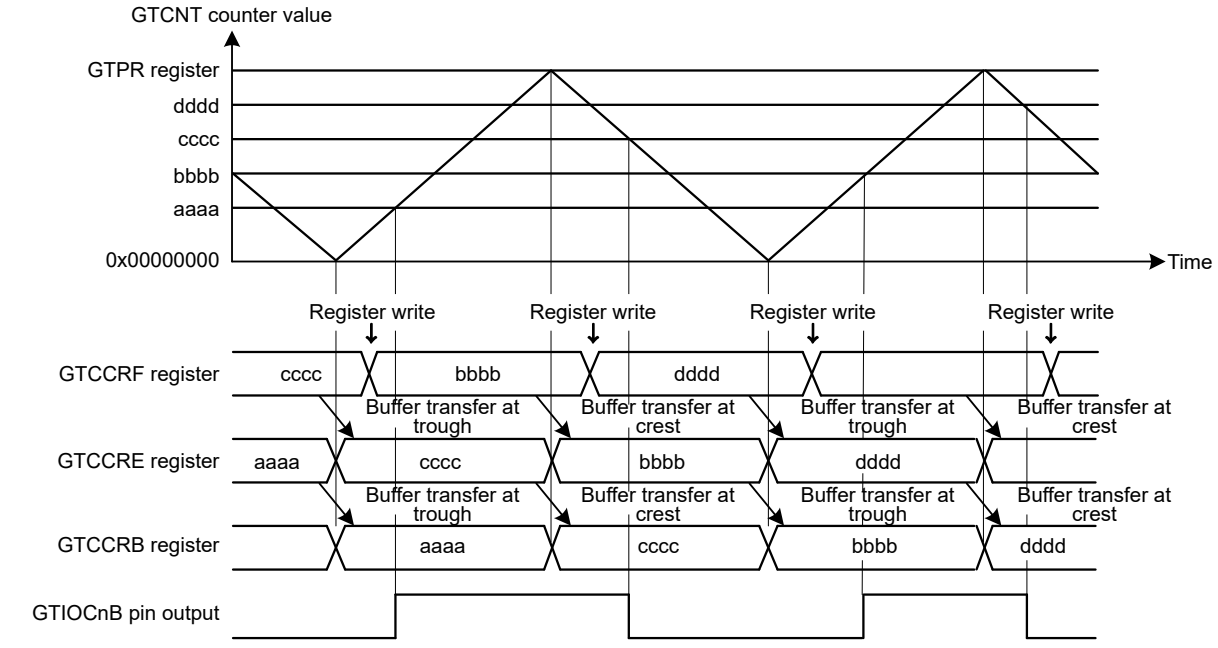


Figure 20.25 Example of GTCCRA and GTCCRB registers buffer operation (output compare, saw-wave PWM mode 2, buffer transfer at GTCCRA register compare match, counter clearing, low output, buffer transfer at GTCCRB register compare match, high output) (n = 0 to 2)



Note: The base address of registers are the same.

Figure 20.26 Example of GTCCRA and GTCCRB double buffer operation with output compare, triangle waves, buffer operation at trough, output toggled at GTCCRA compare match, and output retained at cycle end



Note: The base address of registers are the same.

Figure 20.27 Example of GTCCRA and GTCCRB double buffer operation with output compare, triangle waves, buffer operation at both troughs and crests, output toggled at GTCCRB compare match, and output retained at cycle end

Table 20.20 Example for setting GTCCRA and GTCCRB buffer operation for output compare

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.24, 000b or 0000b (saw-wave PWM mode 1) is set, in Figure 20.25, 0010b (saw-wave PWM mode 2) is set, in Figure 20.26, 100b or 0100b (triangle-wave PWM mode 1) is set, and in Figure 20.27, 101b or 0101b (triangle-wave PWM mode 2) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.24, after 11b is set in the GTUDDTYC.UDF and UD bits, 01b is set in the GTUDDTYC.UDF and UD bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode 2, set the cycle in the GTPR register. In the saw-wave PWM mode 2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 20.24, GTIOA[4:0] = 00110b, in Figure 20.25, GTIOA[4:0] = 00101b and GTIOB[1:0] = 10b, in Figure 20.26, GTIOA[4:0] = 00011b, and in Figure 20.27, GTIOB[4:0] = 00011b.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 20.24, CCRA[1:0] = 01b, in Figure 20.25, CCRA[1:0] = 01b and CCRB[1:0] = 01b, in Figure 20.26, CCRA[1:0] = 1xb, and in Figure 20.27, CCRB[1:0] = 1xb.
9	Set compare match value*1	Set the GTIOCnA pin transition in the GTCCRA register and the GTIOCnB pin transition in the GTCCRB register.
10	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRD and GTCCRF registers, respectively.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRD and GTCCRF registers, respectively.

Note: n: 0 to 3
m: A, B

Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

(2) When GTCCRA or GTCCRB Functions as Input Capture Register

When an input capture is generated, the GTCNT counter value is transferred to GTCCRA and GTCCRB and the stored GTCCRA and GTCCRB register values are transferred to the buffer registers. In input capture operation, the buffer transfer is not performed by the counter clear.

Figure 20.28 and Figure 20.29 show examples of GTCCRA and GTCCRB buffer operation and Table 20.21 shows an example for setting GTCCRA and GTCCRB buffer operation.

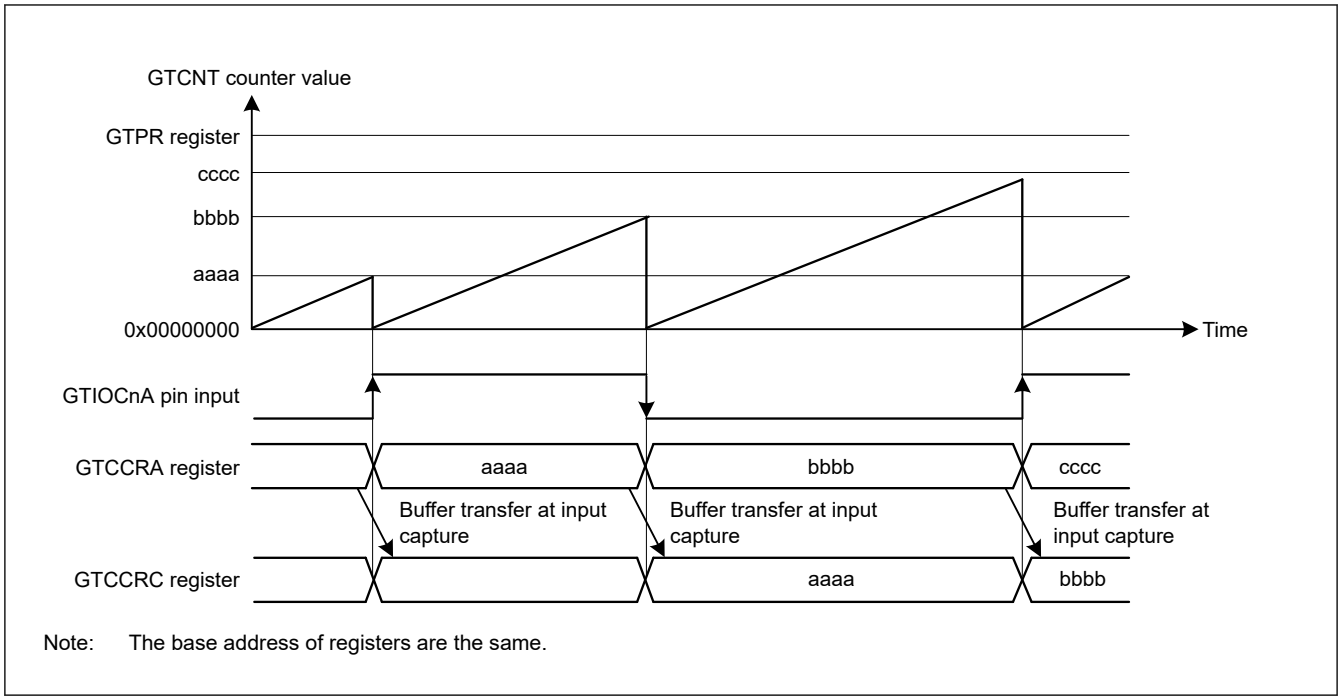


Figure 20.28 Example of GTCCRA and GTCCRB buffer operation with input capture at both edges of GTIOCnA input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOCnA input

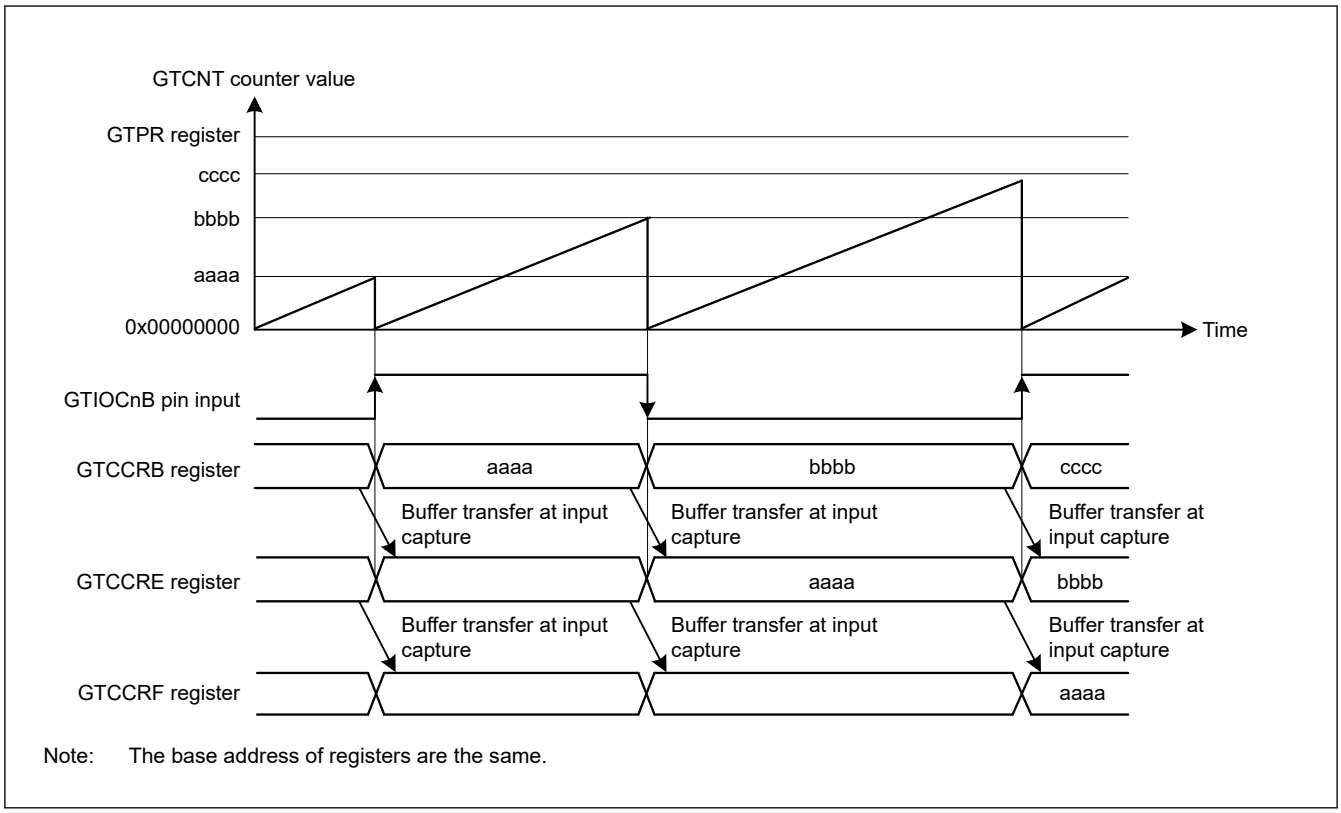


Figure 20.29 Example of GTCCRA and GTCCRB double buffer operation with input capture at both edges of GTIOCnB input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOCnB input

Table 20.21 Example for setting GTCCRA and GTCCRB buffer operation for input capture

No.	Step Name	Description
1	Set operating mode and counter clear sources	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits and count clear source with the GTCSR register. In Figure 20.28, MD[2:0] = 000b or MD[3:0] = 0000b (saw-wave PWM mode 1) and GTCSR = 0x00000F00, and in Figure 20.29, MD[2:0] = 000b or MD[3:0] = 0000b (saw-wave PWM mode 1) and GTCSR = 0x0000F000.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.28, after 11b is set in the GTUDDTYC.UDF and UD bits, 01b is set in the GTUDDTYC.UDF and UD bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode 2, set the cycle in the GTPR register. In the saw-wave PWM mode 2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Select input capture source	Select input capture source in the GTICASR register and GTICBSR register. In Figure 20.28, GTICASR = 0x00000F00, and in Figure 20.29, GTICBSR = 0x0000F000. To perform input capture with sources of other channels, select a group that performs inter-channel cooperation by the GTICCR.ICmGRP bit (m = A or B). For the output-side channel of the input capture sources, set the GTICCR register to enables the input capture sources to output to other channels. For input-side channels, set the GTIOmSR.mSOC bit (m = A or B) to 1 to enable input capture with sources of other channels.
7	Set buffer operation	Set buffer operation with the CCRA and CCRB bits in the GTBER register. In Figure 20.28, CCRA[1:0] = 01b, and in Figure 20.29, CCRB[1:0] = 1xb.
8	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

20.3.2.3 Buffer Operation for GTADTRA and GTADTRB Registers

The GTADTBRA register can function as the GTADTRA buffer register and the GTADTDBRA register can function as the GTADTBRA buffer register (double buffer register for the GTADTRA register). Similarly, the GTADTBRB register can function as the GTADTRB buffer register and the GTADTDBRB register can function as the GTADTBRB buffer register (double buffer register for the GTADTRB register).

To set the GTADTRA or GTADTRB register to function as a double buffer, set the GTBER.ADTDA or ADTDB bit to 1. For single buffer operation, set 0. Not to function as buffer, set the GTBER.ADTTA[1:0] or ADTTB[1:0] bits to 00b.

The buffer transfer timing can be set with the ADTTA[1:0] and ADTTB[1:0] bits to an overflow (in up-counting) or an underflow (in down-counting) in saw-wave mode, with ADTTA[1:0] and ADTTB[1:0] bits to 01b for a crest, to 10b for a trough, or to 11b for both crest and trough in triangle-wave mode or complementary PWM mode.

In saw-wave mode, when the ADTTA[1:0] and ADTTB[1:0] bits are set to value other than 00b and in count operation, the buffer transfer, by similar counter clearing sources in section 20.3.2.1. GTPR Register Buffer Operation, is performed in the same way at an overflow (in up-counting) or an underflow (in down-counting).

In complementary PWM mode, the buffer transfer is performed after one GTCLK cycle from GTCCRD register write of slave channel 2.

In saw-wave mode, the buffer transfer of GTADTRm register by counter clearing can be disabled by GTBER2.CCTADm (m = A, B) bit setting.

In saw-wave mode, the buffer transfer of GTADTRm register by its own compare match can be enabled by the GTBER2.CMTADm (m = A, B) bit setting.

Figure 20.30 to Figure 20.34 show examples of buffer operation of the GTADTRA and GTADTRB registers, and Table 20.22 shows an example for setting buffer operation of the GTADTRA and GTADTRB registers.

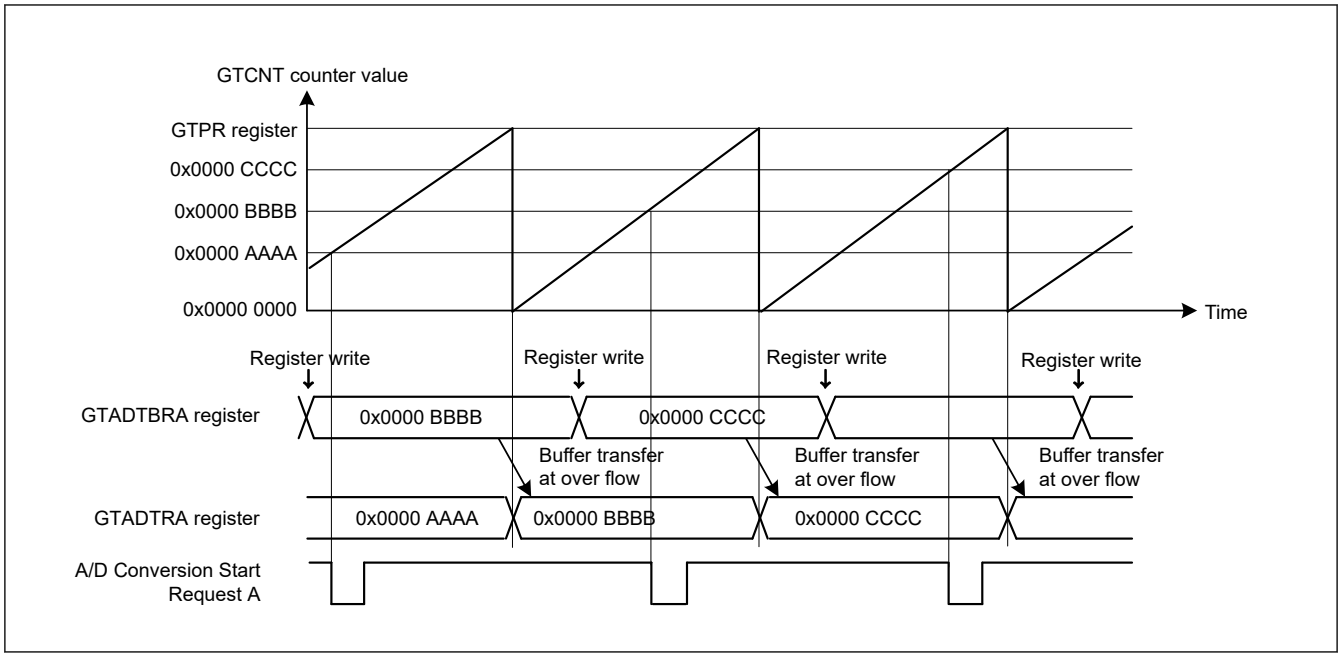


Figure 20.30 Example of buffer operation of the GTADTRA and GTADTRB registers (saw waves in up-counting, A/D conversion start request generated by up-counting)

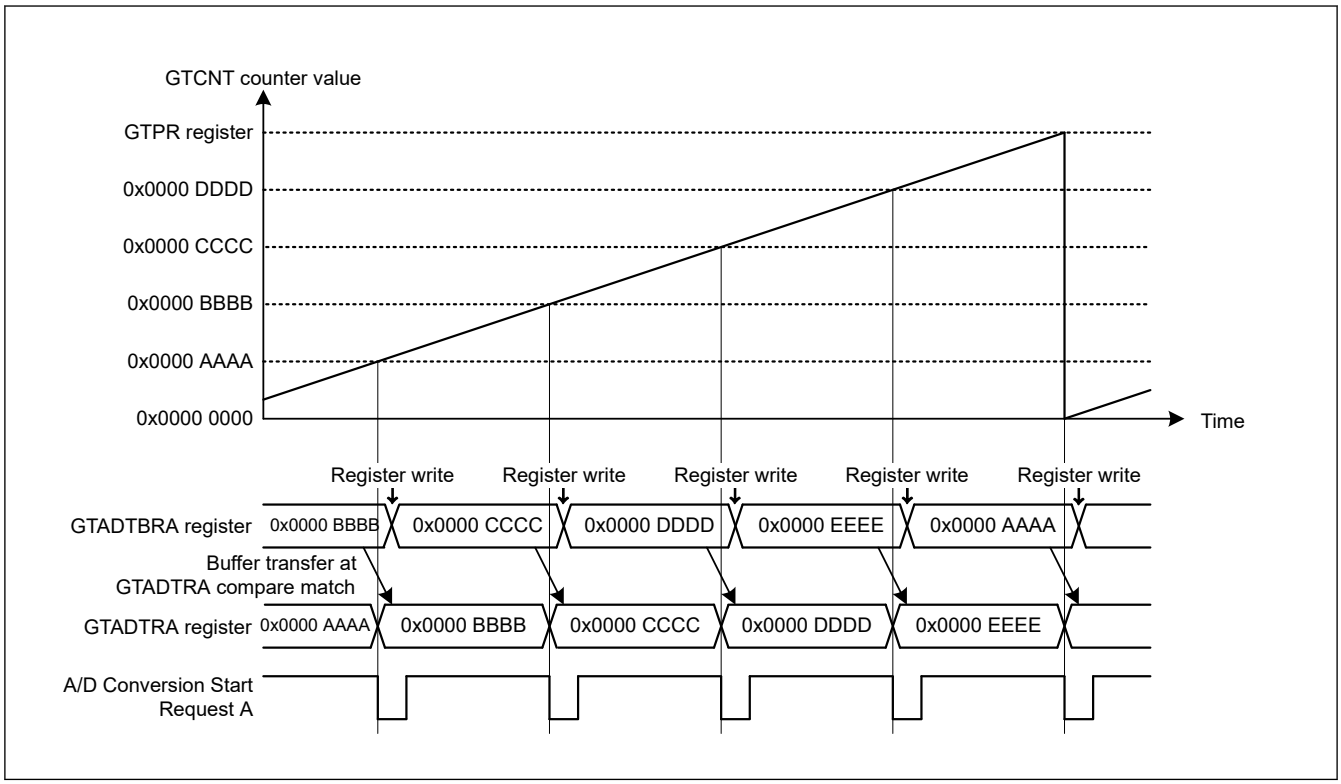


Figure 20.31 Example of buffer operation of the GTADTRA and GTADTRB registers (saw waves in up-counting, buffer transfer at GTADTRA compare match, A/D conversion start request generated by up-counting)

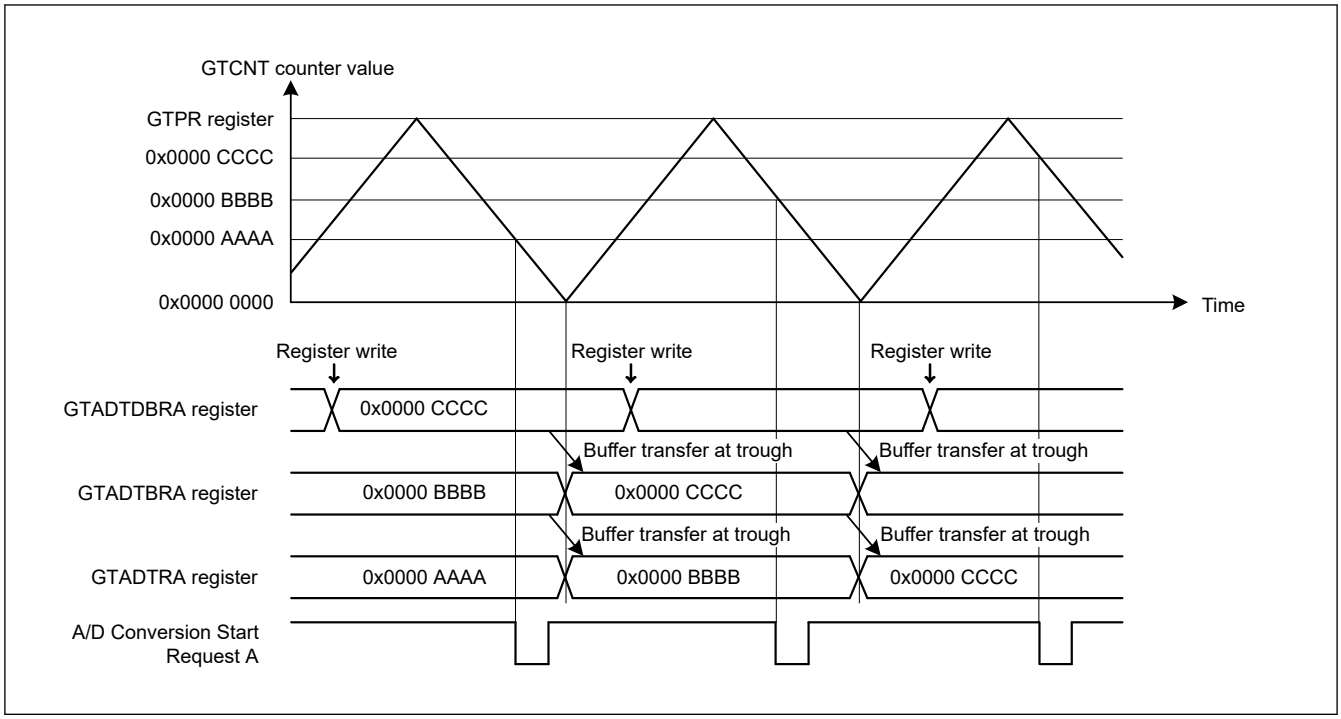


Figure 20.32 Example of double buffer operation of the GTADTRA and GTADTRB registers (triangle waves, buffer transfer at troughs, A/D conversion start request generated by down-counting)

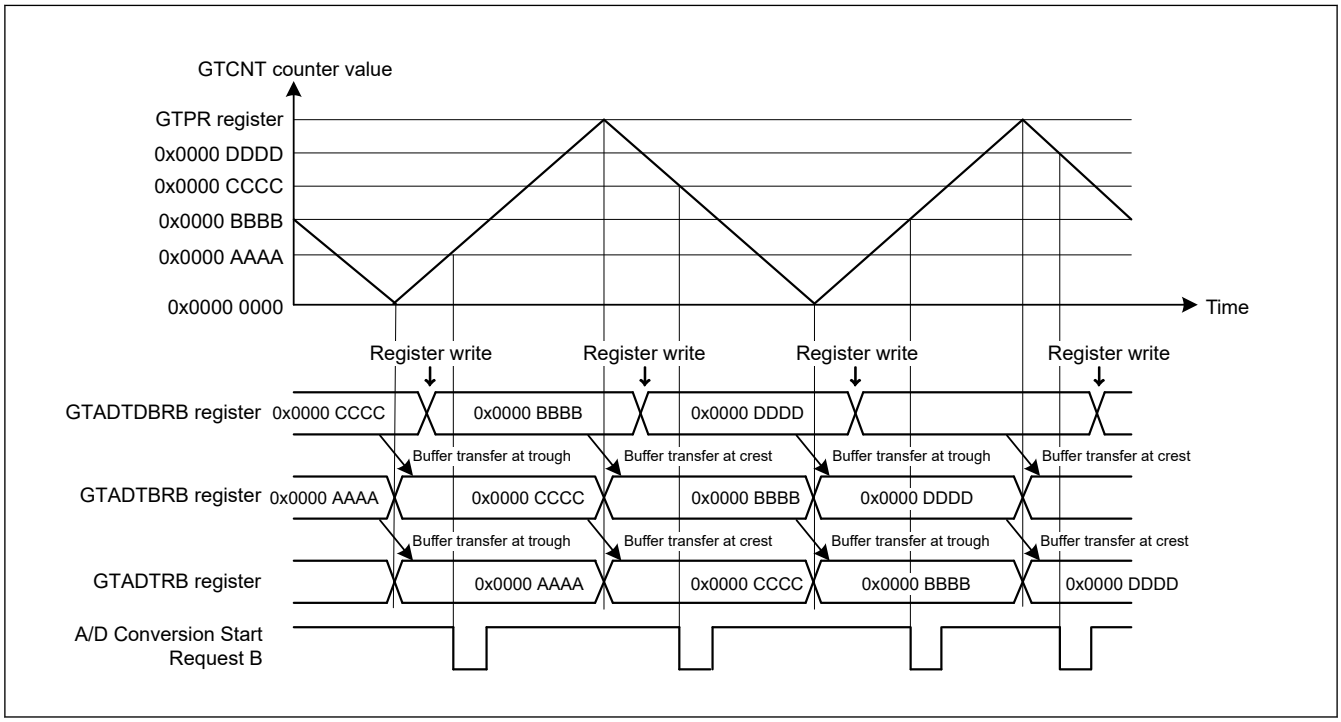


Figure 20.33 Example of double buffer operation of the GTADTRA and GTADTRB registers (triangle waves, buffer transfer at both troughs and crests, A/D conversion start request generated by both up- and down-counting)

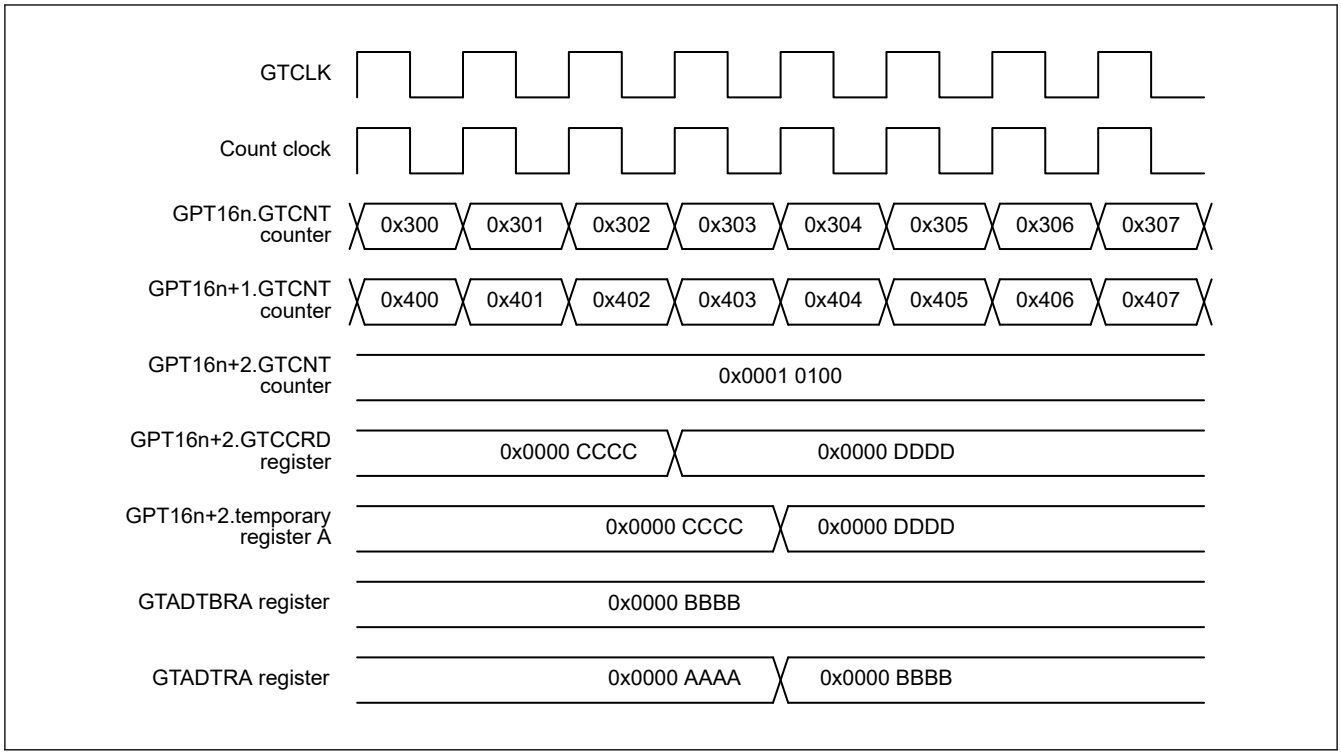


Figure 20.34 Example of buffer operation of the GTADTRA and GTADTRB registers at the GTCCRD register of slave channel 2 updating in complementary PWM mode

Table 20.22 Example for setting buffer operation of the GTADTRA and GTADTRB registers (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.30 and Figure 20.31, 000b or 0000b (saw-wave PWM mode 1) is set, in Figure 20.32 and Figure 20.33, 100b, 101b, 110b, 0100b, 0101b, or 0110b (triangle-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.30 and Figure 20.31, after 11b is set in the GTUDDTYC.UDF and UD bits, 01b is set in the GTUDDTYC.UDF and UD bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Other than the saw-wave PWM mode 2, set the cycle in the GTPR register. In the saw-wave PWM mode 2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set buffer operation	Set buffer operation with the ADTTA[1:0], ADTTB[1:0], ADTDA, and ADTDB bits in GTBER register. To perform buffer transfer at a compare match of the ADTRAm (m = A, B) register, set the GTBER2.CMTADm bit to 1. In Figure 20.30, ADTTA[1:0] bits = 01b, 10b, or 11b and ADTDA bit = 0, in Figure 20.31, CMTADA bit = 1, in Figure 20.32, ADTTA[1:0] bits = 10b and ADTDA bit = 1, and in Figure 20.33, ADTTB[1:0] bits = 11b and ADTDB bit = 1.
7	Set compare match value	Set the A/D conversion start request point in the GTADTRA and GTADTRB registers.
8	Set buffer value	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.

Table 20.22 Example for setting buffer operation of the GTADTRA and GTADTRB registers (2 of 2)

No.	Step Name	Description
9	Enable A/D conversion start request	Set to enable A/D conversion start request with the ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN bits in the GTINTAD register. In Figure 20.30 and Figure 20.31, ADTRAUEN bit = 1, in Figure 20.32, ADTRADEN bit = 1, and in Figure 20.33, ADTRBUEN bit = 1 and ADTRBDEN bit = 1.
10	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
11	Set buffer value of each cycle	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.

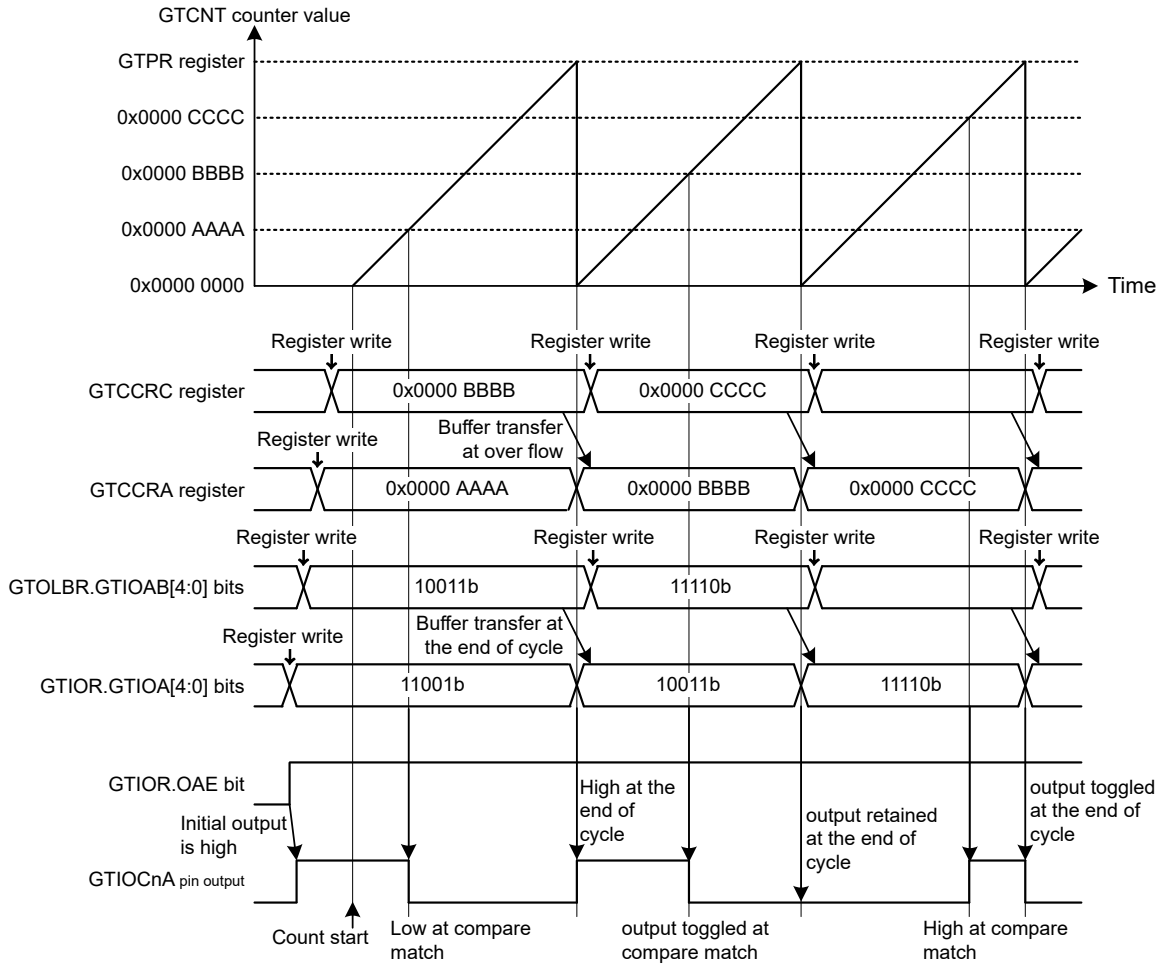
20.3.2.4 Buffer operation for GTIOA and GTIOB

The GTOLBR.GTIOAB[4:0] bits can function as the buffer register of the GTIOR.GTIOA[4:0] bits and the GTOLBR.GTIOBB[4:0] bits can function as the buffer register of the GTIOR.GTIOB[4:0] bits.

Buffer transfer timing can be set with the GTBER2.OLTTm[1:0] bits (m = A, B). It can be selected from the end of cycle or GTCCR register compare match (in saw-wave mode), from crest, trough, or both crest and trough (in triangle-wave mode and complementary PWM mode). In the case that the GTBER2.OLTTm[1:0] bits are 00b, buffer transfer is not performed.

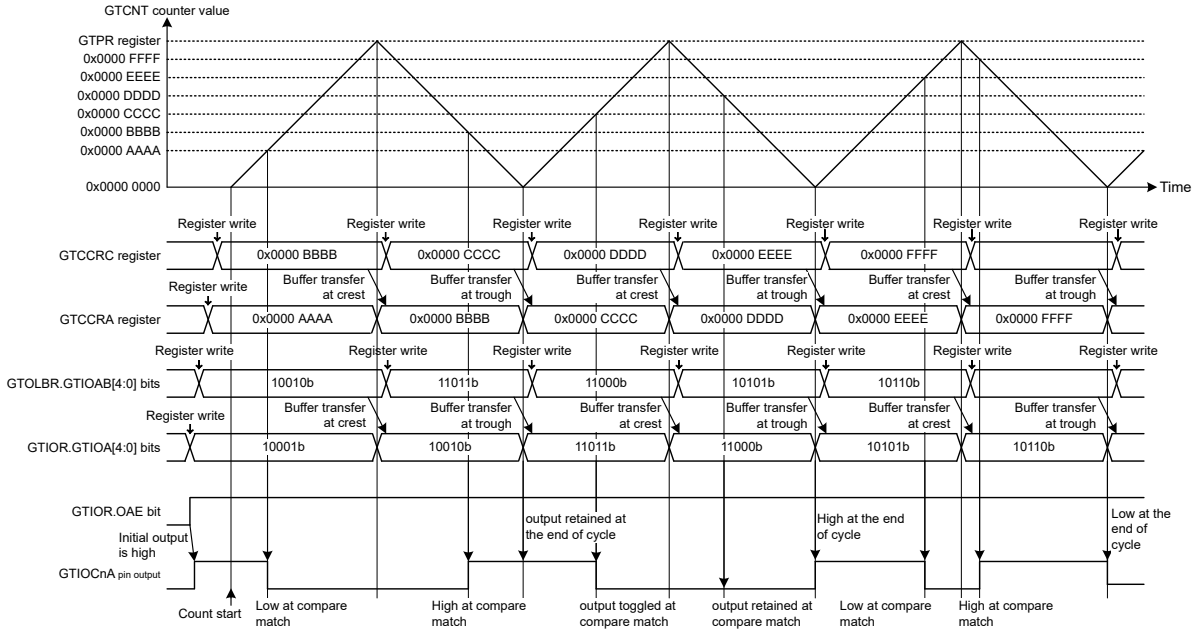
In complementary PWM mode, the setting to convert GTIOA and GTIOB by buffer transfer during the dead time of complementary PWM is prohibited. Therefore, when changing GTIOA and GTIOB by buffer transfer, it is necessary to set the crest and trough timings not to overlap with the dead time and change GTOLBR.GTIOAB and GTOLBR.GTIOBB. When the buffer transfer timing is crest, set the GTCCRm (m = A, C, E) register to satisfy GTCCRm < GTPR. When the buffer transfer timing is trough, set the GTCCRm register to satisfy GTDVU < GTCCRm.

Figure 20.35 to Figure 20.37 show examples of buffer operation of the GTIOR.GTIOA[4:0] and GTIOR.GTIOB[4:0] bits. Table 20.23 shows an example for setting buffer operation of the GTIOR.GTIOA[4:0] and GTIOR.GTIOB[4:0] bits.



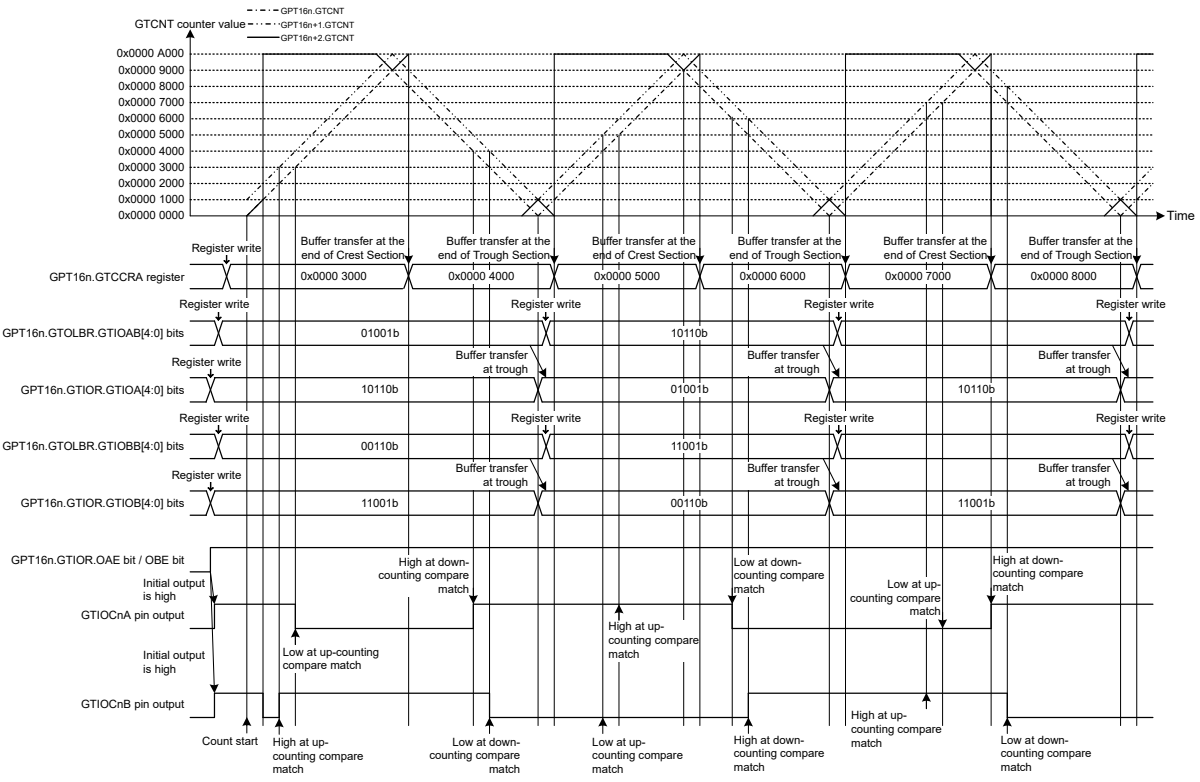
Note: Initial output is synchronized to the count clock.

Figure 20.35 Example of buffer operation of the GTIOA and GTIOB bits (up-counting in saw-wave PWM mode 1, buffer transfer at the end of cycle)



Note: Initial output is synchronized to the count clock.

Figure 20.36 Example of buffer operation of the GTIOA and GTIOB bits (triangle-wave PWM mode 2, buffer transfer at crests and troughs)



Note: Initial output is synchronized to the count clock.

Figure 20.37 Example of buffer operation of the GTIOA and GTIOB bits (complementary PWM mode 3, buffer transfer at troughs)

Table 20.23 Example for setting buffer operation of the GTIOA and GTIOB

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[3:0]. In Figure 20.35, 0000b (saw-wave PWM mode 1) is set, in Figure 20.36, 0101b (triangle-wave PWM mode 2) is set, and in Figure 20.37, 1110b (complementary PWM mode 3) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.35, after 11b is set in GTUDDTYC.UDF and UD, 01b is set in GTUDDTYC.UDF and UD (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits.
4	Set cycle	Other than saw-wave PWM mode 2, set the cycle in GTPR. In saw-wave PWM mode 2, select the counter clear source compare match register GTCCRx (x = A to F) by GTCSR.CSCMSC[2:0] bits and set the cycle in that register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 20.35, GTIOA[4:0] = 11001b, in Figure 20.36, GTIOA[4:0] = 10001b, and in Figure 20.37, GTIOA[4:0] = 10110b and GTIOB[4:0] = 11001b.
7	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with OAE and OBE in GTIOR.
8	Set buffer operation	Set buffer operation with OLTTm[1:0] bits in GTBER2 register. In Figure 20.35, 01b is set in OLTTA[1:0], in Figure 20.36, 11b is set in OLTTA[1:0], in Figure 20.37, 10b is set in OLTTA[1:0] and 10b is set in OLTTB[1:0].
9	Set buffer value	For buffer operation, set the GTIOCnm pin function in one cycle after the current cycle (in saw-wave mode and in triangle-wave/complementary PWM mode with buffer transfer at crest or trough) or half cycle after the current cycle (in triangle-wave/complementary PWM mode with buffer transfer at crest and trough) in the GTOLBR register.
10	Start count operation	Set GTCR.CST to 1 to start count operation.
11	Set buffer value of each cycle	For buffer operation, set the GTIOCnm pin function in one cycle after the current cycle (in saw-wave mode and in triangle-wave/complementary PWM mode with buffer transfer at crest or trough) or half cycle after the current cycle (in triangle-wave/complementary PWM mode with buffer transfer at crest and trough) in the GTOLBR register.

Note: n = 0 to 2
m = A, B

20.3.3 PWM Output Operating Mode

The GPT can output PWM waveforms to the GTIOCnA or GTIOCnB pin (n = 0 to 3) by a compare match between the GTCNT counter and GTCCRA or GTCCRB.

By setting GTDTCR, GTDVU, and GTDVD, the compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

In complementary PWM mode, it is possible to output PWM waveforms (positive phase, negative phase) with dead time that guarantees the linearity of the PWM output pulse width near 0% and 100% duty.

In saw-wave mode other than saw-wave PWM mode 2, or triangle-wave mode, or the master channel of complementary PWM mode, the signal synchronized with the PWM cycle can be output from the GTCPPOn output terminal by setting the GTIOR.PSYE bit to 1. The GTCPPOn output is toggled at the end of cycle in saw-wave mode or at the timing of crest / trough / GTCNT counter clearing in triangle-wave mode or complementary PWM mode. The initial output of GTCPPOn is low, and it becomes high when the count starts.

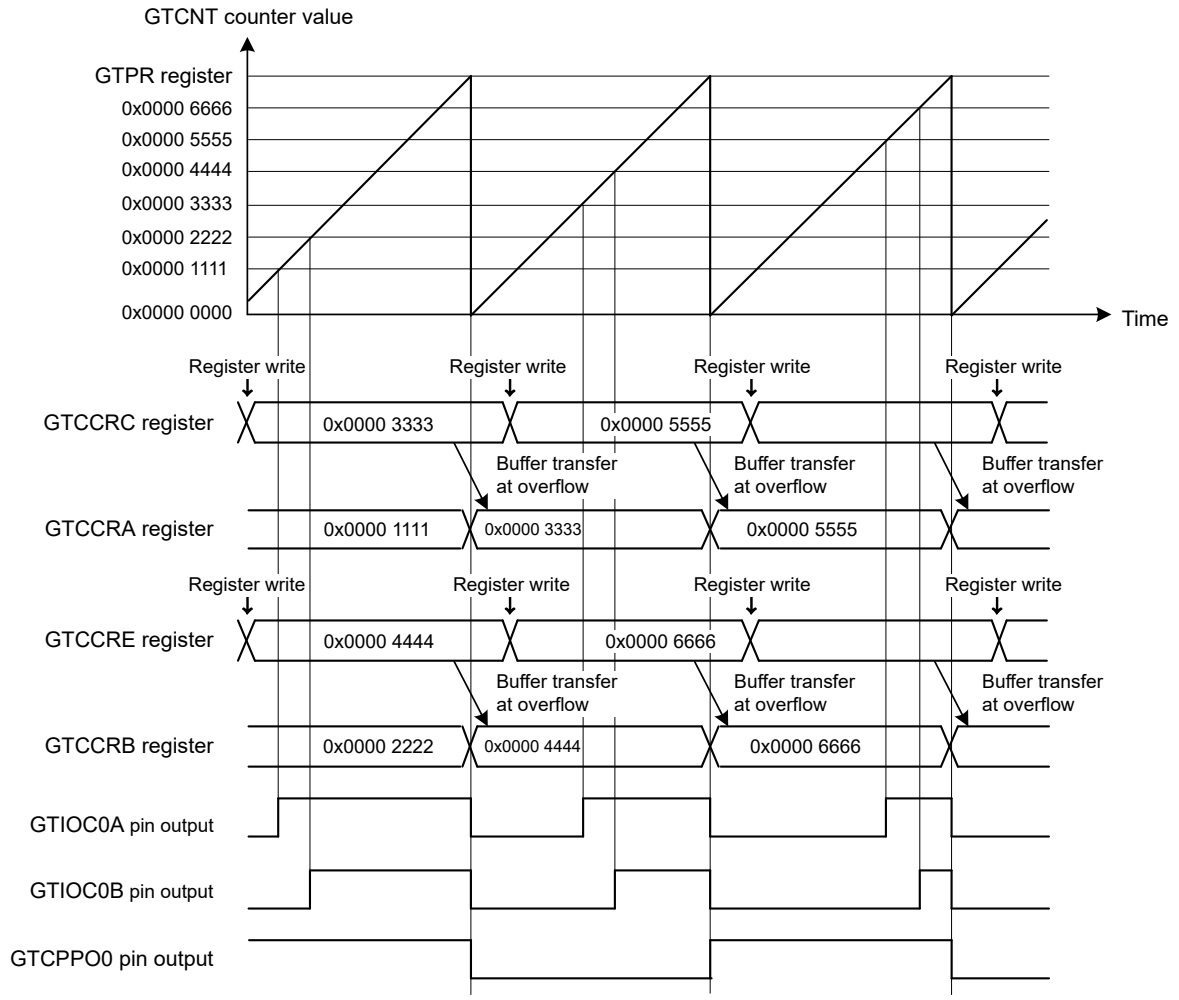
20.3.3.1 Saw-Wave PWM Mode 1

In saw-wave PWM mode 1, GTCNT performs saw-wave (half-wave) operation by setting the cycle in GTPR and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 2) when a GTCCRA or GTCCRB compare match occurs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the cycle end according to the GTIOR setting.

When 0 is set in the GTIOR.OxEOCD (x = A, B) bit and the timing of the end of cycle and the timing of GTCCRx register compare match are the same time, the output pin performs along the PWM output setting for the end of cycle set by the GTIOR.GTIOx[3:2] bits.

When 1 is set in the GTIOR.OxEOCD bit, GTIOCnx output is retained.

Figure 20.38 shows an example of saw-wave PWM mode 1 operation, and Table 20.24 shows an example for setting saw-wave PWM mode 1.



Note: The base address of registers are the same.

Figure 20.38 Example of saw-wave PWM mode 1 operation with up-counting, buffer operation, high output at GTCCRA/GTCCRB compare match, low output at cycle end, and GTIOR.PSYE = 1

Table 20.24 Example for setting saw-wave PWM mode 1 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.38, 000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.38, after 11b is set in the GTUDDTYC.UDF and UD bits, 01b is set in the GTUDDTYC.UDF and UD bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 20.38, GTIOA[4:0] = 00110b and GTIOB[4:0] = 00110b.
7	Enable GTCPPOn pin output	Set to enable/disable the GTCPPOn pin output with the PSYE bit in the GTIOR register.

Table 20.24 Example for setting saw-wave PWM mode 1 (2 of 2)

No.	Step Name	Description
8	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
9	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 20.38, CCRA[1:0] = 01b and CCRB[1:0] = 01b.
10	Set compare match value*1	Set the GTIOCnA pin transition in the GTCCRA register and the GTIOCnB pin transition in the GTCCRB register.
11	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.
12	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
13	Set buffer value for each cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

Note: n: 0 to 3
m: A, B

Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

20.3.3.2 Saw-Wave PWM Mode 2

The saw-wave PWM mode 2 is a mode in which the GTCNT counter is operated as a saw wave by up-counting without using the GTPR register, and the PWM waveform is output by the compare match of the GTCCRA and GTCCRB registers. The pin output level can be selected from low output, high output, or toggle output separately for a compare match according to the GTIOR register setting.

The GTIOCnA pin is used as an output pin. Use the GTIOR.GTIOB[1:0] bits for setting the GTIOCnA pin output at a compare match of the GTCCRB register.

When a counter clear occurs due to the GTCNT counter clearing source selected in the GTCSR register, this is handled at the end of cycle and PWM output operation at the end of the cycle selected with the GTIOR.GTIOA[3:2] bits is performed. If a counter clear (at the end of cycle) conflicts with a PWM output change due to a GTCCRx (x = A, B) register compare match, PWM output operation is performed at the end of cycle (when GTIOR.OxEOCD bit = 0) or the PWM output is retained (when GTIOR.OxEOCD bit = 1).

Figure 20.39 to Figure 20.41 shows an example of saw-wave PWM mode 2 operation. Table 20.25 shows an example for setting saw-wave PWM mode 2.

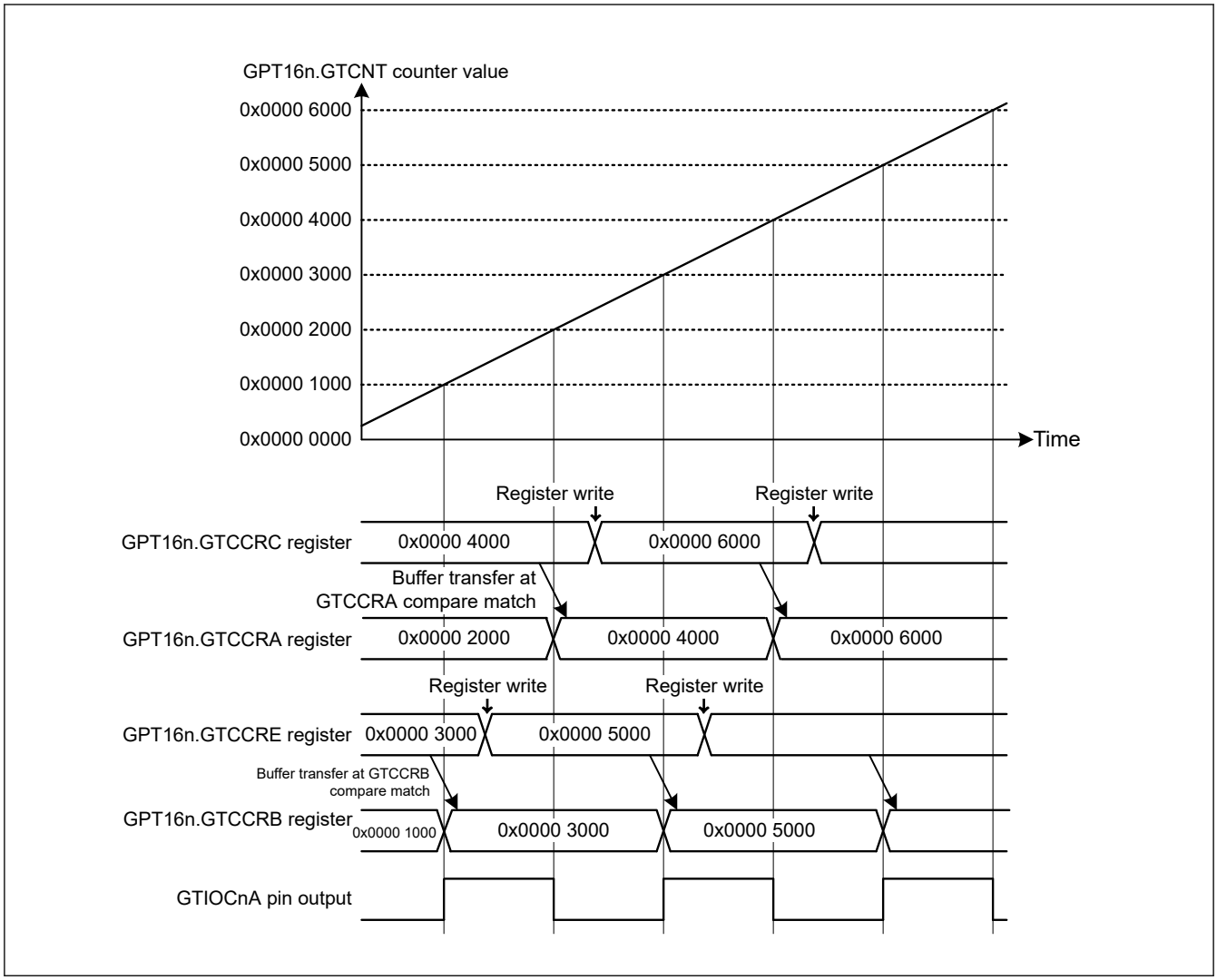


Figure 20.39 Example of saw-wave PWM mode 2 operation (low output at GTCCRA register compare match, high output at GTCCRB register compare match, single buffer operation, no clear setting) (n = 0 to 2)

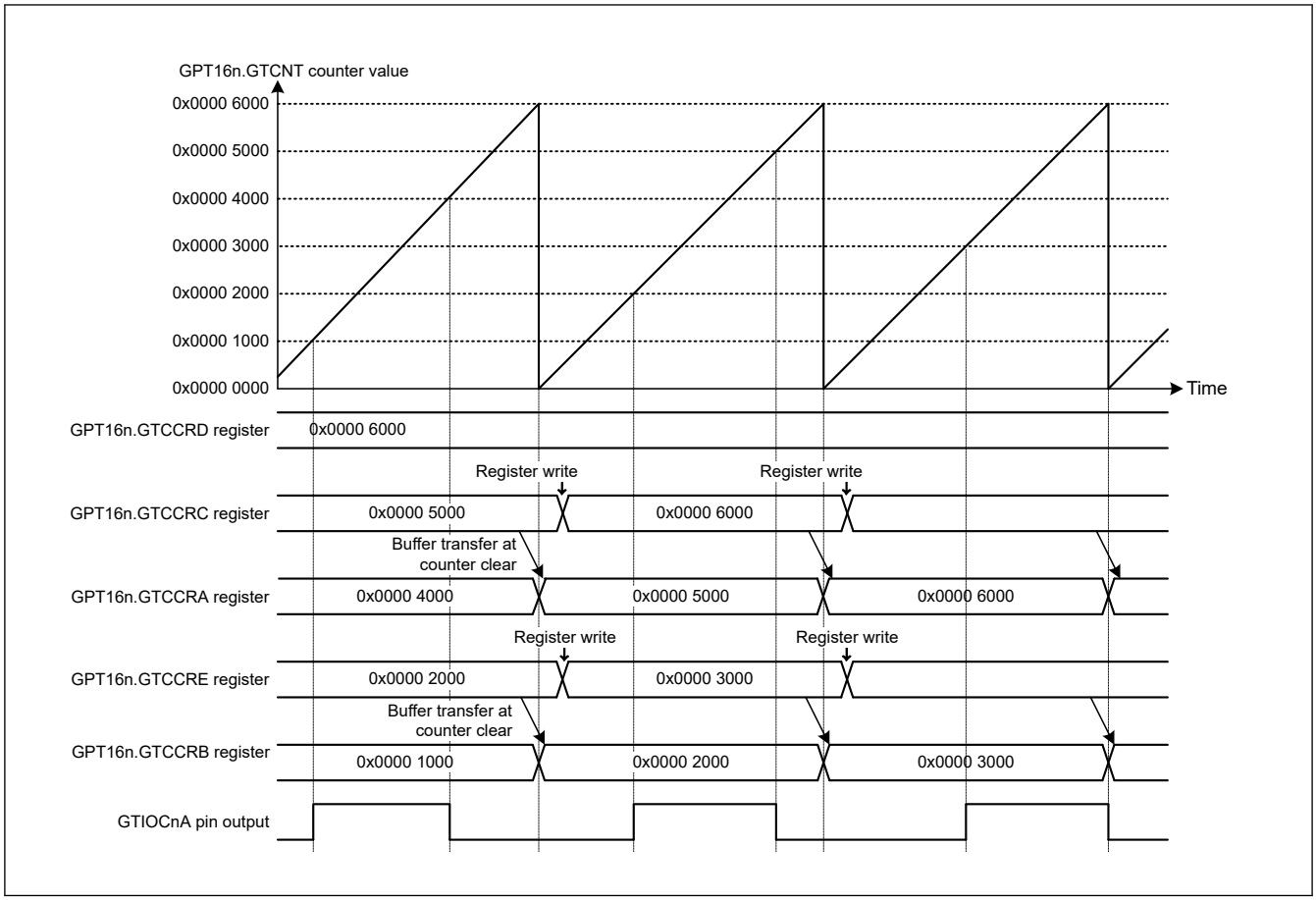


Figure 20.40 Example of saw-wave PWM mode 2 operation (low output at GTCCRA register compare match, high output at GTCCRB register compare match, low output at the end of cycle, single buffer operation, cleared at GTCCRD register compare match, GTIOR.OAEOCD bit = 0) (n = 0 to 2)

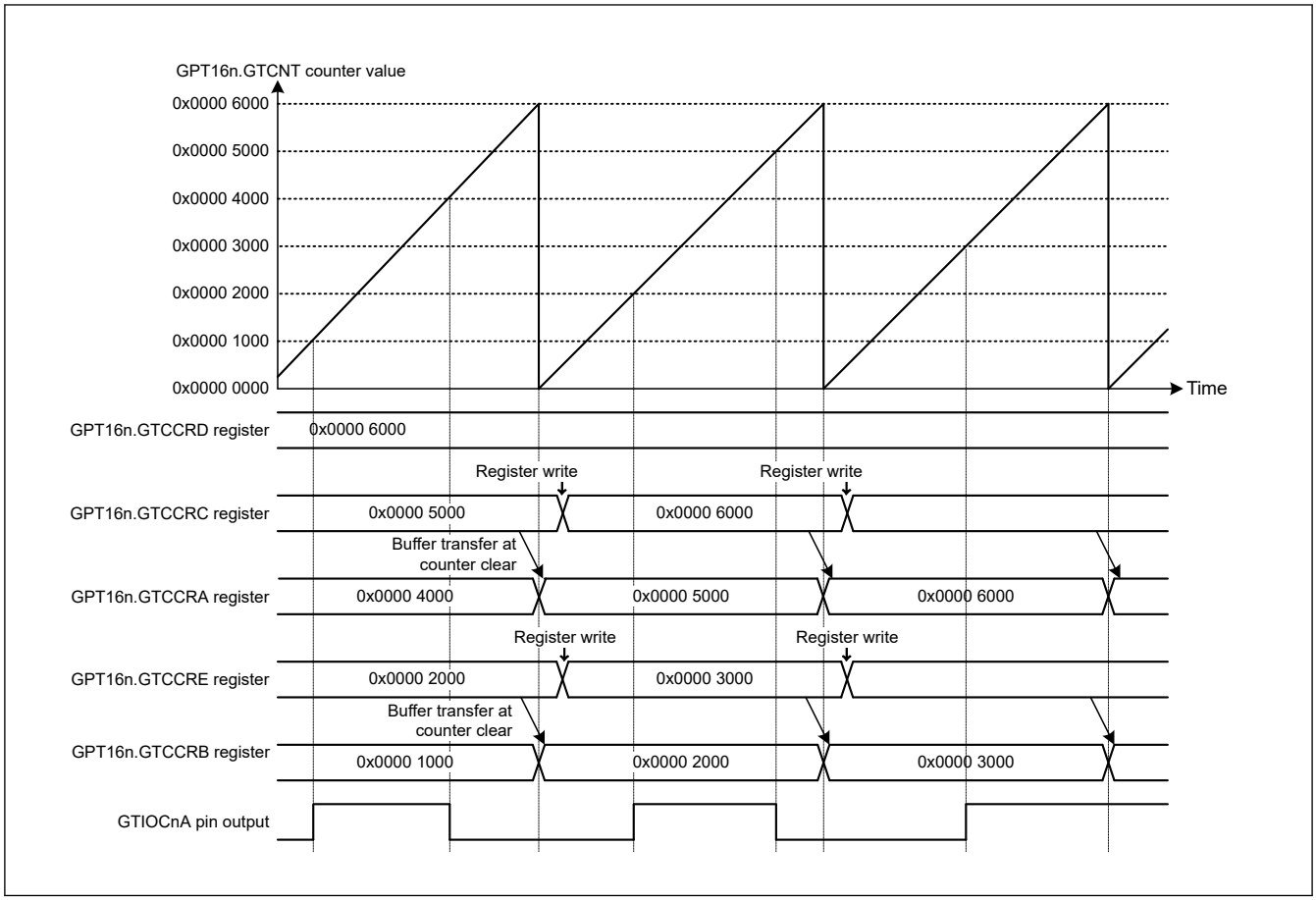


Figure 20.41 Example of saw-wave PWM mode 2 operation (low output at GTCCRA register compare match, high output at GTCCRB register compare match, low output at the end of cycle, single buffer operation, cleared at GTCCRD register compare match, GTIOR.OAEOCD bit = 1) (n = 0 to 2)

Table 20.25 Example for setting saw-wave PWM mode 2

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[3:0]. In Figure 20.39 to Figure 20.41, 0010b (saw-wave PWM mode 2) is set.
2	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits.
3	Set initial value for counter	Set the initial value in the GTCNT counter.
4	Set GTIOCnA pin function	Set the GTIOCnA pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 20.39 to Figure 20.41, GTIOA[4:0] = 00001b and GTIOB[1:0] = 10b.
5	Enable GTIOCnA pin output*1	Set to enable the GTIOCnA pin output with OAE in GTIOR.
6	Set buffer operation	Set buffer operation with CCRA[1:0] and CCRB[1:0] bits in GTBER In Figure 20.39 to Figure 20.41, CCRA[1:0] = 01b and CCRB[1:0] = 01b.
7	Set compare match value*1	Set the GTIOCnA pin changing point in GTCCRA and GTCCRB.
8	Set buffer value	For buffer operation, set the GTCCRA register value and the GTCCRB register value which used in one cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTCCRA register value and the GTCCRB register value which used in two cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.
9	Start count operation	Set GTCR.CST to 1 to start count operation.
10	Set buffer value of each cycle	For buffer operation, set the GTCCRA register value and the GTCCRB register value which used in one cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTCCRA register value and the GTCCRB register value which used in two cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

Note: n: 0 to 2
Note 1. When using PWM delay generation circuit, change the order of setting for the GTIOCnA pin output enable and setting for a compare match value.

20.3.3.3 Saw-Wave One-Shot Pulse Mode

The saw-wave one-shot pulse mode is a mode in which the cycle is set in GTPR, the GTCNT counter performs saw-wave (half-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 3) at a compare match of GTCCRA or GTCCRB with buffer operation fixed.

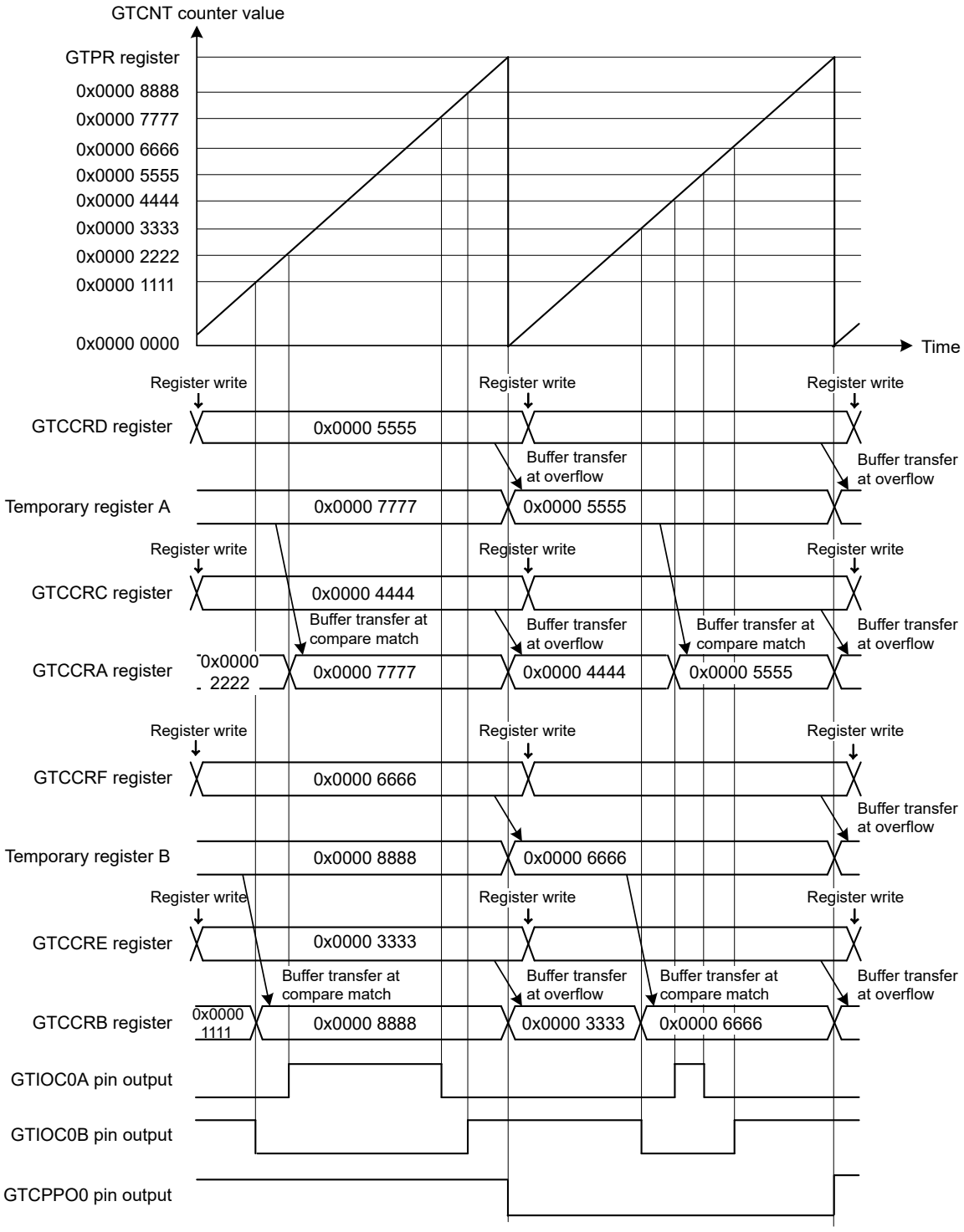
Buffer operation in saw-wave one-shot pulse mode is different from the usual buffer operation. Buffer transfer is performed from:

- GTCCRC to GTCCRA at the cycle end
- GTCCRE to GTCCRB at the cycle end
- GTCCRD to temporary register A at the cycle end
- GTCCRF to temporary register B at the cycle end
- Temporary register A to GTCCRA at a GTCCRA compare match
- Temporary register B to GTCCRB at a GTCCRB compare match.

The pin output value can be selected from low output, high output, or toggled output separately for a compare match and the cycle end according to the GTIOR setting. When the GTBER.CCRSWT bit is set to 1 while count operation is stopped, the buffer is transferred forcibly from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B. By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

When the GTBER.DBRTECm (m = A, B) bit is set to 1, transfer from an intermediate buffer to the GTCCRm register is repeated on a cyclic basis with using the temporary register x (x = C, E) and temporary register m which operate as intermediate buffers for the GTCCRx and GTCCRm registers, respectively, even while buffer transfer is disabled (repeated double buffer operation function during disabling of buffer transfer). For details, see section 20.8.2.2. Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.

Figure 20.42 shows an example of saw-wave one-shot pulse mode operation, and Table 20.26 shows an example for setting saw-wave one-shot pulse mode.



Note: The base address of registers are the same.

Figure 20.42 Example of saw-wave one-shot pulse mode operation with up-counting, low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/ GTCCRB compare match, output retained at cycle end, and GTIOR.PSYE = 1

Table 20.26 Example setting for saw-wave one-shot pulse mode

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.42, 001b (saw-wave one-shot pulse mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.42, after 11b is set in the GTUDDTYC.UDF and UD bits, 01b is set in the GTUDDTYC.UDF and UD bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 20.42, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
7	Enable GTCPPOn pin output	Set to enable/disable the GTCPPOn pin output with the PSYE bit in the GTIOR register.
8	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
9	Set compare match value*1	Set the GTIOCnA pin transition immediately after the count start in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
10	Set forcible buffer transfer*1	Set the GTBER.CCRSWT bit to 1 to transfer buffer register data forcibly.
11	Set buffer value	For buffer operation, set the GTIOCnA pin transition in one cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
12	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
13	Set buffer value for each cycle	For buffer operation, set the GTIOCnA pin transition in one cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.

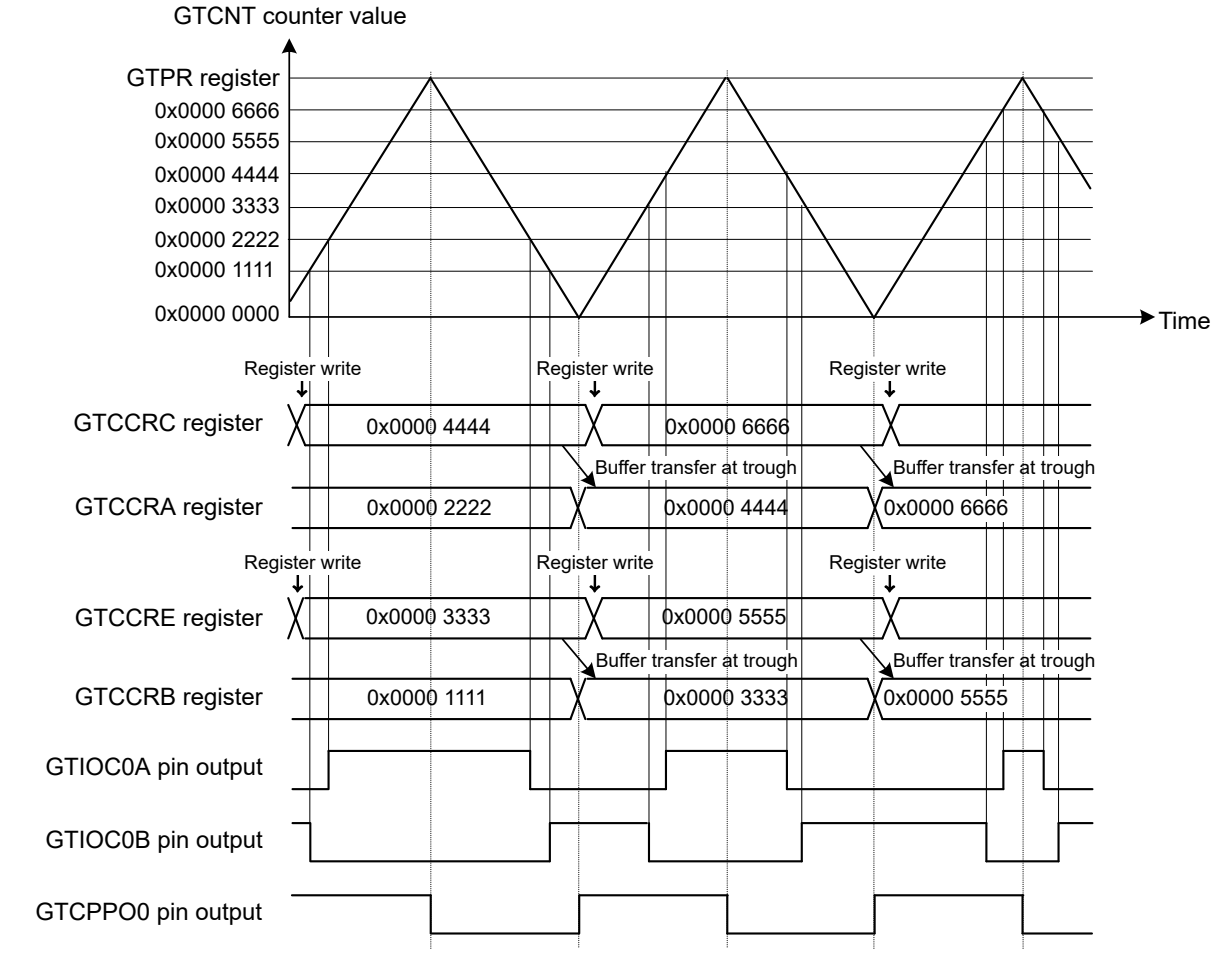
Note: n: 0 to 3
m: A, B
Note 1. When using PWM delay generation circuit, change the order of Enable GTIOCnm pin output and Set compare match value + Set forcible buffer transfer.

20.3.3.4 Triangle-Wave PWM Mode 1 (32-Bit Transfer at Trough)

The triangle-wave PWM mode 1 is a mode in which the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 3) when a GTCCRA or GTCCRB compare match occurs. Buffer transfer is performed at the trough. The pin output value can be selected from low output, high output, or toggled output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 20.43 shows an example of a triangle-wave PWM mode 1 operation, and Table 20.27 shows an example for setting a triangle-wave PWM mode 1.



Note: The base address of registers are the same.

Figure 20.43 Example of triangle-wave PWM mode 1 operation with buffer operation, low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/ GTCCRB register compare match, output retained at cycle end, and GTIOR.PSYE = 1

Table 20.27 Example setting for triangle-wave PWM mode 1 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.43, 100b (triangle-wave PWM mode 1) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 20.43, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTCPPOn pin output	Set to enable/disable the GTCPPOn pin output with the PSYE bit in the GTIOR register.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 20.43, CCRA[1:0] = 01b and CCRB[1:0] = 01b.
9	Set compare match value*1	Set the GTIOCnA and GTIOCnB pins transitions in the GTCCRA and GTCCRB registers, respectively.

Table 20.27 Example setting for triangle-wave PWM mode 1 (2 of 2)

No.	Step Name	Description
10	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

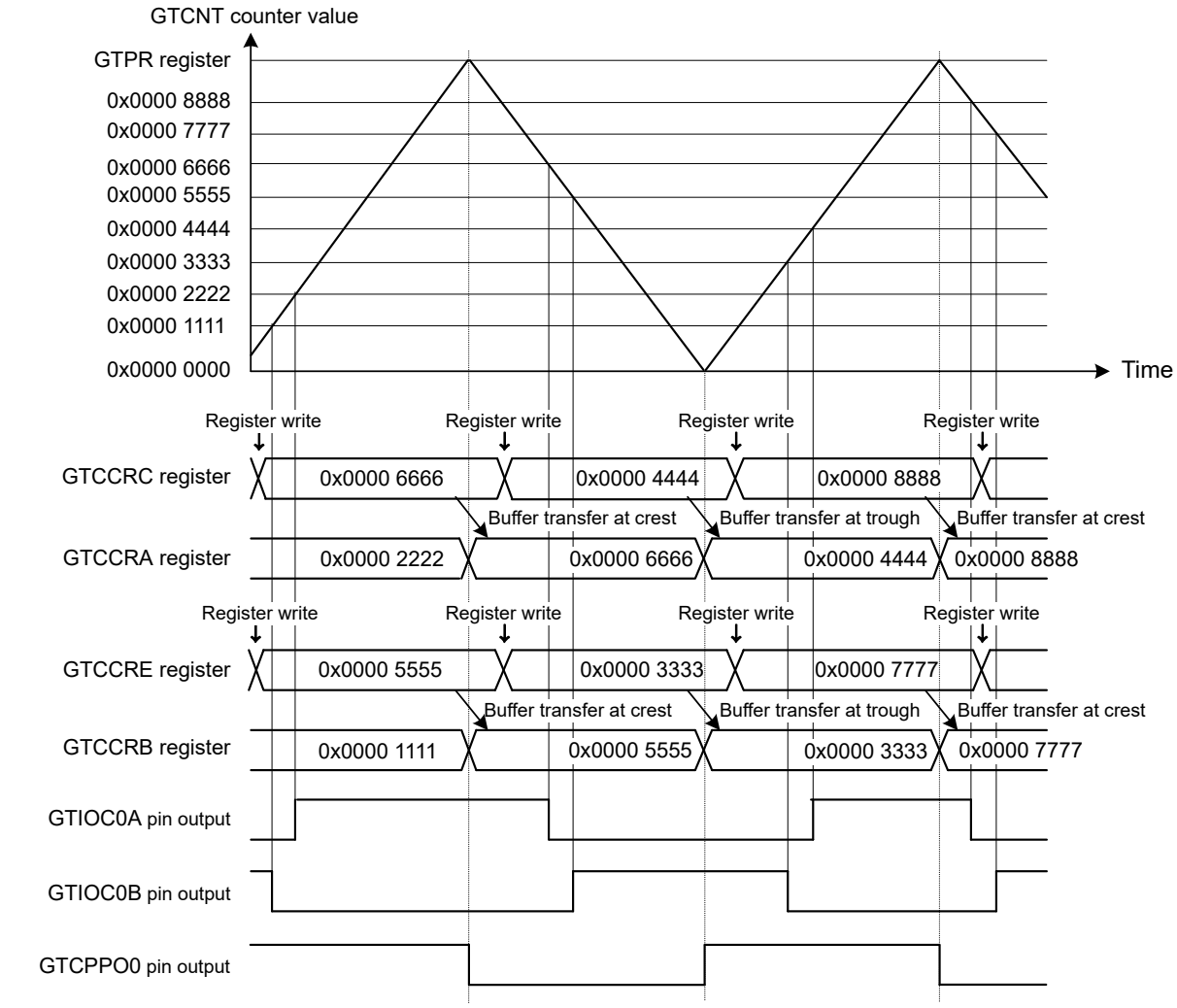
Note: n: 0 to 3
m: A, B
Note 1. When using PWM delay generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

20.3.3.5 Triangle-Wave PWM Mode 2 (32-Bit Transfer at Crest and Trough)

Similarly to triangle-wave PWM mode 1, in triangle-wave PWM mode 2 the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 3) when a GTCCRA or GTCCRB compare match occurs. The buffer transfer is performed at both crests and troughs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 20.44 shows an example of triangle-wave PWM mode 2 operation, and Table 20.28 shows an example for setting triangle-wave PWM mode 2.



Note: The base address of registers are the same.

Figure 20.44 Example of triangle-wave PWM mode 2 operation with buffer operation, low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/ GTCCRB compare match, output retained at cycle end, and GTIOR.PSYE = 1

Table 20.28 Example for setting triangle-wave PWM mode 2 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.44, 101b (triangle-wave PWM mode 2) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 20.44, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTCPPOn pin output	Set to enable/disable the GTCPPOn pin output with the PSYE bit in the GTIOR register.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 20.44, CCRA[1:0] = 01b and CCRB[1:0] = 01b.

Table 20.28 Example for setting triangle-wave PWM mode 2 (2 of 2)

No.	Step Name	Description
9	Set compare match value*1	Set the GTIOCnA and GTIOCnB pins transitions in the GTCCRA and GTCCRB registers, respectively.
10	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in half cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRD and GTCCRF registers, respectively.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each half cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in half cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in GTCCRD and GTCCRF registers, respectively.

Note: n: 0 to 3
m: A, B
Note 1. When using PWM delay generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

20.3.3.6 Triangle-Wave PWM Mode 3 (64-Bit Transfer at Trough)

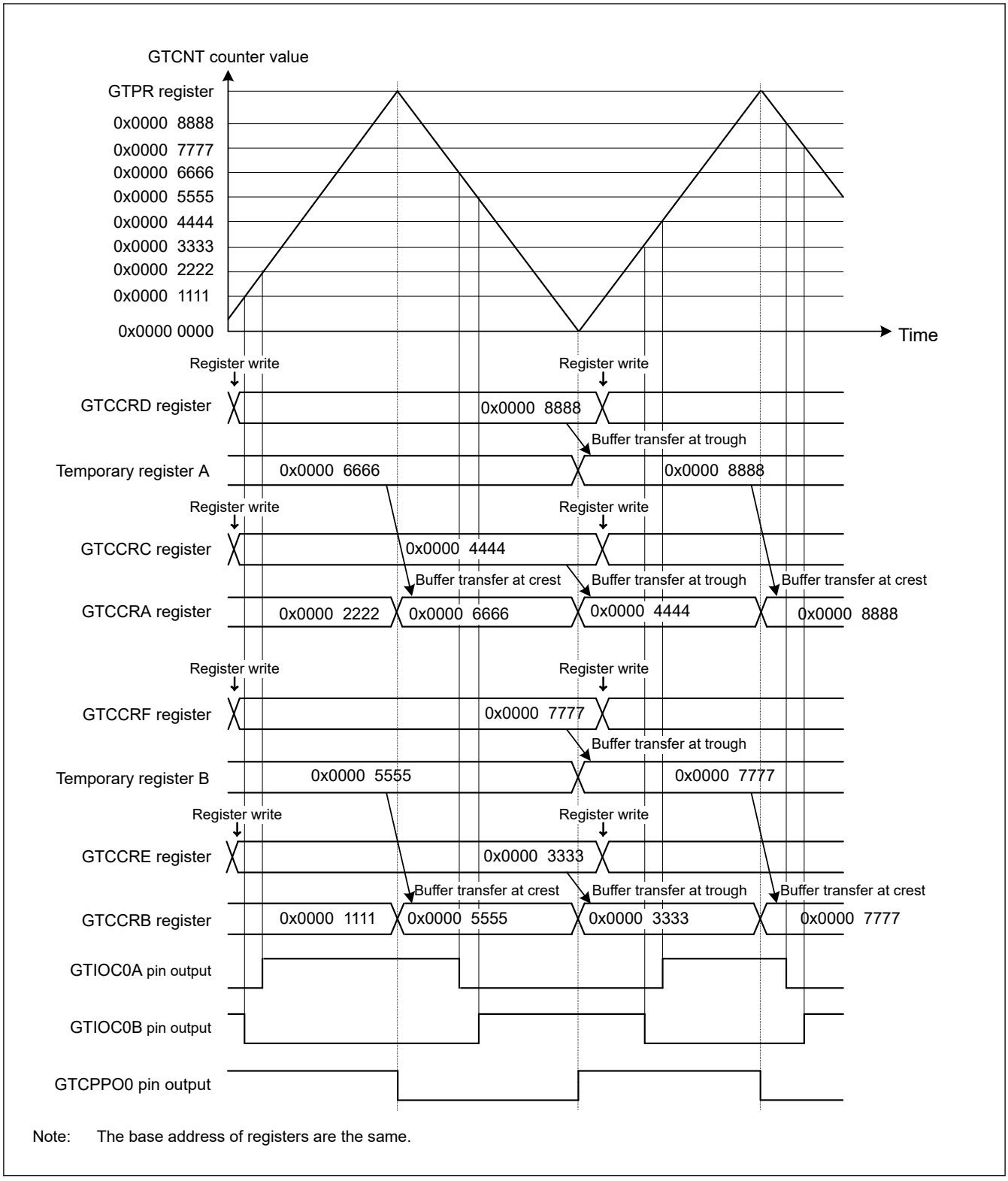
The triangle-wave PWM mode 3 is a mode in which the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 3) at a compare match of GTCCRA or GTCCRB with buffer operation fixed. Buffer operation in triangle-wave PWM mode 3 is different from the usual buffer operation. Buffer transfer is performed from the following:

- GTCCRC to GTCCRA at the trough
- GTCCRE to GTCCRB at the trough
- GTCCRD to temporary register A at the trough
- GTCCRF to temporary register B at the trough
- Temporary register A to GTCCRA at the crest
- Temporary register B to GTCCRB at the crest.

The pin output value can be selected from low output, high output, or toggled output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR, GTDVU, and GTDVD, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 20.45 shows an example of triangle-wave PWM mode 3 operation, and Table 20.29 shows an example for setting triangle-wave PWM mode 3.



Note: The base address of registers are the same.

Figure 20.45 Example of triangle-wave PWM mode 3 operation with low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/GTCCRB compare match, output retained at cycle end, and GTIOR.PSYE = 1

Table 20.29 Example setting for triangle-wave PWM mode 3 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.45, 110b (triangle-wave PWM mode 3) is set.

Table 20.29 Example setting for triangle-wave PWM mode 3 (2 of 2)

No.	Step Name	Description
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 20.45, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTCPPOn pin output	Set to enable/disable the GTCPPOn pin output with the PSYE bit in the GTIOR register.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set compare match value*1	Set the GTIOCnA pin transition immediately after the count start in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
9	Set forcible buffer transfer*1	Set the GTBER.CCRSWT bit to 1 to transfer buffer register data forcibly.
10	Set buffer value	Set the GTIOCnA pin transition in 1 cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each cycle	Set the GTIOCnA pin transition in 1 cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.

Note: n: 0 to 3
m: A, B
Note 1. When using PWM delay generation circuit, change the order of Enable GTIOCnm pin output and Set compare match value + Set forcible buffer transfer.

20.3.3.7 Complementary PWM Modes 1, 2, 3

In complementary PWM mode, a three-phase PWM waveform with dead time that ensures the linearity in the vicinity of duty 0% and 100% can be output using the GTCNT counter of consecutive three channels. There are four modes depending on the differences in buffer operation:

- Complementary PWM mode 1 (transfer at crests)
- Complementary PWM mode 2 (transfer at troughs)
- Complementary PWM mode 3 (transfer at crests and troughs)
- Complementary PWM mode 4 (immediate transfer).

Figure 20.46 shows the block diagram in complementary PWM modes 1 to 3.

In the three consecutive channels, the lowest channel is referred to as master channel, and the adjacent upper two channels are referred to as slave channel 1 (lower) and slave channel 2 (upper).

The GTCNT counter of each channel performs individual count operation under the cycle operation by the master channel. In each channel, compare match with the GTCCRA register is performed selecting one of the three GTCNT counters in each operation section, and a positive-phase waveform and a negative-phase waveform are output from the GTIOCn+iA pin (i = 0, 1, 2) and the GTIOCn+iB pin respectively with a non-overlapping section of the dead time value set in the GTDVU register of the master channel.

The GTCCRA register performs buffer operation by the GTCCRC register, temporary register A, and GTCCRD register. In complementary PMW mode 3, setting the GTBER2.CP3DB bit to 1 also enables buffer operation of the GTCCRA register by the GTCCRE register, temporary register B, and GTCCRF register, allowing double buffer operation.

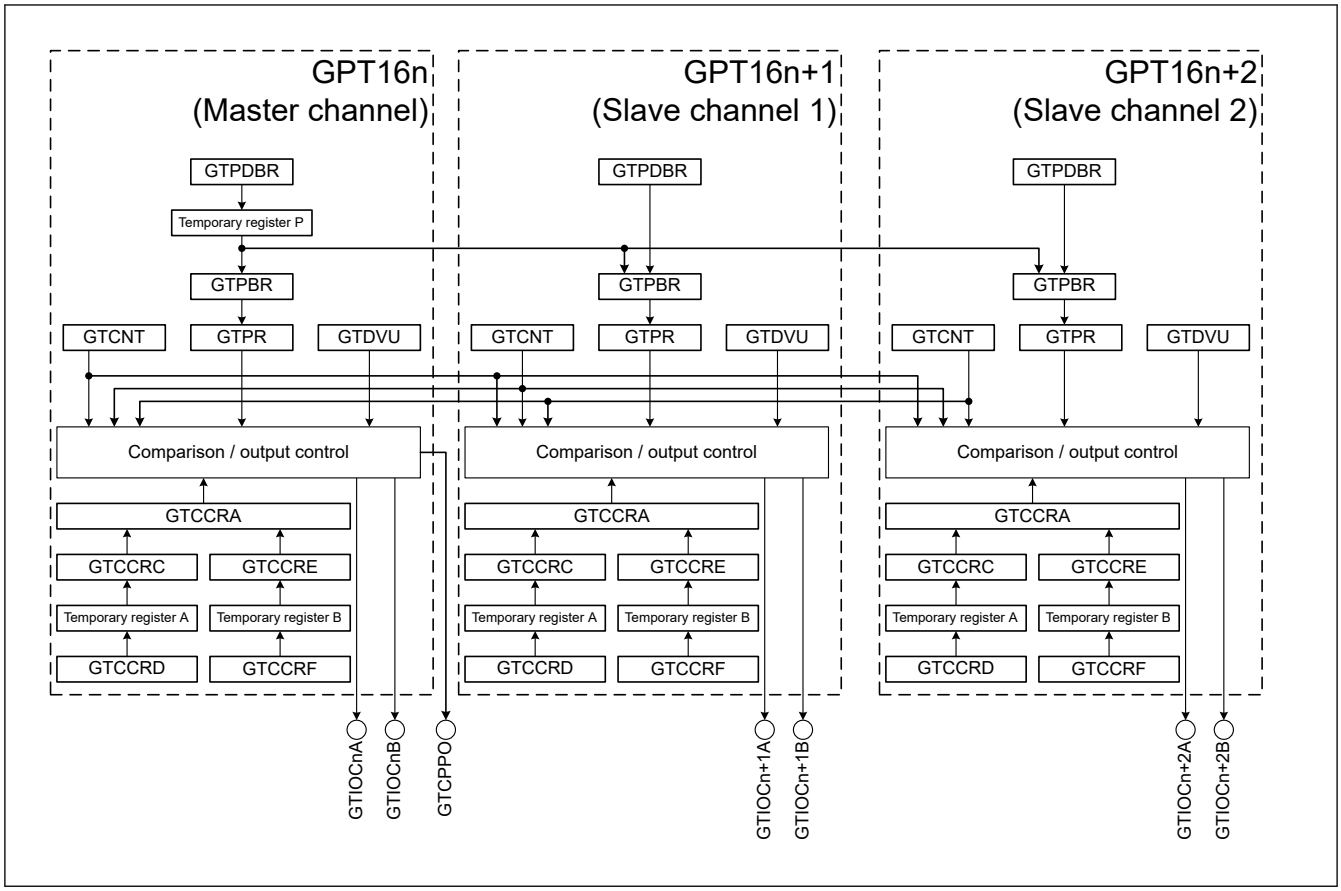


Figure 20.46 Block diagram in complementary PWM mode 1, 2, 3 (n = 0)

The GPT16n.GTCNT counter performs count operation for triangle waves using the GPT16n.GTPR register as a cycle register. A section where the GPT16n.GTCNT counter value is not larger than the dead time value is referred to as trough section.

The GPT16n+1.GTCNT counter performs count operation with the value (GPT16n.GTCNT counter value + dead time value set in the GPT16n.GTDVU register). A section where the GPT16n+1.GTCNT counter value is larger than the GPT16n.GTPR register value is referred to as crest section.

Crest section and trough section are classified into up-counting crest section, down-counting crest section, up-counting trough section, and down-counting trough section according to counting direction. A section between trough section and crest section is referred to as up-counting middle section or down-counting middle section according to counting direction. A section equivalent to the up-counting trough section after starting count operation is referred to as initial output section where operation differs partially from other up-counting trough sections.

The GPT16n+2.GTCNT counter functions as a counter to ensure linearity in the vicinity of duty 0% and 100%. In a crest section, this counter performs count operation for a triangle wave (up-counting after down-counting) with the value (GPT16n.GTPR register value + dead time value) as an initial value and the GPT16n.GTPR register value as a trough. This counter is cleared to 0 at the end of the crest section, and then stops counting until the next trough section. In a trough section, this counter performs count operation for a triangle wave with an initial value of 0 and dead time value as a crest. This counter becomes the value (GPT16n.GTPR register value + dead time value) at the end of the trough section, and then stops counting until the next crest section. In the initial output section, however, this counter counts up with an initial value of 0 until the dead time value, and then becomes the value (GPT16n.GTPR register value + dead time value).

If the counter stops and then restarts in complementary PWM mode, the counter of each channel returns to the initial value after starting count operation, and then starts counting from the initial output section.

Table 20.30 and Table 20.31 shows count operation (counting direction/counting range) in each section. In these tables, registers with no channel identification indicate that the same value is stored in them of the master channel, slave channel 1, and slave channel 2.

Table 20.30 Count operation in complementary PWM mode

Counter	Initial value	Initial Output section (after Start)	Up-Counting Middle section	Up-Counting Crest section	Down-Counting Crest section
GPT16n.GTCNT	0	Up-counting 0 to GTDVU	Up-counting GTDVU+1 to GTPR-GTDVU	Up-counting GTPR-GTDVU+1 to GTPR	Down-counting GTPR-1 to GTPR-GTDVU
GPT16n+1.GTCNT	GTDVU	Up-counting GTDVU to GTDVU×2	Up-counting GTDVU×2+1 to GTPR	Up-counting GTPR+1 to GTPR+GTDVU	Down-counting GTPR+GTDVU-1 to GTPR
GPT16n+2.GTCNT	0	Up-counting 0 to GTDVU	Stop GTPR+GTDVU	Down-counting GTPR+GTDVU-1 to GTPR	Up-counting GTPR+1 to GTPR+GTDVU

Table 20.31 Count operation in complementary PWM mode

Counter	Down-Counting Middle section	Down-Counting Trough section	Up-Counting Trough section
GPT16n.GTCNT	Down-counting GTPR-GTDVU-1 to GTDVU	Down-counting GTDVU-1 to 0	Up-counting 1 to GTDVU
GPT16n+1.GTCNT	Down-counting GTPR-1 to GTDVU×2	Down-counting GTDVU×2-1 to GTDVU	Up-counting GTDVU+1 to GTDVU×2
GPT16n+2.GTCNT	Stop 0	Up-counting 1 to GTDVU	Down-counting GTDVU-1 to 0

In complementary PWM mode, the GTCCRA register performs unusual buffer operation.

Data transfers from the GTCCRD register to the temporary register A and from the GTCCRF register to the temporary register B are performed at the same time in three channels by writing a value to the GTCCRD register of the GPT16n+2 channel.

Data transfers from the temporary register A and temporary register B to the GTCCRC and GTCCRE registers vary depending on the transfer timing to the temporary register A and temporary register B. Data transfers from the GTCCRC and GTCCRE registers to the GTCCRA register are performed according to each complementary PWM mode name (crest transfer, trough transfer, and crest/trough transfer).

Buffer operation of the GTPR register in complementary PWM mode is described in section 20.3.2.1. GTPR Register Buffer Operation. Do not perform buffer operation for the GTDVU register in complementary PWM mode.

Table 20.32 shows buffer transfer timing during single buffer operation in complementary PWM modes 1 to 3. Table 20.33 shows buffer transfer timing during double buffer operation in complementary PWM mode 3.

Table 20.32 Single buffer transfer timing in complementary PWM modes 1, 2, 3

Buffer transfer	Complementary PWM mode 1	Complementary PWM mode 2	Complementary PWM mode 3 (single buffer)
GTCCRD ↓ Temporary register A	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT16n+2)	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT16n+2)	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT16n+2)
Temporary register A ↓ GTCCRC	(1) When data is transferred to temporary register A in up-counting middle section After one GTCLK cycle from transfer to temporary register A (2) When data is transferred to temporary register A in a section other than up-counting middle section At the end of trough section	(1) When data is transferred to temporary register A in down-counting middle section After one GTCLK cycle from transfer to temporary register A (2) When data is transferred to temporary register A in a section other than down-counting middle section At the end of crest section	(1) When data is transferred to temporary register A in middle section After one GTCLK cycle from transfer to temporary register A (2) When data is transferred to temporary register A in a section other than middle section At the end of crest and trough sections
GTCCRC ↓ GTCCRA	At the end of crest section Counter clear in up-counting middle section and crest section	At the end of trough section (excluding initial output section) Counter clear in down-counting middle section and trough section	At the end of crest section At the end of trough section (excluding initial output section) Counter clear

Table 20.33 Double buffer transfer timing in complementary PWM mode 3

Transfer from GTCCRD to GTCCRA		Transfer from GTCCRF to GTCCRA	
Buffer transfer	**Transfer timing**	**Buffer transfer**	**Transfer timing**
GTCCRD ↓ Temporary register A	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT16n+2)	GTCCRF ↓ Temporary register B	After one GTCLK cycle from GTCCRD register write of slave channel 2 (GPT16n+2)
Temporary register A ↓ GTCCRC	(1) When data is transferred to temporary register A in middle section After one GTCLK cycle from transfer to temporary register A (2) When data is transferred to temporary register A in a section other than middle section At the end of crest and trough sections	Temporary register B ↓ GTCCRE	(1) When data is transferred to temporary register B in middle section After one GTCLK cycle from transfer to temporary register B (2) When data is transferred to temporary register B in a section other than middle section At the end of crest and trough sections
GTCCRC ↓ GTCCRA	At the end of crest section Counter clear	GTCCRE ↓ GTCCRA	At the end of trough section (excluding initial output section)

An output level change in the positive-phase waveform from the GTIOCn+iA pin (i = 0, 1, 2) and the negative-phase waveform from the GTIOCn+iB pin occurs at a compare match in combination of counters and registers determined for each operation section. In middle sections, the positive-phase waveform output level changes at a compare match of the GPT16n.GTCNT counter and the GTCCRA register, and the negative-phase waveform output level changes at a compare match between the GPT16n+1.GTCNT counter and the GTCCRA register. In crest and trough sections, compare match operation is performed using the GPT16n+2.GTCNT counter, GTCCRC register, and GTCCRE register to ensure the linearity in the vicinity of duty 0% and 100%.

When the compare match value equals to or is larger than the GPT16n.GTPR register value, the duty becomes 0% (positive-phase waveform OFF, negative-phase waveform ON). When the compare match value is 0, the duty becomes 100% (positive-phase waveform ON, negative-phase waveform OFF).

Table 20.34 lists combinations of counters and registers used for compare match operation to generate a positive-phase waveform and a negative-phase waveform in each operation section.

Table 20.34 Combinations of counters and registers for compare match operation in complementary PWM mode

	Up-Counting Middle section	**Up-Counting Crest section**	**Down-Counting Crest section**	**Down-Counting Middle section**	**Down-Counting Trough section**	**Up-Counting Trough section**
Negative-phase OFF	GPT16n+1.GTCNT	GPT16n+1.GTCNT	—	GPT16n+2.GTCNT *1	GPT16n+2.GTCNT	GPT16n+1.GTCNT
	GTCCRA	GTCCRA	—	GTCCRC (GTCCRE for double buffer operation)	GTCCRC (GTCCRE for double buffer operation)	GTCCRC (GTCCRE for double buffer operation)
Positive-phase ON	GPT16n.GTCNT	GPT16n.GTCNT	GPT16n+2.GTCNT	—	GPT16n.GTCNT*1	GPT16n.GTCNT
	GTCCRA	GTCCRA	GTCCRA	—	GTCCRC (GTCCRE for double buffer operation)	GTCCRC (GTCCRE for double buffer operation)
Positive-phase OFF	GPT16n+2.GTCNT *1	GPT16n+2.GTCNT	GPT16n.GTCNT	GPT16n.GTCNT	GPT16n.GTCNT	—
	GTCCRC	GTCCRC	GTCCRC	GTCCRA	GTCCRA	—
Negative-phase ON	—	GPT16n+1.GTCNT *1	GPT16n+1.GTCNT	GPT16n+1.GTCNT	GPT16n+1.GTCNT	GPT16n+2.GTCNT
	—	GTCCRC	GTCCRC	GTCCRA	GTCCRA	GTCCRA

Note 1. Compare match is performed only at the time of final count in the target section, but is not performed at count values other than the final count.

For normal complementary PWM mode waveform, a PWM waveform change occurs in the order of negative-phase OFF → positive-phase ON → positive-phase OFF → negative-phase ON. However, this order may vary depending on operation section and register values. In this case, OFF takes precedence in trough sections and ON takes precedence in crest sections (for negative-phase waveforms), and ON takes precedence in trough sections and OFF takes precedence in crest sections (for positive-phase waveforms). A lower-priority compare match that occurs at the same time or after a higher-priority compare match is ignored.

In the initial output section, the initial output set in the GTIOR register is retained. When the GTCCRA register value is larger than the GTDVU register value at the end of the initial output section, negative phase is enabled. In the case that the GTCCRA register value is not larger than the GTDVU register value, positive phase is enabled.

As operation examples of normal complementary PWM mode waveform where compare match operation occurs in the middle section, Figure 20.47 and Figure 20.48 show complementary PWM mode 1, Figure 20.49 and Figure 20.50 show complementary PWM mode 2, Figure 20.51 and Figure 20.52 show single buffer complementary PWM mode 3, and Figure 20.53 and Figure 20.54 show double buffer complementary PWM mode 3.

Figure 20.55 to Figure 20.66 show complementary PWM mode waveforms where compare match operation occurs in crest sections and trough sections and differences due to compare match occurrence order.

Figure 20.67 and Figure 20.68 show examples of initial output operation according to the GTCCRA register value.

Table 20.35 shows an example for setting complementary PWM modes 1 to 3.

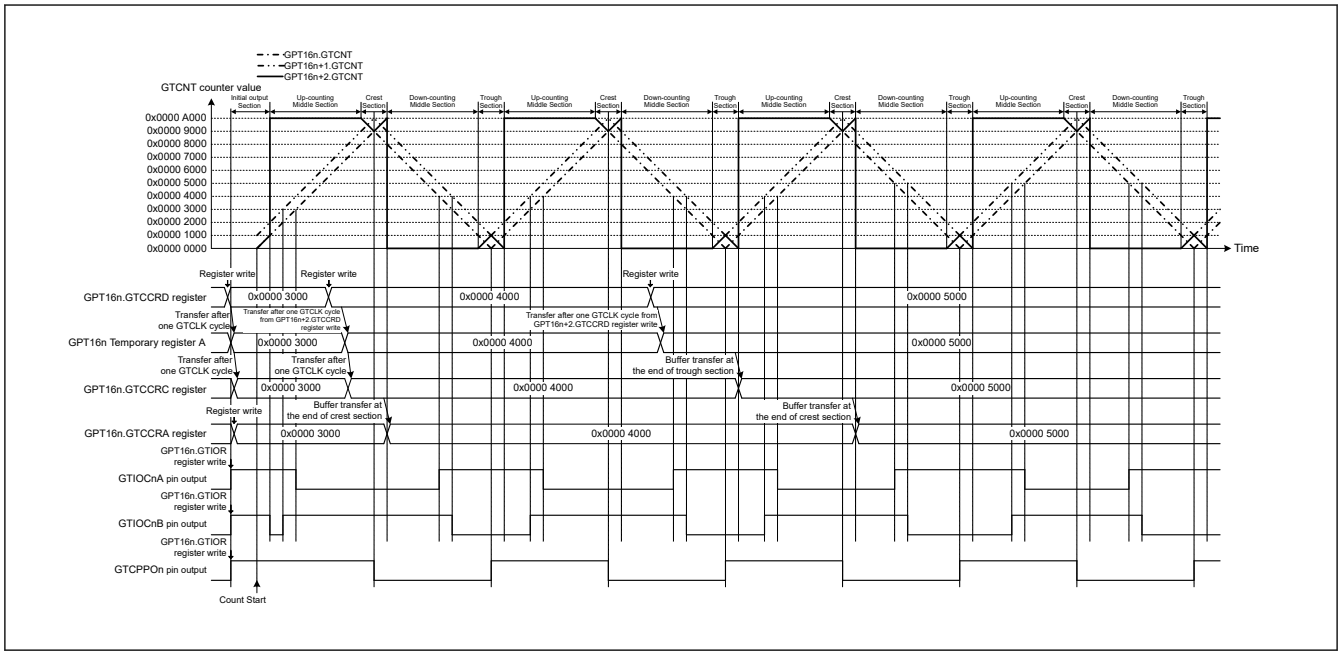


Figure 20.47 Example of complementary PWM mode 1 operation (GTIOCnA pin = high/GTIOCnB pin = high as initial output, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in middle section) (n = 0)

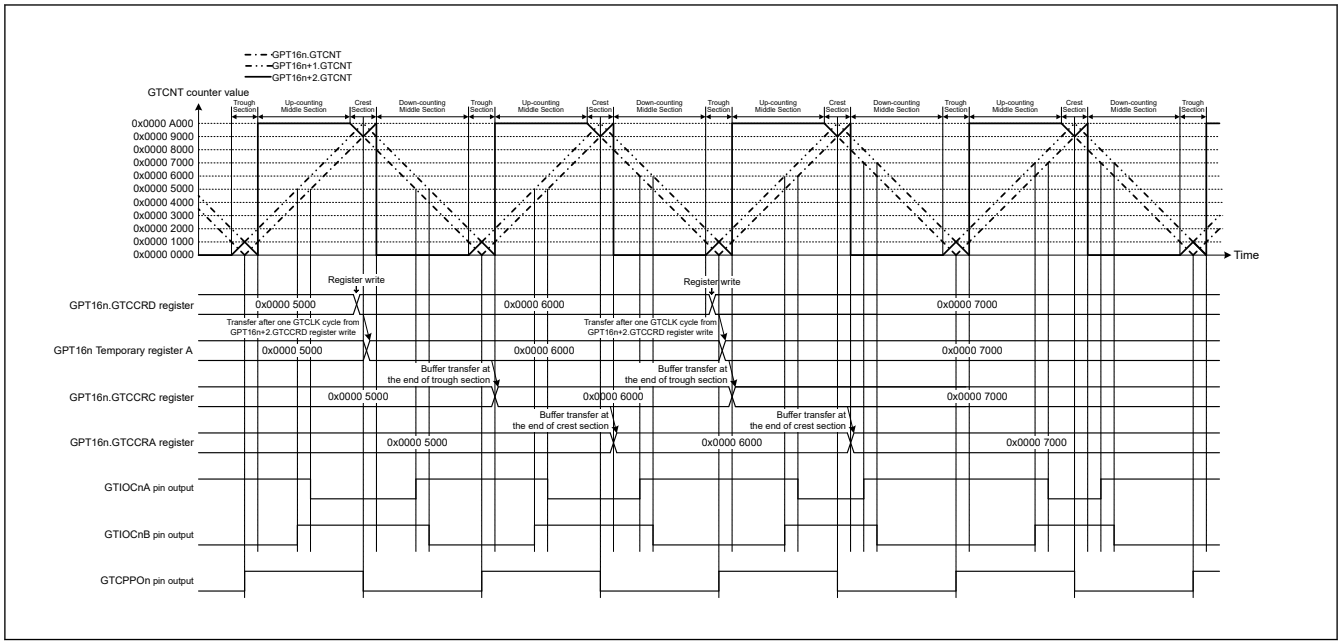


Figure 20.48 Example of complementary PWM mode 1 operation (GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in crest and trough sections) (n = 0)

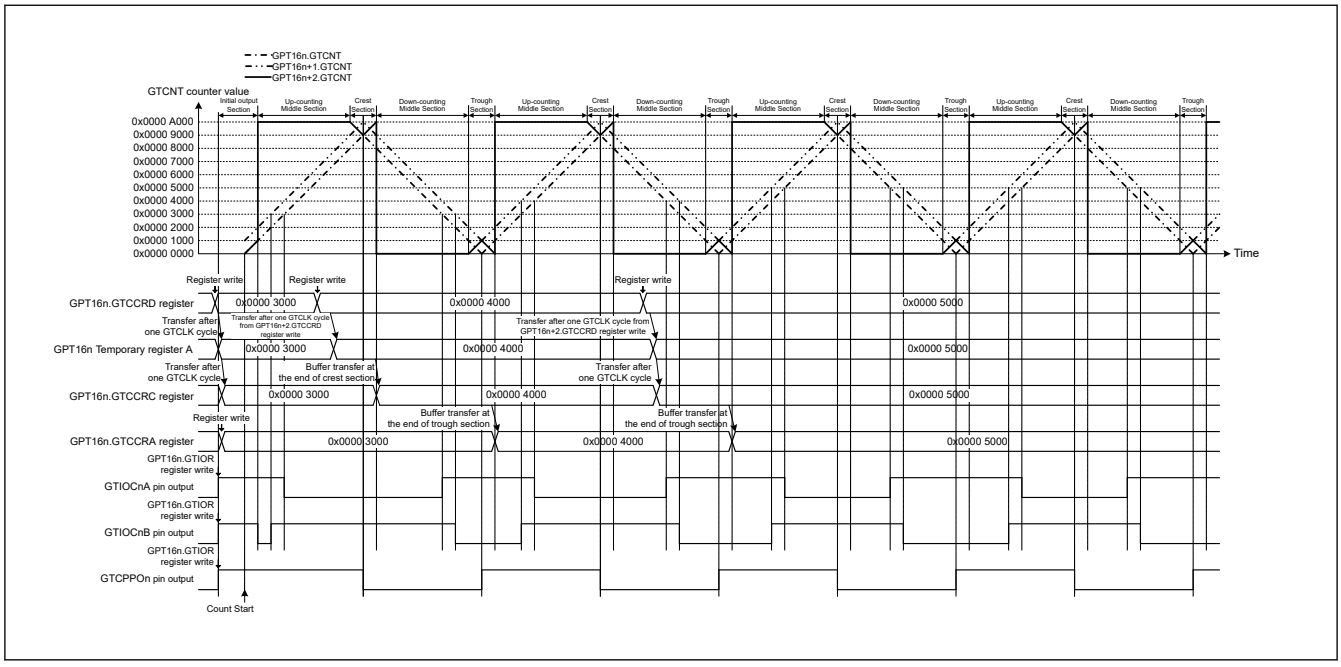


Figure 20.49 Example of complementary PWM mode 2 operation (GTIOCnA pin = high/GTIOCnB pin = high as initial output, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in middle section) (n = 0)

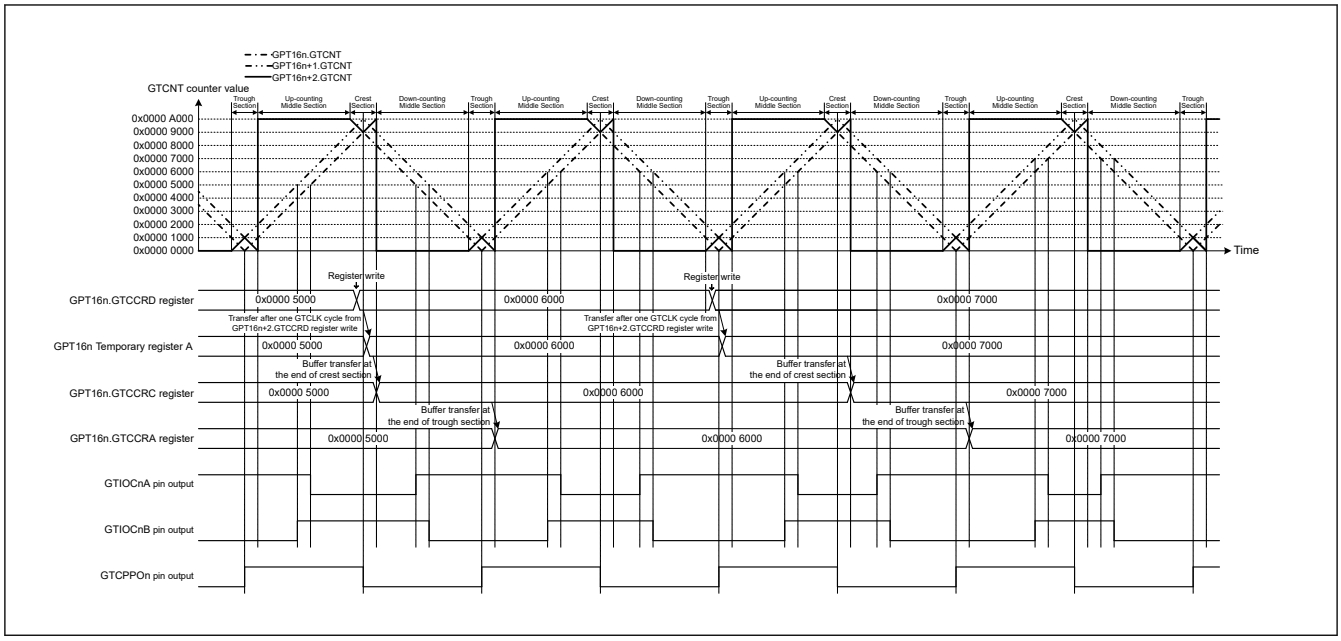


Figure 20.50 Example of complementary PWM mode 2 operation (GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in crest and trough sections) (n = 0)

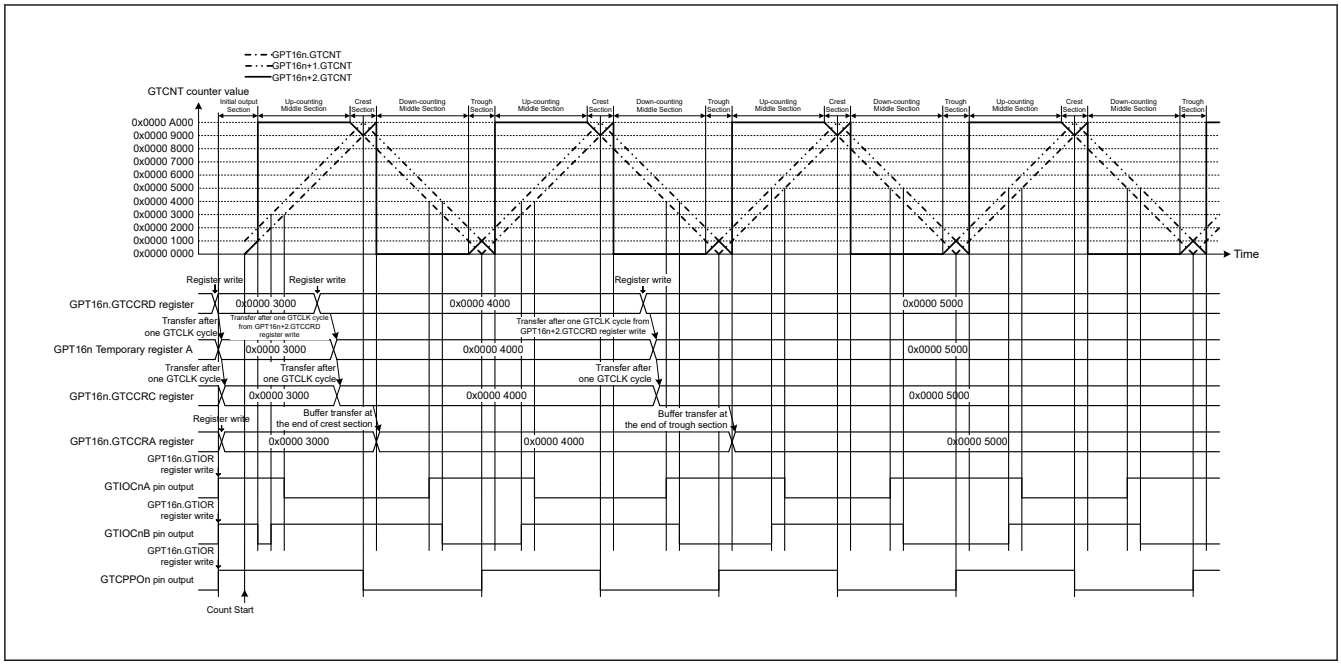


Figure 20.51 Example of complementary PWM mode 3 operation (single buffer operation, GTIOCnA pin = high/GTIOCnB pin = high as initial output, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in middle section) (n = 0)

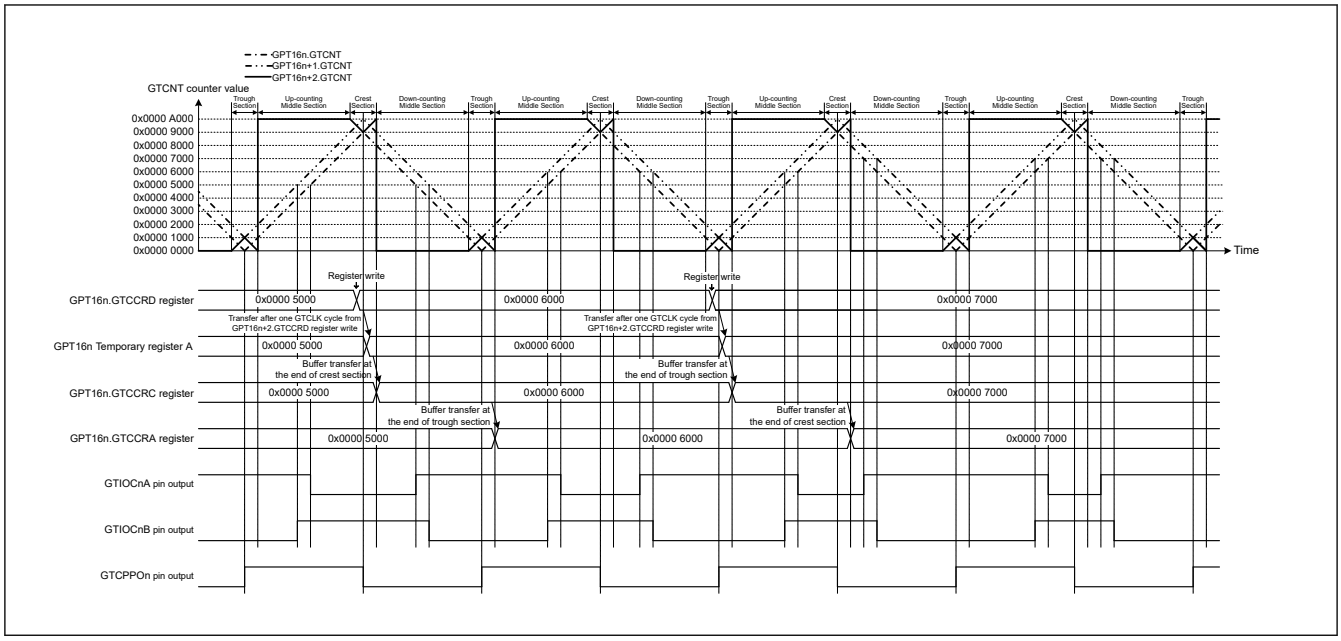


Figure 20.52 Example of complementary PWM mode 3 operation (single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in crest and trough sections) (n = 0)

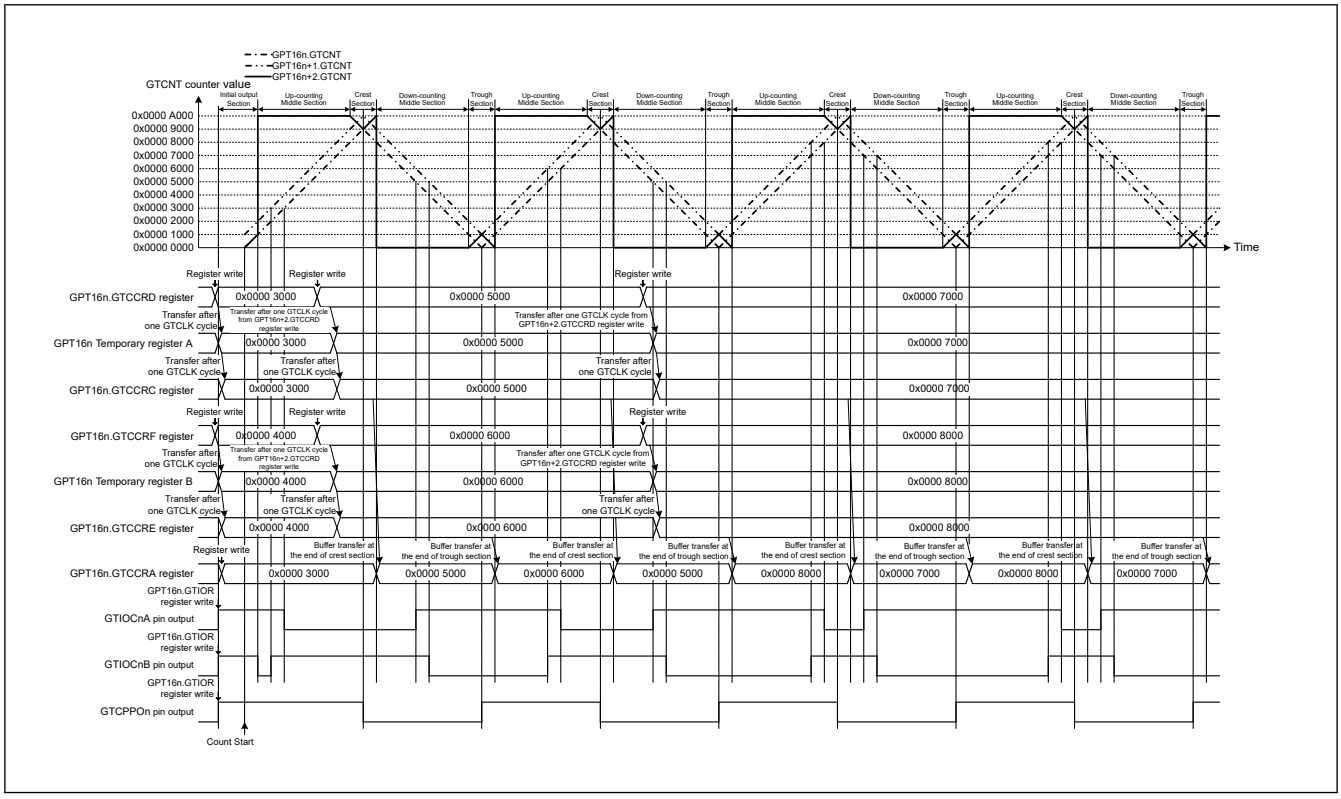


Figure 20.53 Example of complementary PWM mode 3 operation (double buffer operation, GTIOCnA pin = high/GTIOCnB pin = high as initial output, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in middle section) (n = 0)

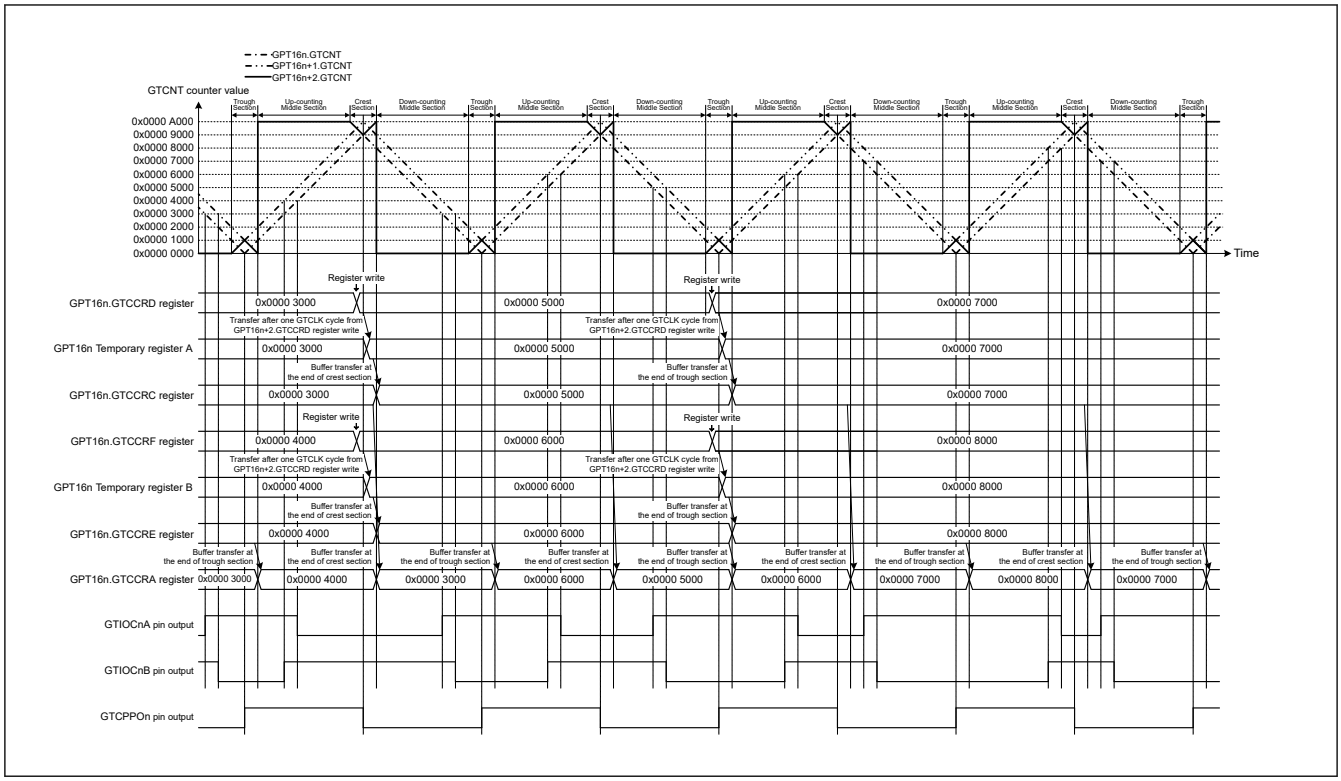


Figure 20.54 Example of complementary PWM mode 3 operation (double buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000 and updating GTCCRD register in crest and trough sections) (n = 0)

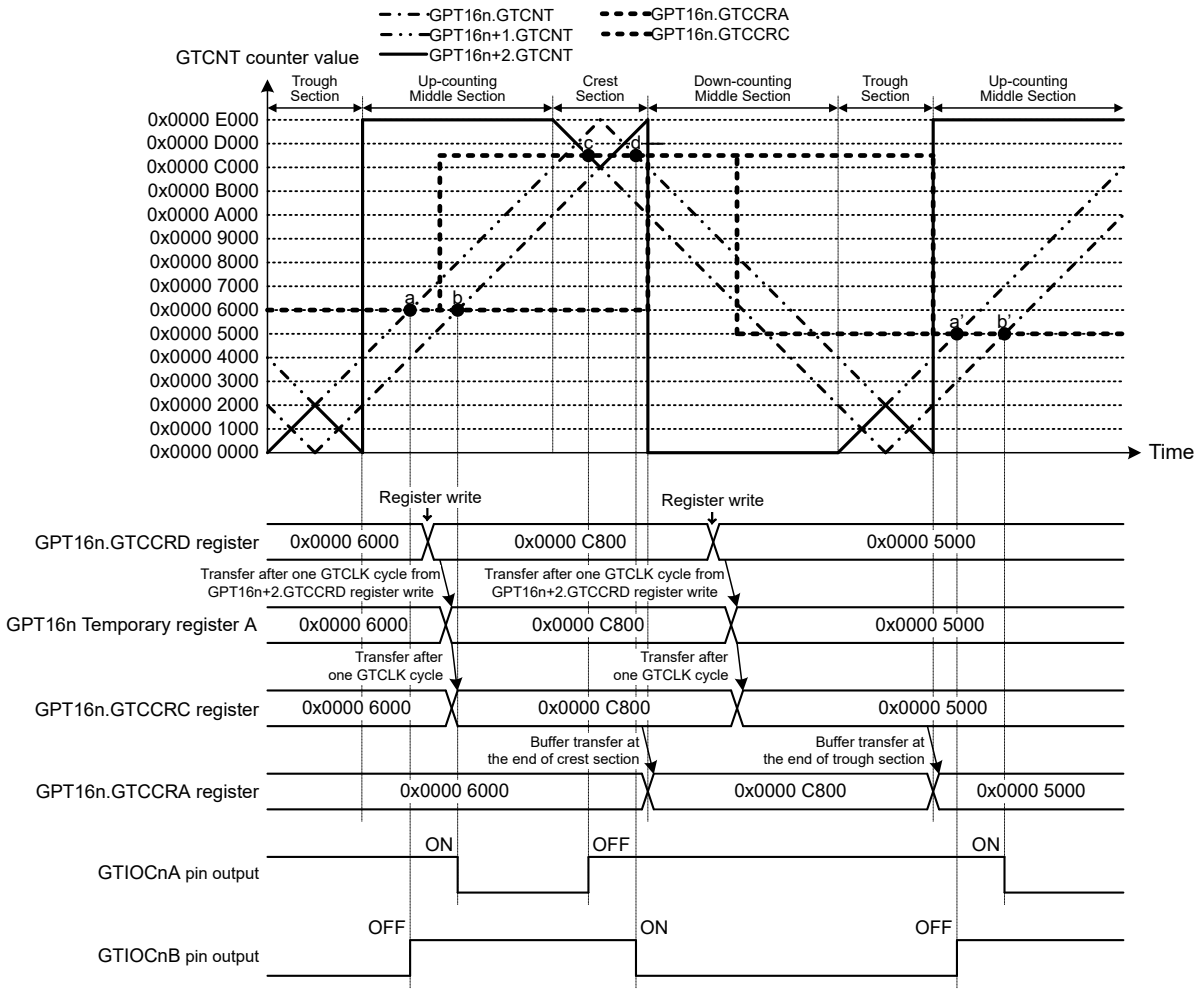


Figure 20.55 Example of compare match generation sequence operation in complementary PWM mode (complementary PWM mode 3 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: a → b → c → d) (n = 0)

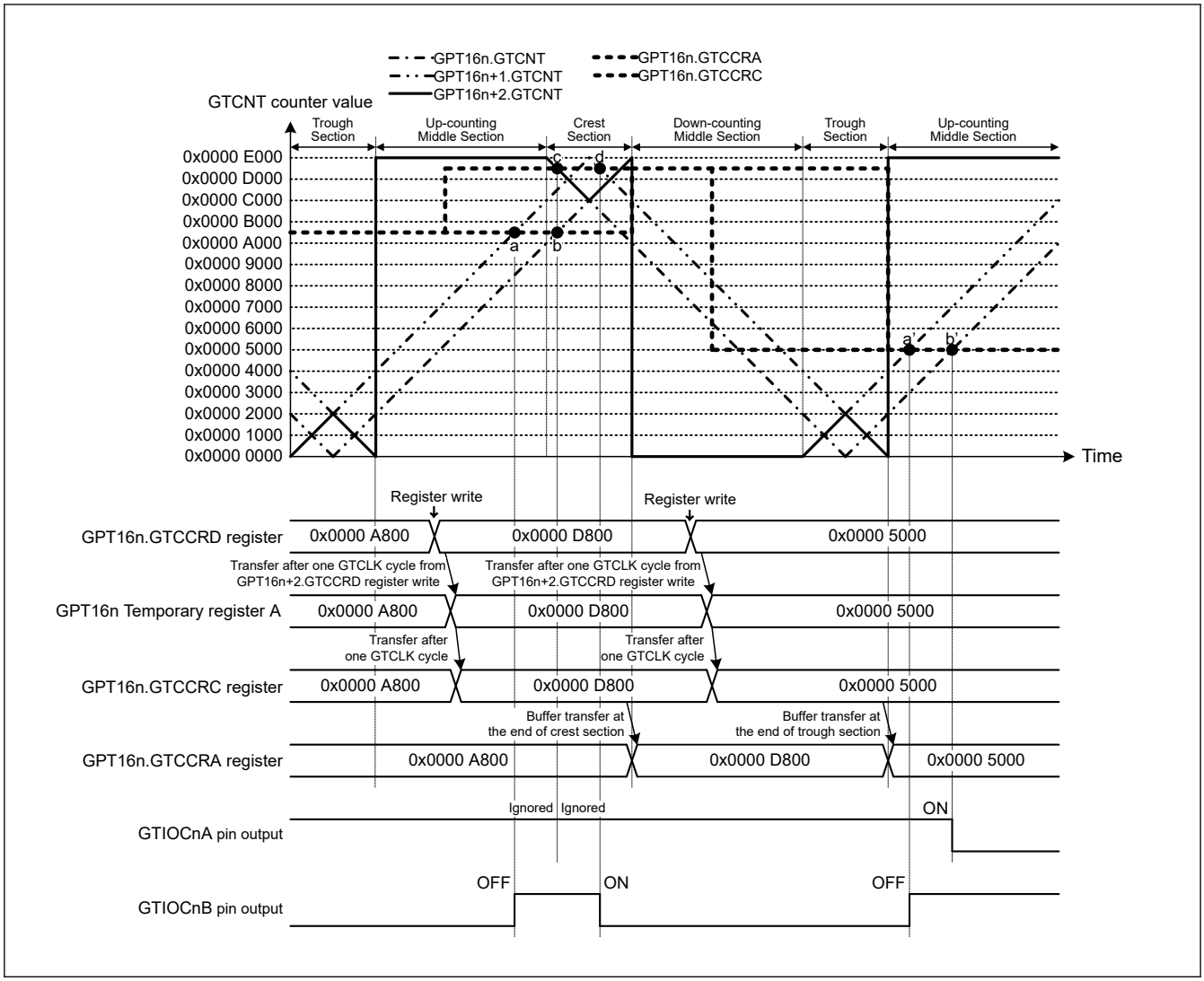


Figure 20.56 Example of compare match generation sequence operation in complementary PWM mode (complementary PWM mode 3 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: a → (b, c) → d) (n = 0)

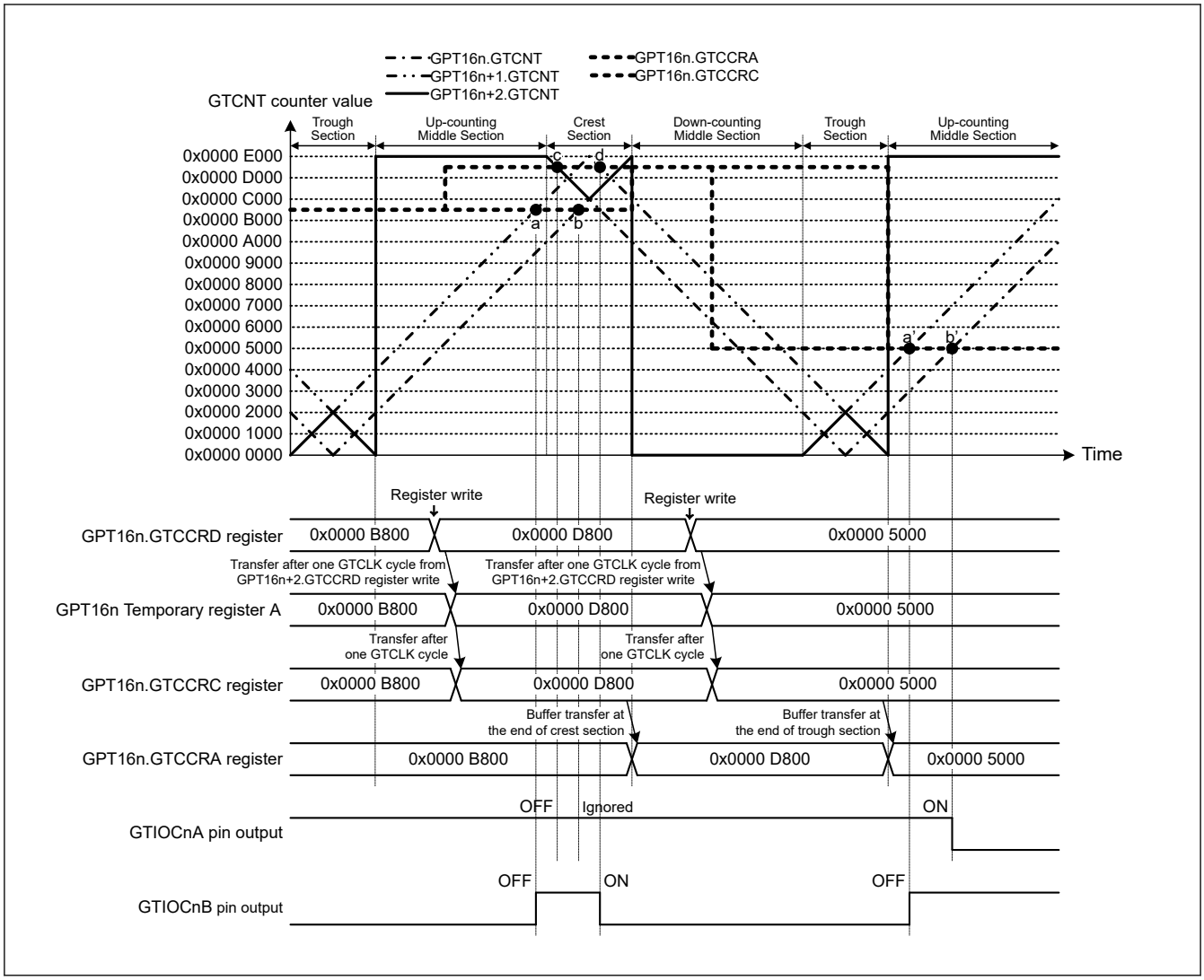


Figure 20.57 Example of compare match generation sequence operation in complementary PWM mode (complementary PWM mode 3 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: a → c → b → d) (n = 0)

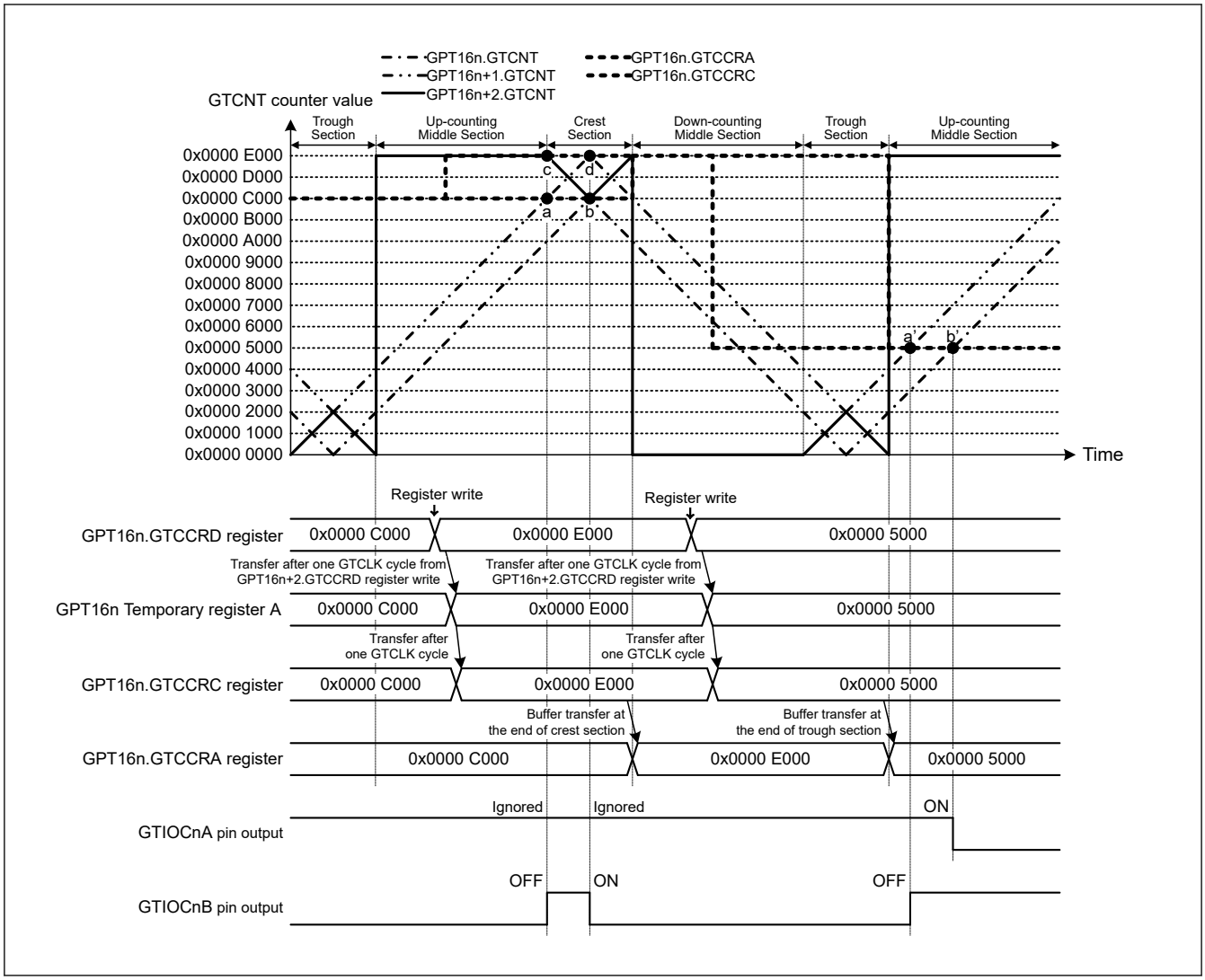


Figure 20.58 Example of compare match generation sequence operation in complementary PWM mode (complementary PWM mode 3 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: (a, c) → (b, d)) (n = 0)

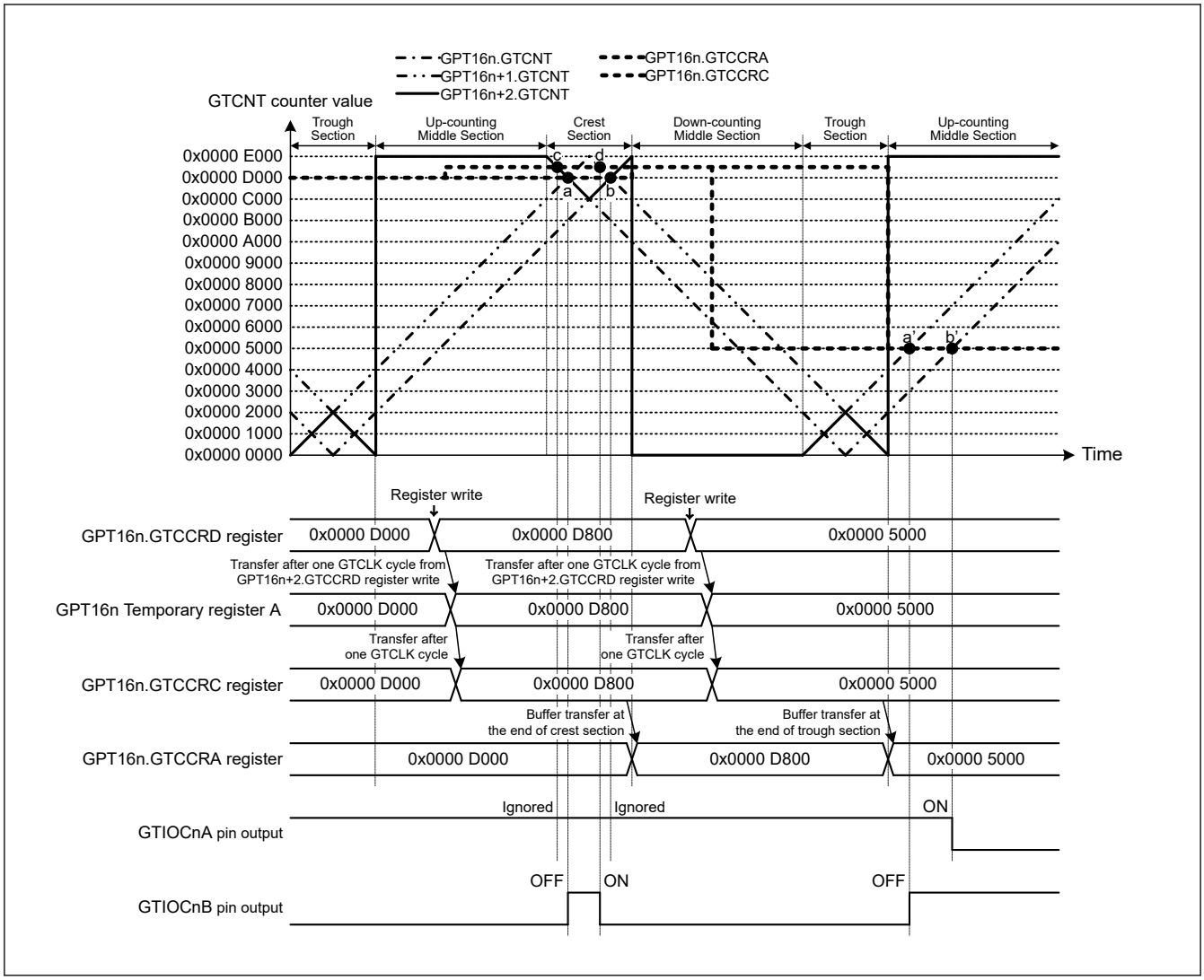


Figure 20.59 Example of compare match generation sequence operation in complementary PWM mode (complementary PWM mode 3 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: c → a → d → b) (n = 0)

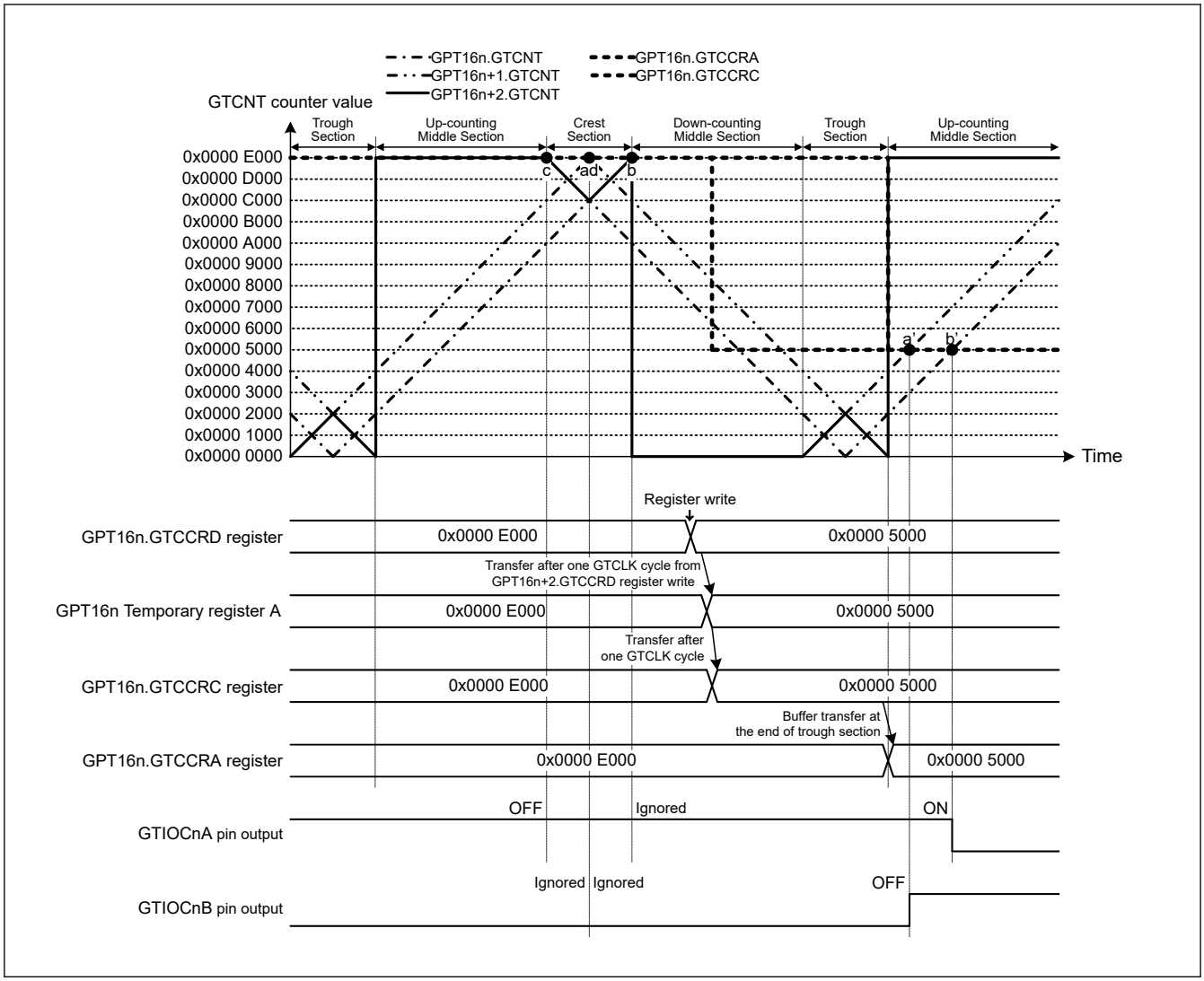


Figure 20.60 Example of compare match generation sequence operation in complementary PWM mode (complementary PWM mode 3 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: c → (a, d) → b) (n = 0)

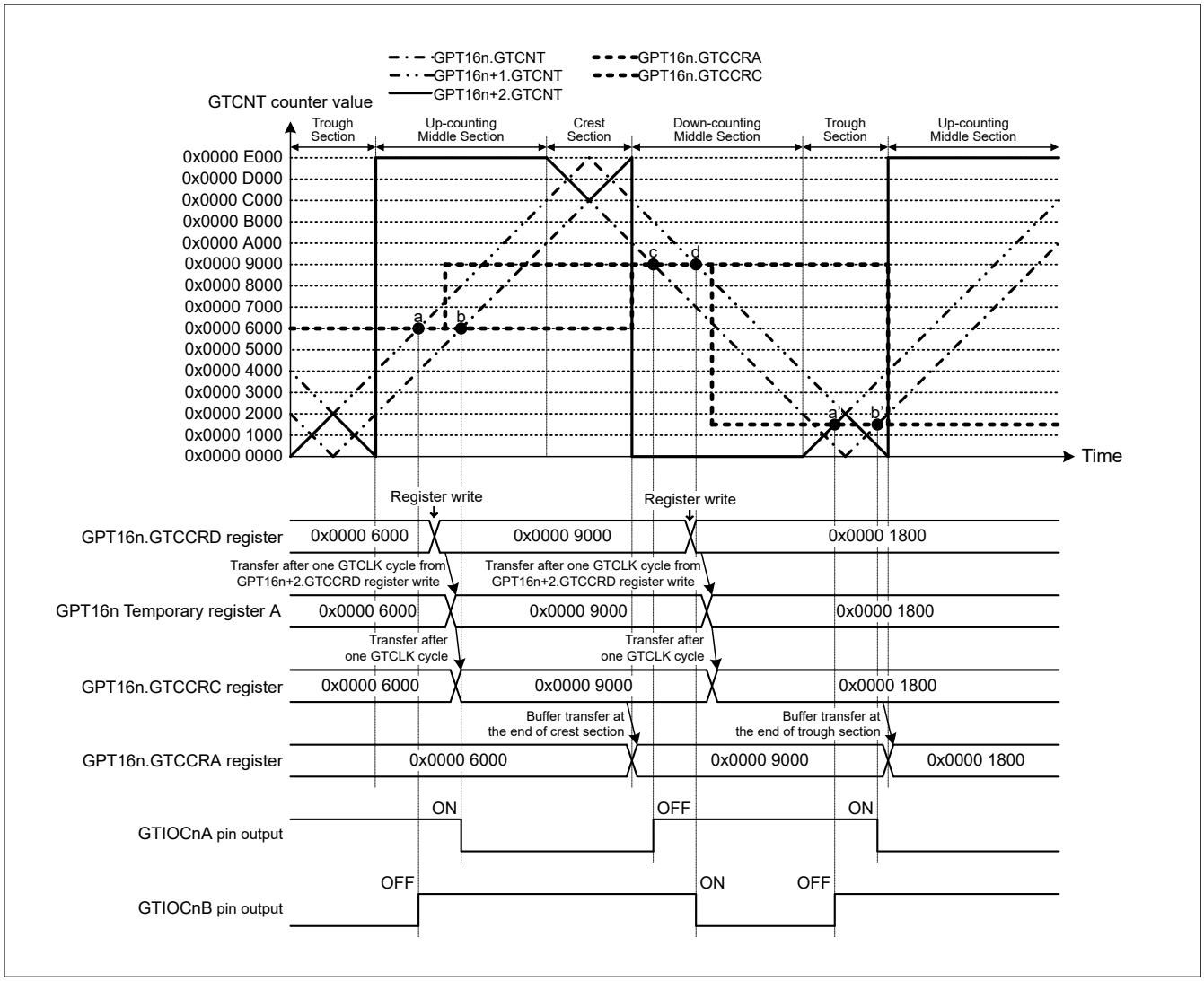


Figure 20.61 Example of compare match generation sequence operation in complementary PWM mode (complementary PWM mode 3 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: c → d → a' → b') (n = 0)

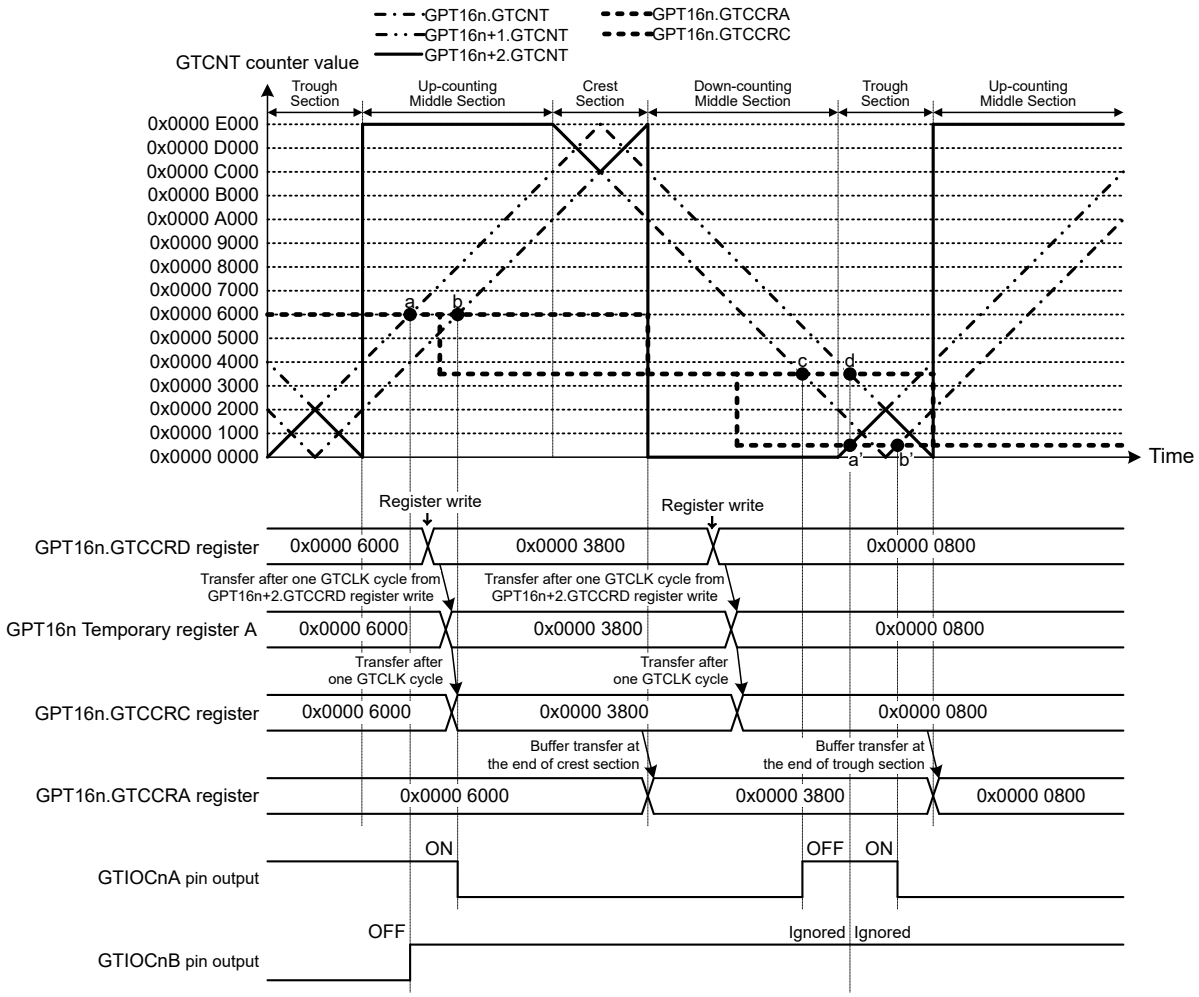


Figure 20.62 Example of compare match generation sequence operation in complementary PWM mode (complementary PWM mode 3 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: c → (d, a') → b') (n = 0)

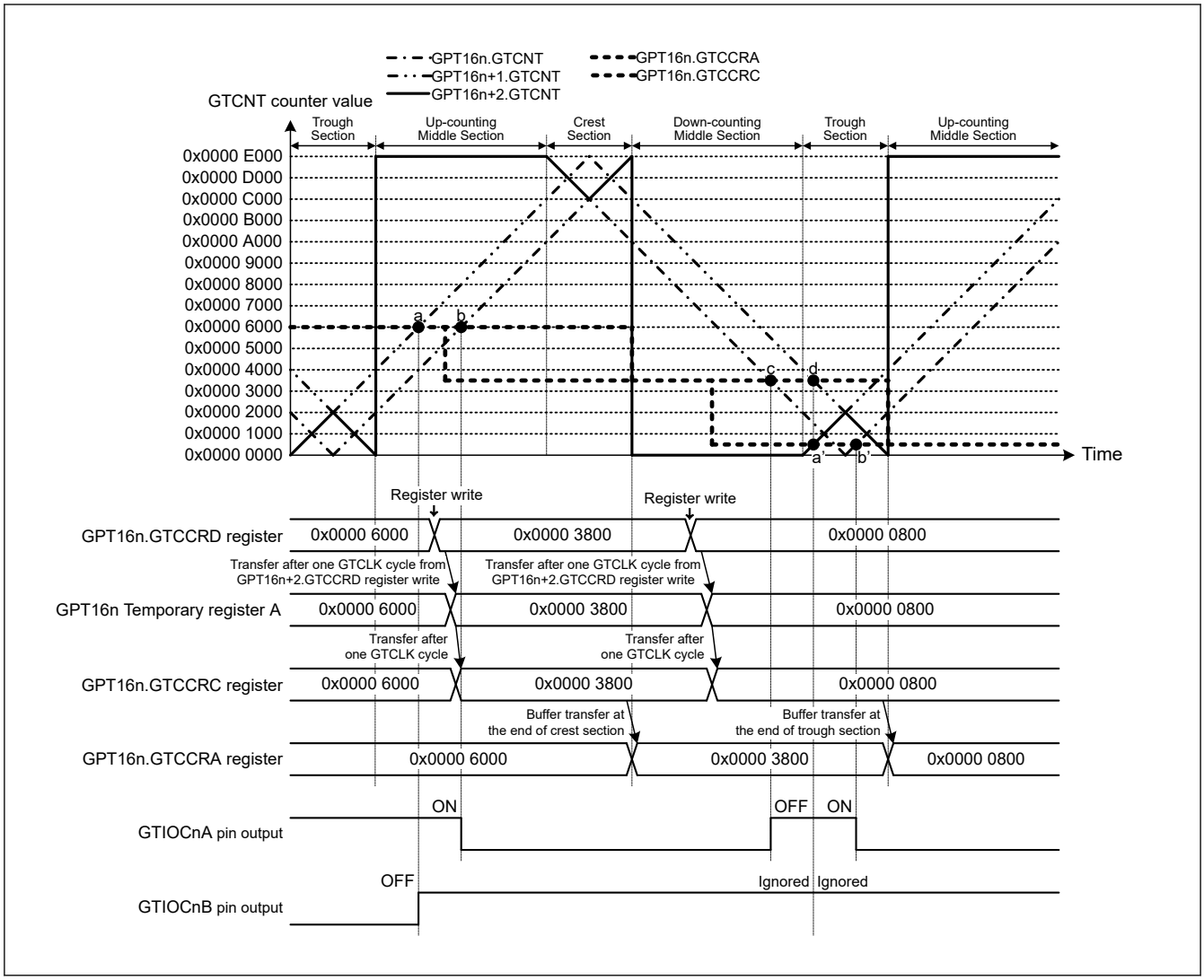


Figure 20.63 Example of compare match generation sequence operation in complementary PWM mode (complementary PWM mode 3 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: c → a' → d → b') (n = 0)

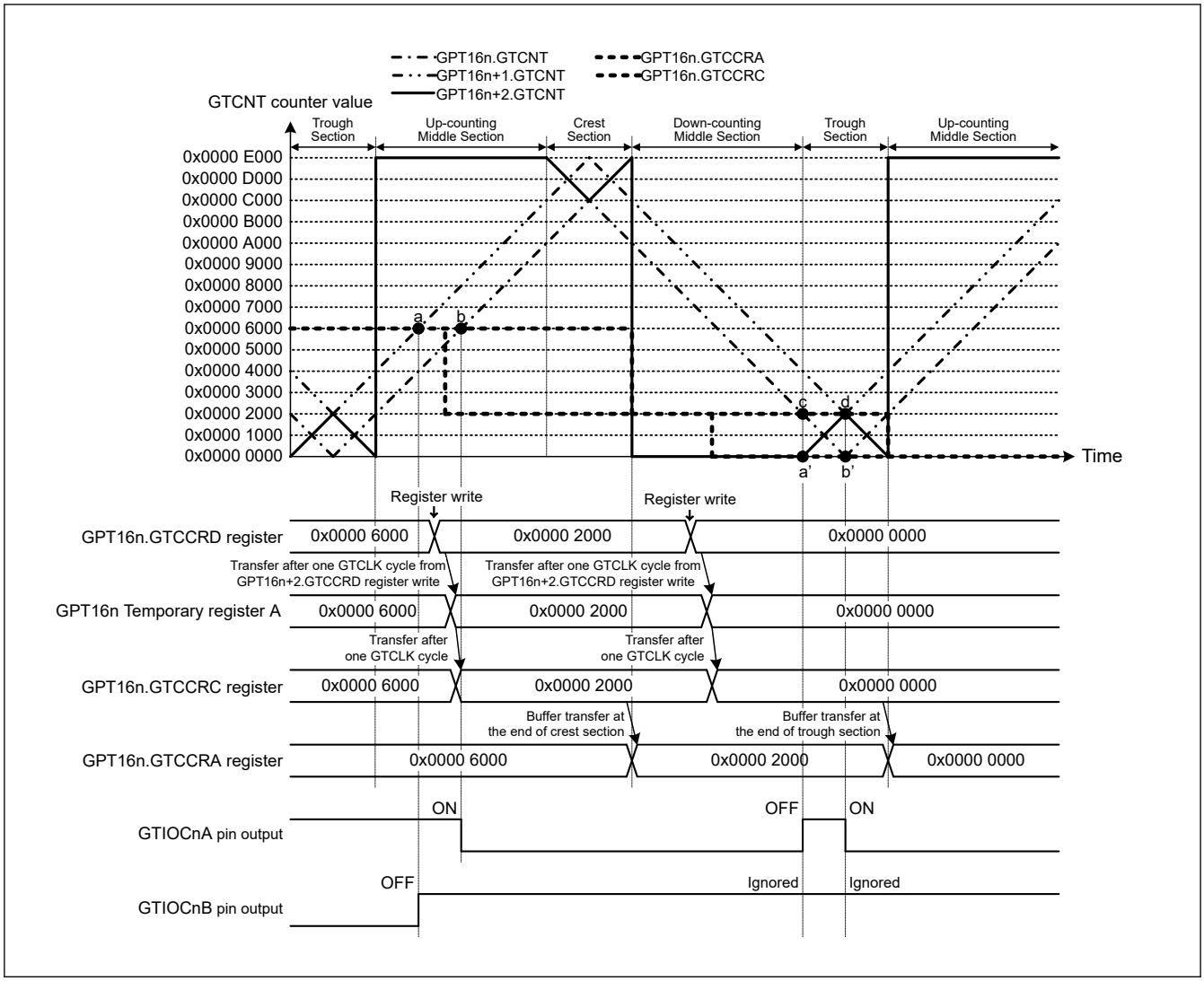


Figure 20.64 Example of compare match generation sequence operation in complementary PWM mode (complementary PWM mode 3 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: (c, a') → (d, b')) (n =0)

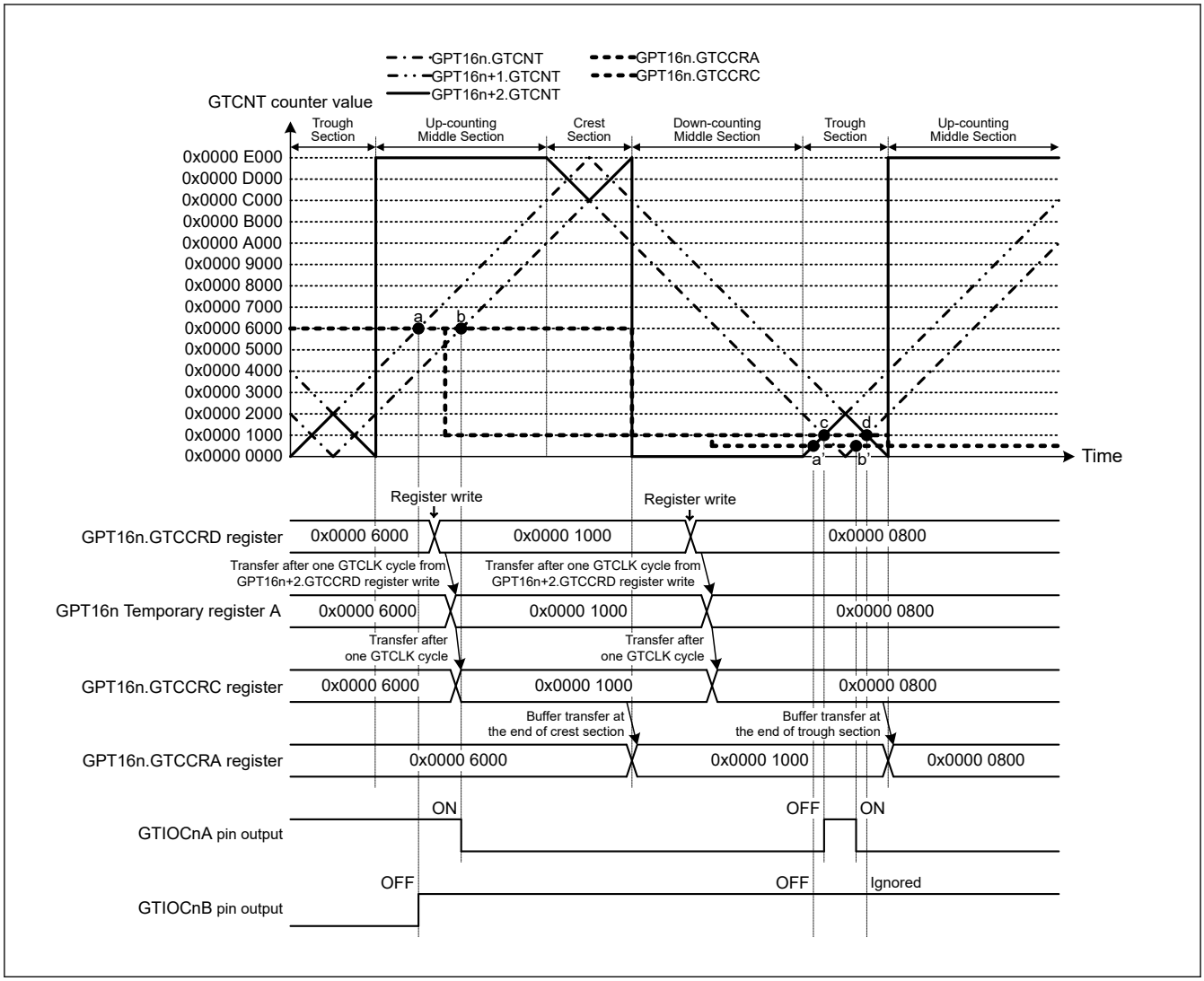


Figure 20.65 Example of compare match generation sequence operation in complementary PWM mode (complementary PWM mode 3 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: a' → c → b' → d) (n =0)

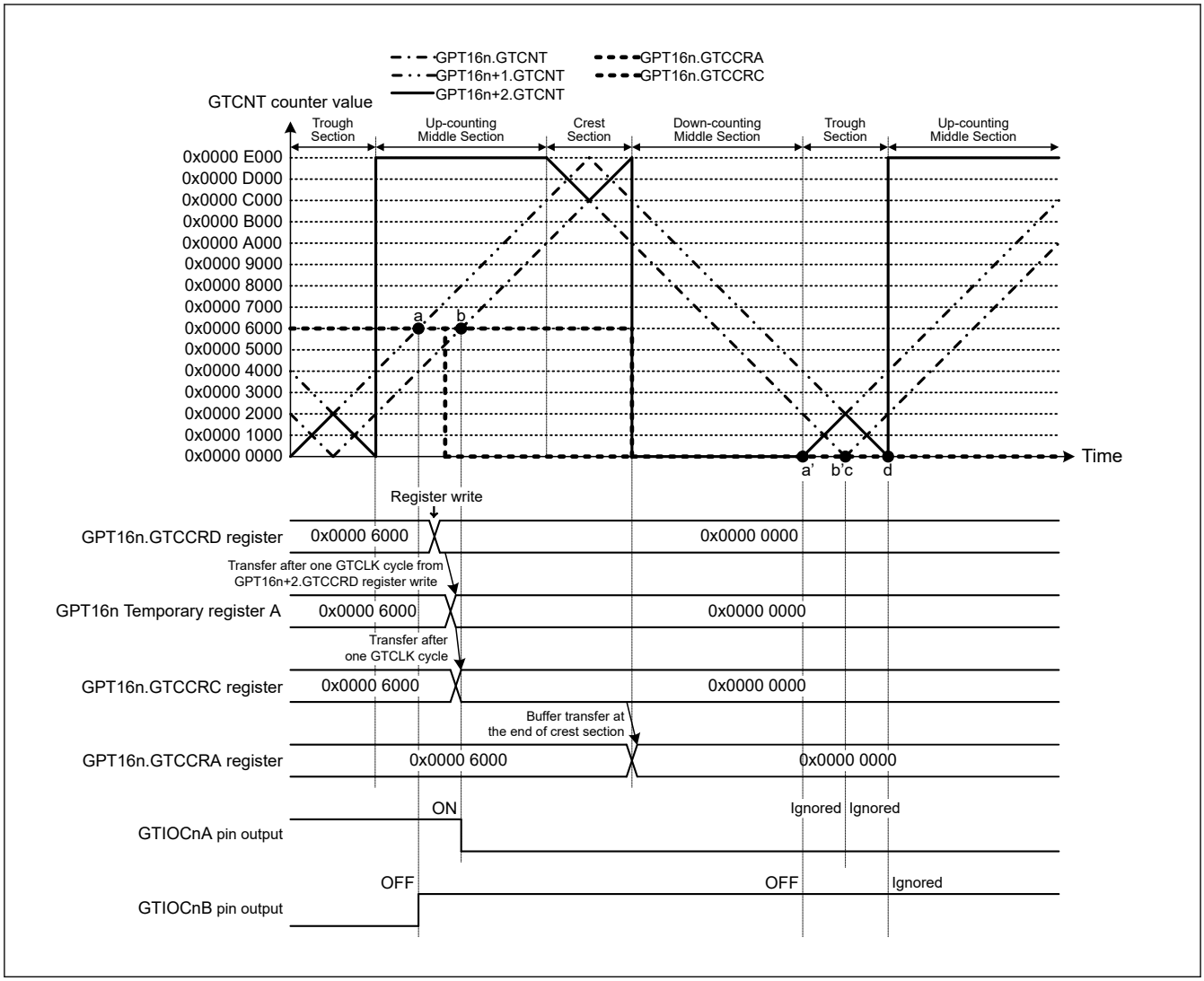


Figure 20.66 Example of compare match generation sequence operation in complementary PWM mode (complementary PWM mode 3 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 2000, compare match generation order: a' → (b', c) → d) (n = 0)

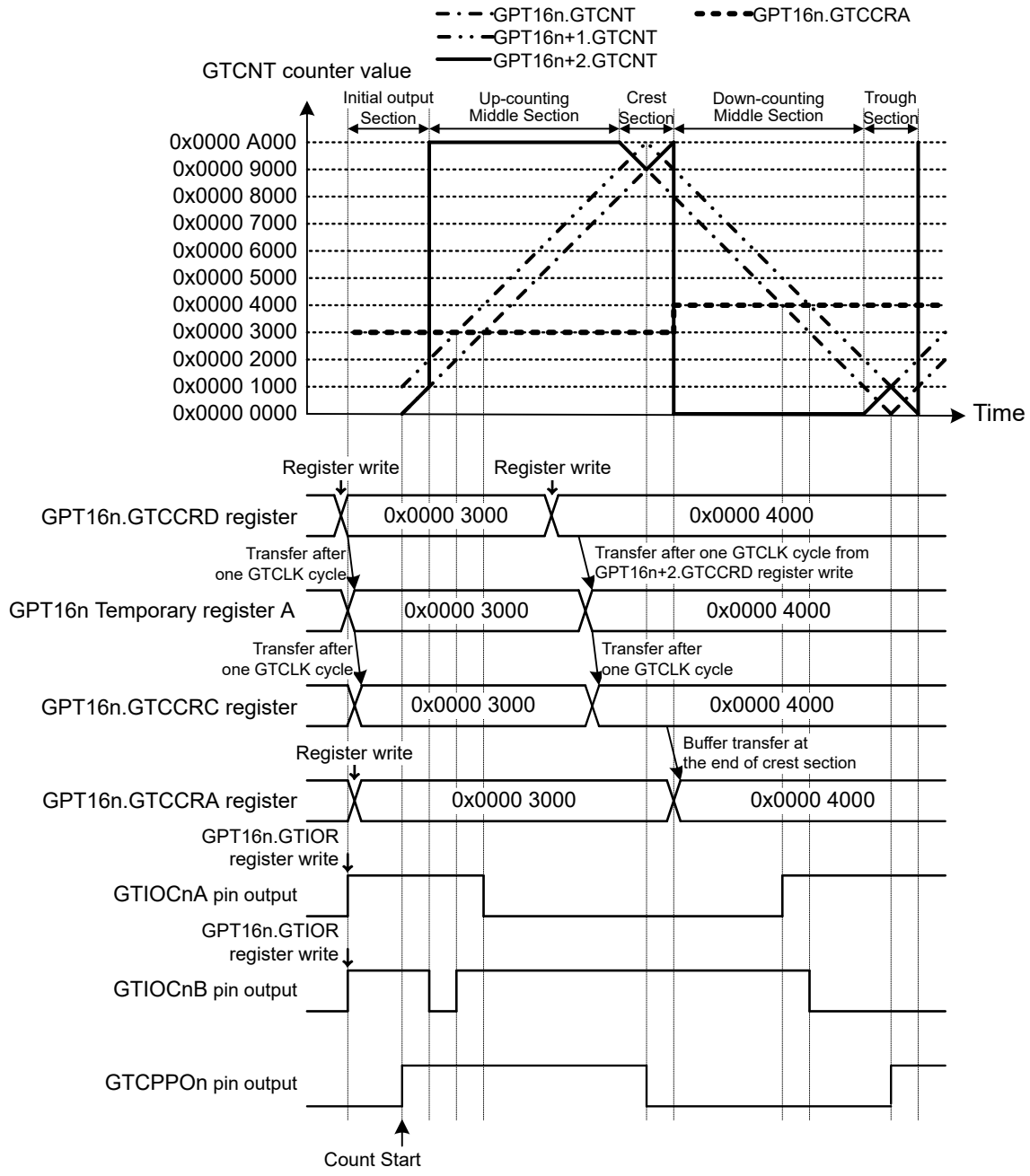


Figure 20.67 Example of complementary PWM mode initial output operation (complementary PWM mode 1 operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000, in the case that the initial GTCCRA register value is larger than the dead time value) (n = 0)

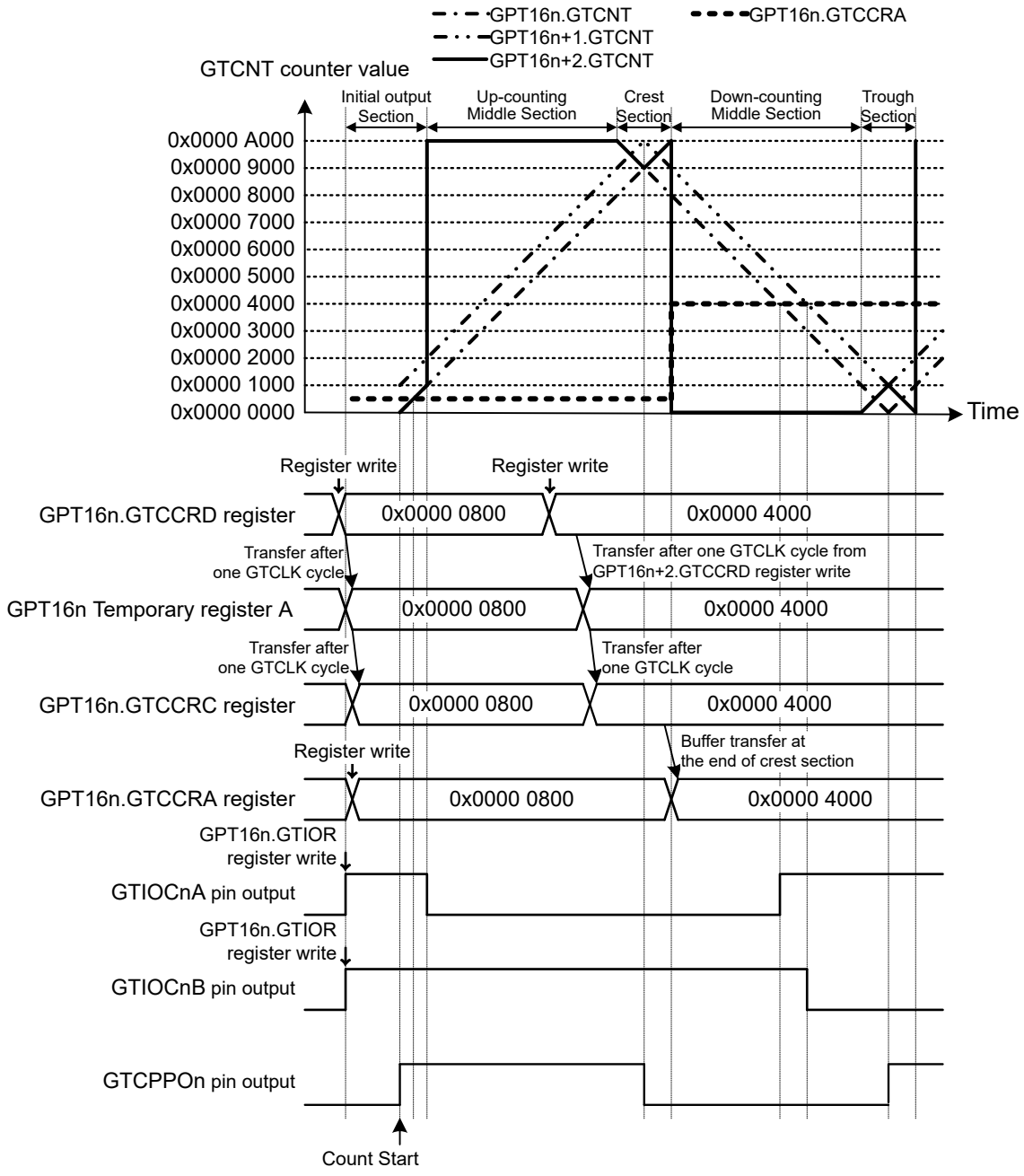


Figure 20.68 Example of complementary PWM mode initial output operation (complementary PWM mode 1 operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, the dead time value is 0x0000 1000, in the case that the initial GTCCRA register value is equal to or less than the dead time value) (n = 0)

Table 20.35 Example for setting complementary PWM modes 1, 2, 3 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[3:0] bits of GPT16n channel.
2	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits of GPT16n channel.
3	Set cycle	Set the cycle in GTPR, GTPBR, GTPDBR of GPT16n channel.

Table 20.35 Example for setting complementary PWM modes 1, 2, 3 (2 of 2)

No.	Step Name	Description
4	Set GTIOCnm /GTIOCn+1m / GTIOCn+2m pin function	Set the function of the GTIOCnm, GTIOCn+1m, and GTIOCn+2m pins with GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register of the GPT16n, GPT16n+1, and GPT16n+2 channels.
5	Enable GTCPPOn pin output	Set to enable or disable PWM synchronous output from the GTCPPOn pin with the PSYE bit in the GTIOR register of the GPT16n channel.
6	Enable GTIOCnm / GTIOCn+1m /GTIOCn+2m pin output	Set to enable the output from the GTIOCnm, GTIOCn+1m, and GTIOCn+2m pins with the OAE and OBE bits in the GTIOR register of the GPT16n, GPT16n+1, and GPT16n+2 channels.
7	Set buffer operation	In complementary PWM mode 3, set buffer operation with the GTBER2.CP3DB bit of the GPT16n, GPT16n+1, and GPT16n+2 channels.
8	Set compare match value	Set the output pin changing point during up-counting after count start in the GTCCRA register of the GPT16n, GPT16n+1, and GPT16n+2 channels.
9	Set buffer value	For single buffer operation, set data to be transferred for the first buffer transfer to the GTCCRA register after count start in the GTCCRD register. For buffer operation in complementary PWM mode 2, set the same value as the GTCCRA register in the GTCCRD register because the first buffer transfer does not take place. For double buffer operation in complementary PWM mode 3, set data to be transferred at the first end of crest in the GTCCRD register and set data to be transferred at the first end of trough in the GTCCRF register.
10	Set dead time value	Set the dead time value in GTDVU of GPT16n channel.
11	Start count operation	Set GTCR.CST of GPT16n channel to 1 to start count operation.
12	Set buffer value for each cycle	For single buffer operation, set data to be transferred for the next buffer transfer to the GTCCRA register in the GTCCRD register. For double buffer operation in complementary PWM mode 3, set data to be transferred for the next buffer transfer in the GTCCRF register. Finally, make settings for the GPT16n+2.GTCCRD register (data is transferred to the temporary register).

Note: n = 0
m = A, B

20.3.3.8 Complementary PWM Mode 4

In complementary PWM mode 4, the value written to the GTCCRD and GTCCRF registers is immediately applied to compare match operation by transferring data also to the GTCCRA register during buffer transfer to a temporary register before crest or trough transfer timing.

Figure 20.69 shows the block diagram in complementary PWM mode 4.

In the configuration of complementary PWM mode 4, a buffer transfer path from the GTCCRD register to the GTCCRC and GTCCRA registers and a buffer transfer path from the GTCCRF register to the GTCCRE and GTCCRA registers are added to other complementary PWM modes shown in Table 20.35.

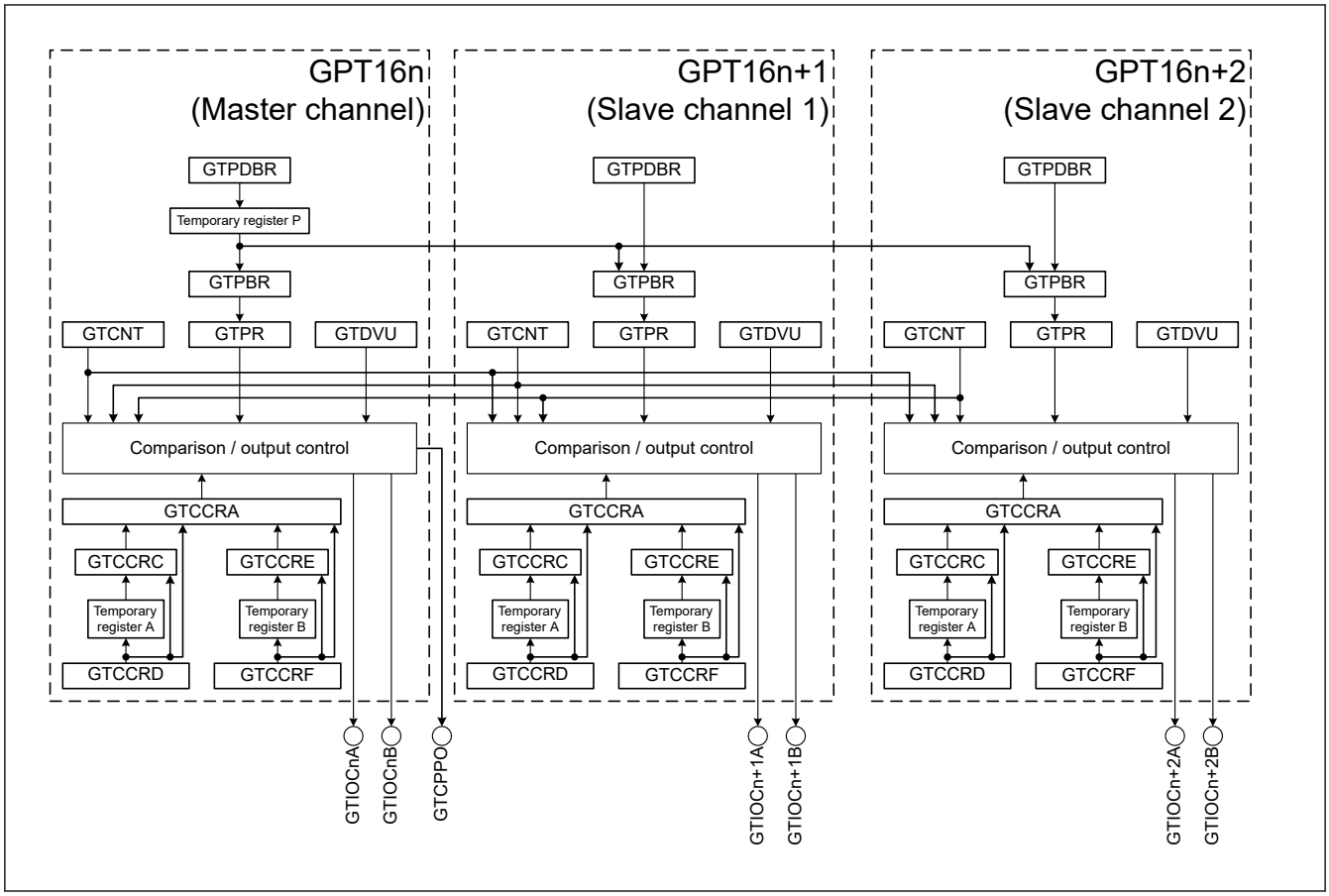


Figure 20.69 Block diagram in complementary PWM mode 4 (n = 0)

Count operation is performed in the same way as complementary PWM modes 1 to 3. See Table 20.30 and Table 20.31.

With respect to buffer operation and PWM waveform changes in complementary PWM mode 4, buffer transfer (shown in Figure 20.69) from the GTCCRD and GTCCRF registers is added based on the complementary PWM mode 3 operation. Buffer transfer and PWM waveform are controlled according to operation section, state of comparison with the GTCCRA register, and write value.

The double buffer function by writing to the GTCCRF register can be enabled or disabled with the GTBER2.CP3DB bit. In double buffer operation, the value written to the GTCCRD register is used as a compare match value for positive-phase OFF (negative-phase ON) during down-counting, and the value written to the GTCCRF register is used as a compare match value for negative-phase OFF (positive-phase ON) during up-counting. Transfer register, transfer value, and PWM output changes are controlled by operation section (where the value is written), state of comparison with the GTCCRA register, and write value.

Double buffer operation is guaranteed only in the up-counting middle section and down-counting middle section. Setting a value not larger than the dead time value and not less than the count cycle is prohibited.

In single buffer operation, a compare match value is written only to the GTCCRD register, and transfer register, transfer value, and PWM output changes are controlled by operation section (where the value is written), state of comparison with the GTCCRA register, and write value.

Transfer from the GTCCRD register to the temporary register A and transfer from the GTCCRF register to the temporary register B are performed in the same way as other complementary PWM modes. Transfers are concurrently performed in three channels by writing a value to the GPT16n+2.GTCCRD register. Transfer from the GTCCRD register to the GTCCRC register, GTCCRA register, temporary register B, and GTCCRE register and transfer from the GTCCRF register to the GTCCRE register and the GTCCRA register are performed at the same time as the above-mentioned transfer to the temporary register.

Table 20.36 and Table 20.37 show immediate buffer transfer (for transfer to the temporary register by writing a value to the GPT16n+2.GTCCRD register) to the GTCCRC and GTCCRA registers by writing a value to the GTCCRD register during single buffer operation in complementary PWM mode 4 for each compare match state in each operation section. Transfers (from the GTCCRD register to the temporary register A, from the temporary register A to the GTCCRC register, and from

the GTCCRC register to the GTCCRA register) other than those shown in Table 20.36 and Table 20.37 are the same as single buffer transfer in complementary PWM mode 3 shown in Table 20.32.

Table 20.38 and Table 20.39 show immediate buffer transfer (for transfer to the temporary register by writing a value to the GPT16n+2.GTCCRD register) to each register by writing a value to the GTCCRD and GTCCRF registers during double buffer operation in complementary PWM mode 4 for each compare match state in each operation section. Transfers (from the GTCCRD register to the temporary register A, from the GTCCRF register to the temporary register B, from the temporary register A to the GTCCRC register, from the temporary register B to the GTCCRE register, and from the GTCCRC and GTCCRE registers to the GTCCRA register) other than those shown in Table 20.38 and Table 20.39 are the same as double buffer transfer in complementary PWM mode 3 shown in Table 20.33.

Table 20.36 Immediate single buffer transfer from GTCCRD register in complementary PWM mode 4 (1 of 2)

Operation section	Compare match state	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Up-counting middle section	Before up-counting compare match	GTCCRD	(i) When GTCCRD > GPT16n+1.GTCNT GTCCRD (ii) When GTCCRD ≤ GPT16+1.GTCNT GPT16n+1.GTCNT Negative-phase OFF
	Up-counting dead time period	GTCCRD	No transfer
	After up-counting compare match	GTCCRD	No transfer
Up-counting crest section	Before up-counting compare match	Before down-counting compare match GTCCRD After down-counting dead time start No transfer	Before down-counting compare match (i) When GTCCRD > GPT16n+1.GTCNT GTCCRD (ii) When GTCCRD ≤ GPT16n+1.GTCNT GPT16n+1.GTCNT Negative-phase OFF After down-counting dead time start No transfer
	Up-counting dead time period	Before down-counting compare match GTCCRD After down-counting dead time start No transfer	No transfer
	After up-counting compare match	Before down-counting compare match (i) When GTCCRD < GPT16n+2.GTCNT GTCCRD (ii) When GTCCRD ≥ GPT16n+2.GTCNT GPT16n+2.GTCNT Positive-phase OFF After down-counting dead time start No transfer	No transfer

Table 20.36 Immediate single buffer transfer from GTCCRD register in complementary PWM mode 4 (2 of 2)

Operation section	Compare match state	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Down-counting crest section	Before down-counting compare match	Up-counting dead time period (i) When GTCCRD < GPT16n+1.GTCNT GTCCRD (ii) When GTCCRD ≥ GPT16n+1.GTCNT GPT16n+1.GTCNT Negative-phase ON After up-counting compare match (i) When GTCCRD < GPT16n.GTCNT GTCCRD (ii) When GTCCRD ≥ GPT16n.GTCNT GPT16n.GTCNT Positive-phase OFF	No transfer
	Down-counting dead time period	No transfer	No transfer
	After down-counting compare match	No transfer	No transfer

Table 20.37 Immediate single buffer transfer from GTCCRD register in complementary PWM mode 4 (1 of 2)

Operation section	Compare match state	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Down-counting middle section	Before down-counting compare match	GTCCRD	(i) When GTCCRD < GPT16n.GTCNT GTCCRD (ii) When GTCCRD ≥ GPT16n.GTCNT GPT16n.GTCNT Positive-phase OFF
	Down-counting dead time period	GTCCRD	No transfer
	After down-counting compare match	GTCCRD	No transfer

Table 20.37 Immediate single buffer transfer from GTCCRD register in complementary PWM mode 4 (2 of 2)

Operation section	Compare match state	Immediate Transfer Destination Register	
		GTCCRC	GTCCRA
Down-counting trough section	Before down-counting compare match	Before up-counting compare match GTCCRD After up-counting dead time start No transfer	Before up-counting compare match (i) When GTCCRD < GPT16n.GTCNT GTCCRD (ii) When GTCCRD ≥ GPT16n.GTCNT GPT16n.GTCNT Positive-phase OFF After up-counting dead time start No transfer
	Down-counting dead time period	Before up-counting compare match GTCCRD After up-counting dead time start No transfer	No transfer
	After down-counting compare match	Before up-counting compare match (i) When GTCCRD > GPT16n+2.GTCNT GTCCRD (ii) When GTCCRD ≤ GPT16n+2.GTCNT GPT16n+2.GTCNT Negative-phase OFF After up-counting dead time start No transfer	No transfer
Up-counting trough section	Before up-counting compare match	Down-counting dead time period (i) When GTCCRD > GPT16n.GTCNT GTCCRD (ii) When GTCCRD ≤ GPT16n.GTCNT GPT16n.GTCNT Positive-phase ON After down-counting compare match (i) When GTCCRD > GPT16n+1.GTCNT GTCCRD (ii) When GTCCRD ≤ GPT16n+1.GTCNT GPT16n+1.GTCNT Negative-phase OFF	No transfer
	Up-counting dead time period	No transfer	No transfer
	After up-counting compare match	No transfer	No transfer

Table 20.38 Immediate double buffer transfer from GTCCRD and GTCCRF registers in complementary PWM mode 4

Operation section	Compare match state	Immediate Transfer Destination Register		
		GTCCRC	GTCCRE	GTCCRA
Up-counting middle section	Before up-counting compare match	GTCCRD	GTCCRF	(i) When GTCCRF > GPT16n+1.GTCNT GTCCRF (ii) When GTCCRF ≤ GPT16+1.GTCNT GPT16n+1.GTCNT Negative-phase OFF
	Up-counting dead time period	GTCCRD	GTCCRF	No transfer
	After up-counting compare match	GTCCRD	GTCCRF	No transfer

Table 20.39 Immediate double buffer transfer from GTCCRD and GTCCRF registers in complementary PWM mode 4

Operation section	Compare match state	Immediate Transfer Destination Register		
		GTCCRC	GTCCRE	GTCCRA
Down-counting middle section	Before down-counting compare match	GTCCRD	GTCCRF	(i) When GTCCRD < GPT16n.GTCNT GTCCRD (ii) When GTCCRD ≥ GPT16n.GTCNT GPT16n.GTCNT Positive-phase OFF
	Down-counting dead time period	GTCCRD	GTCCRF	No transfer
	After down-counting compare match	GTCCRD	GTCCRF	No transfer

Figure 20.70 to Figure 20.74 show examples of single buffer operation for each operation section in complementary PWM mode 4.

Figure 20.75 to Figure 20.78 show examples of double buffer operation for each operation section in complementary PWM mode 4.

Table 20.40 shows an example for setting complementary PWM mode 4.

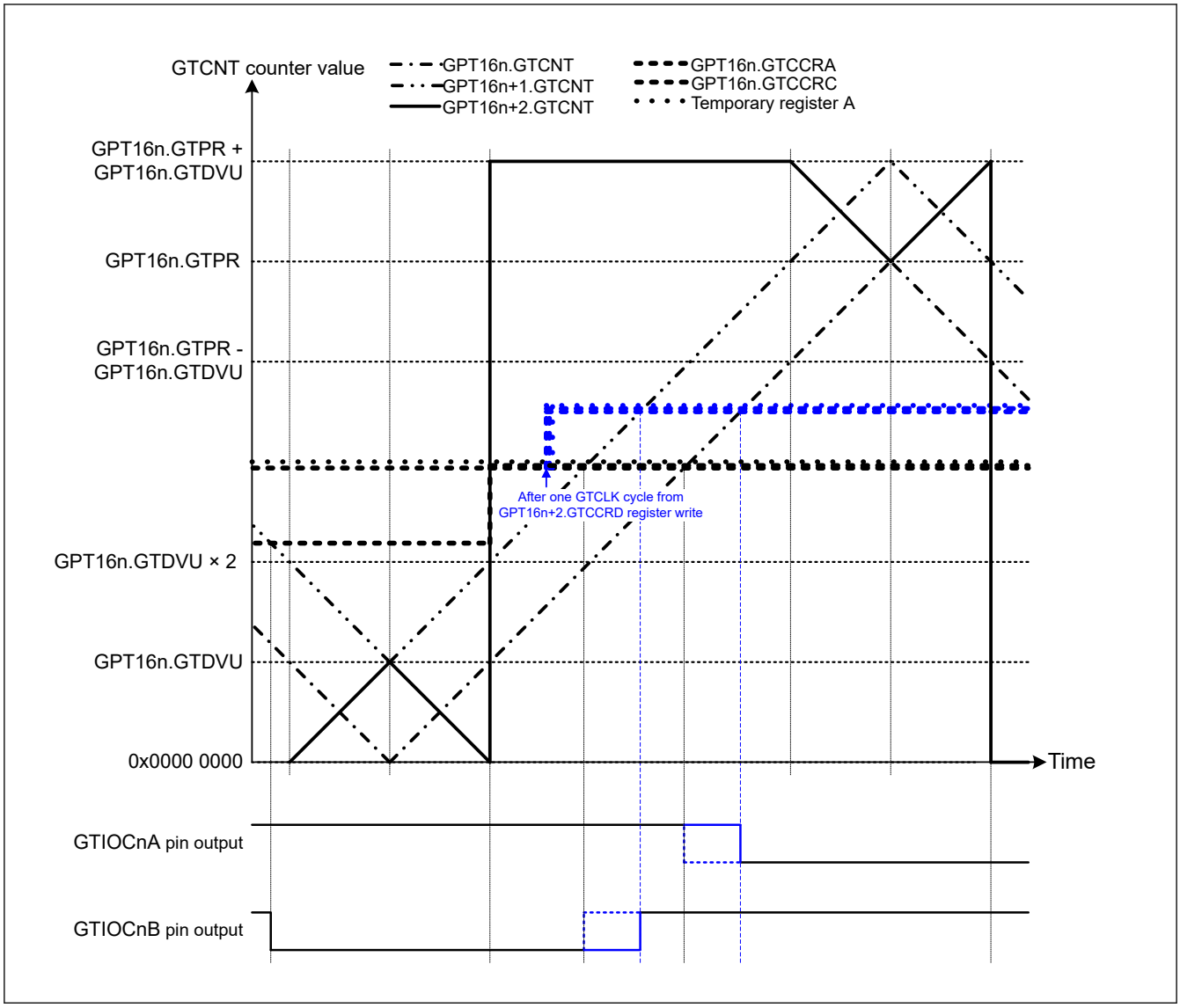


Figure 20.70 **Example of complementary PWM mode 4 single buffer operation (up-counting middle section) (complementary PWM mode 4 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/ GTIOCnB pin = low at GTCCRA register compare match during down-counting, when a value larger than the GPT16n+1.GTCNT value is written to GTCCRD register before up-counting compare match) (n = 0)**

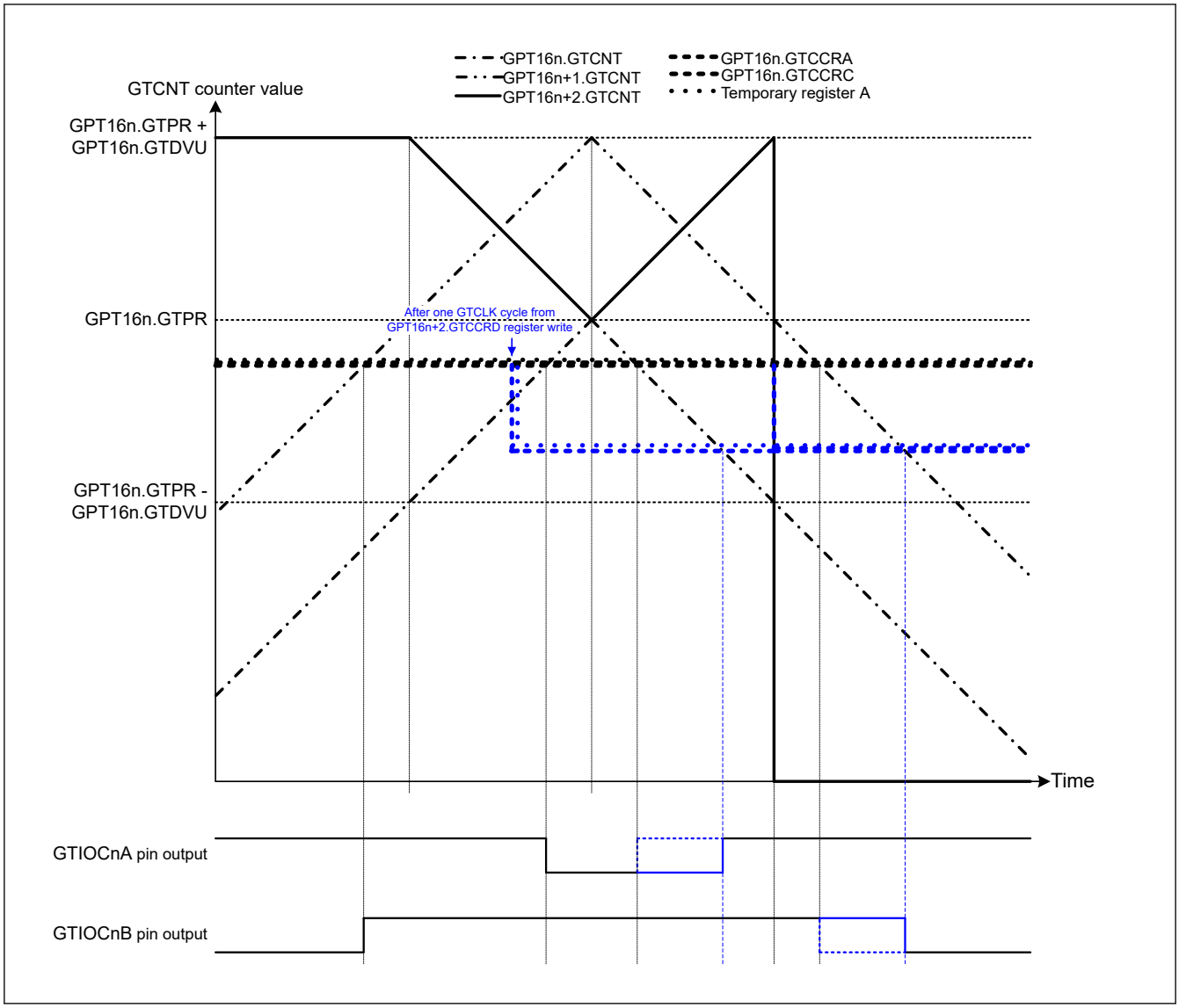


Figure 20.71 Example of complementary PWM mode 4 single buffer operation (up-counting crest section) (complementary PWM mode 4 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/GTIOCnB pin = low at GTCCRA register compare match during down-counting, when a value is written to GTCCRD register during the up-counting dead time) (n = 0)

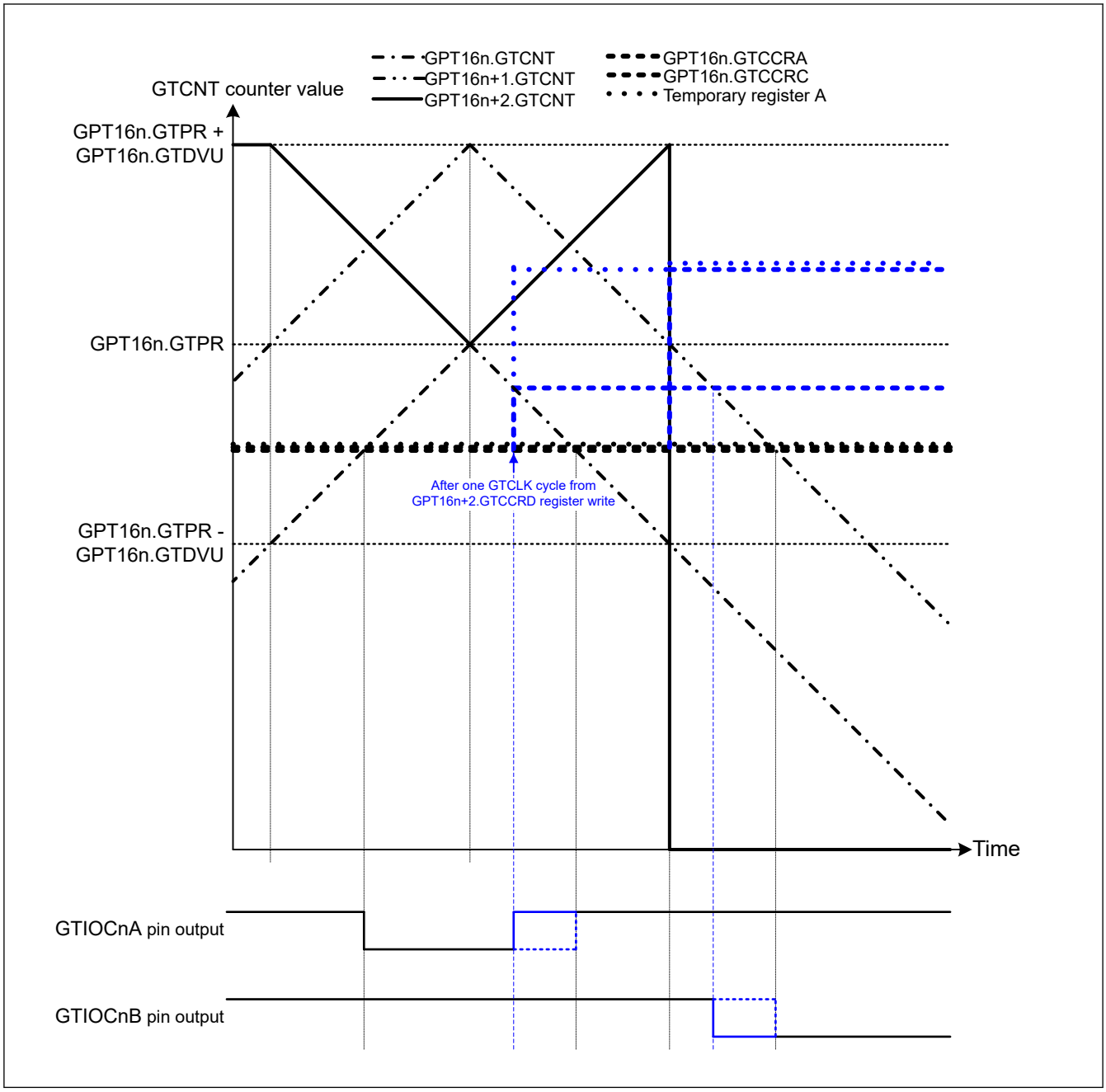


Figure 20.72 Example of complementary PWM mode 4 single buffer operation (down-counting crest section) (complementary PWM mode 4 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/ GTIOCnB pin = low at GTCCRA register compare match during down-counting, when a value not less than the GPT16n.GTCNT value is written to GTCCRD register after up-counting compare match before down-counting compare match) (n = 0)

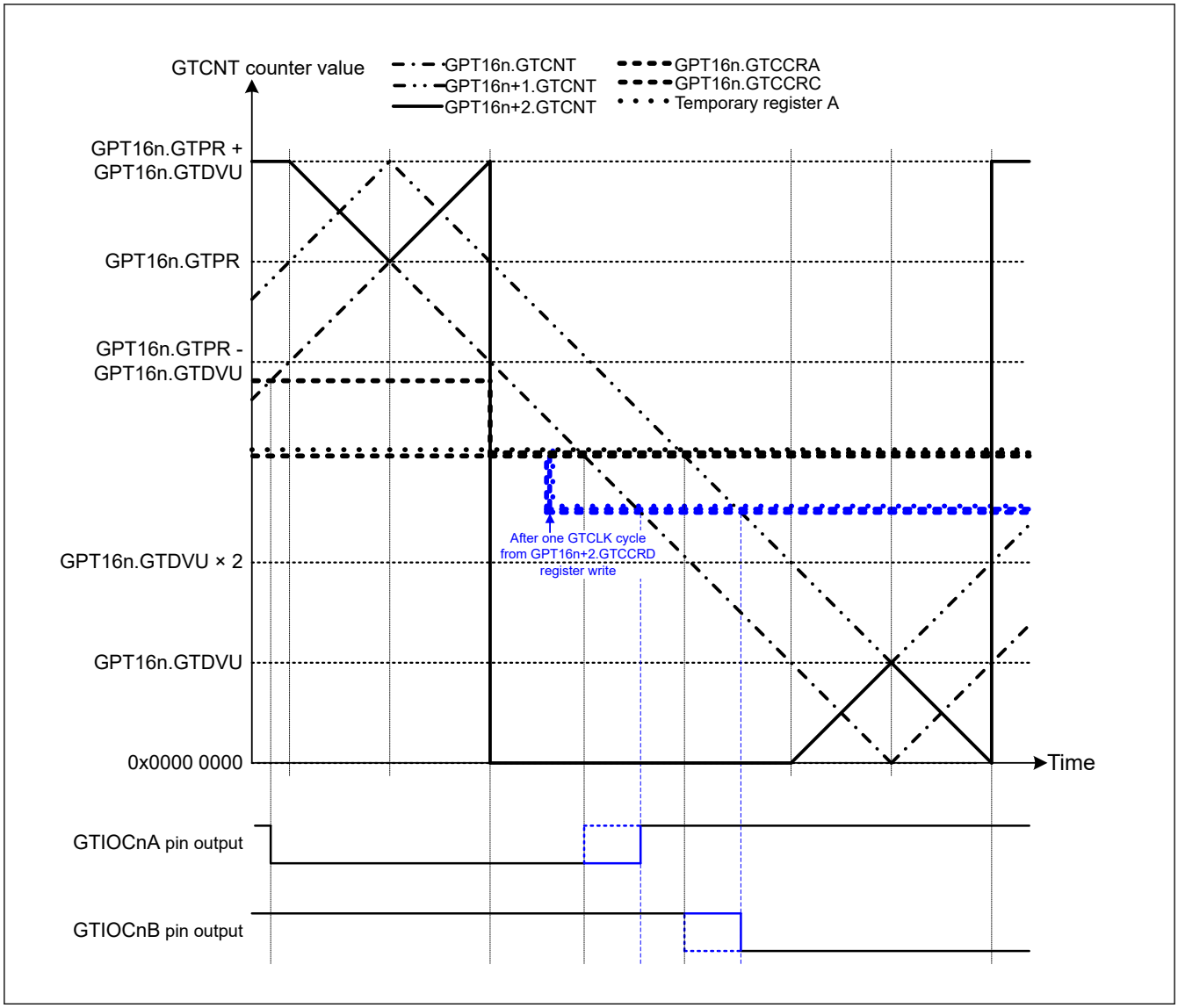


Figure 20.73 **Example of complementary PWM mode 4 single buffer operation (down-counting middle section) (complementary PWM mode 4 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/ GTIOCnB pin = low at GTCCRA register compare match during down-counting, when a value smaller than the GPT16n.GTCNT value is written to GTCCRD register before down-counting compare match) (n = 0)**

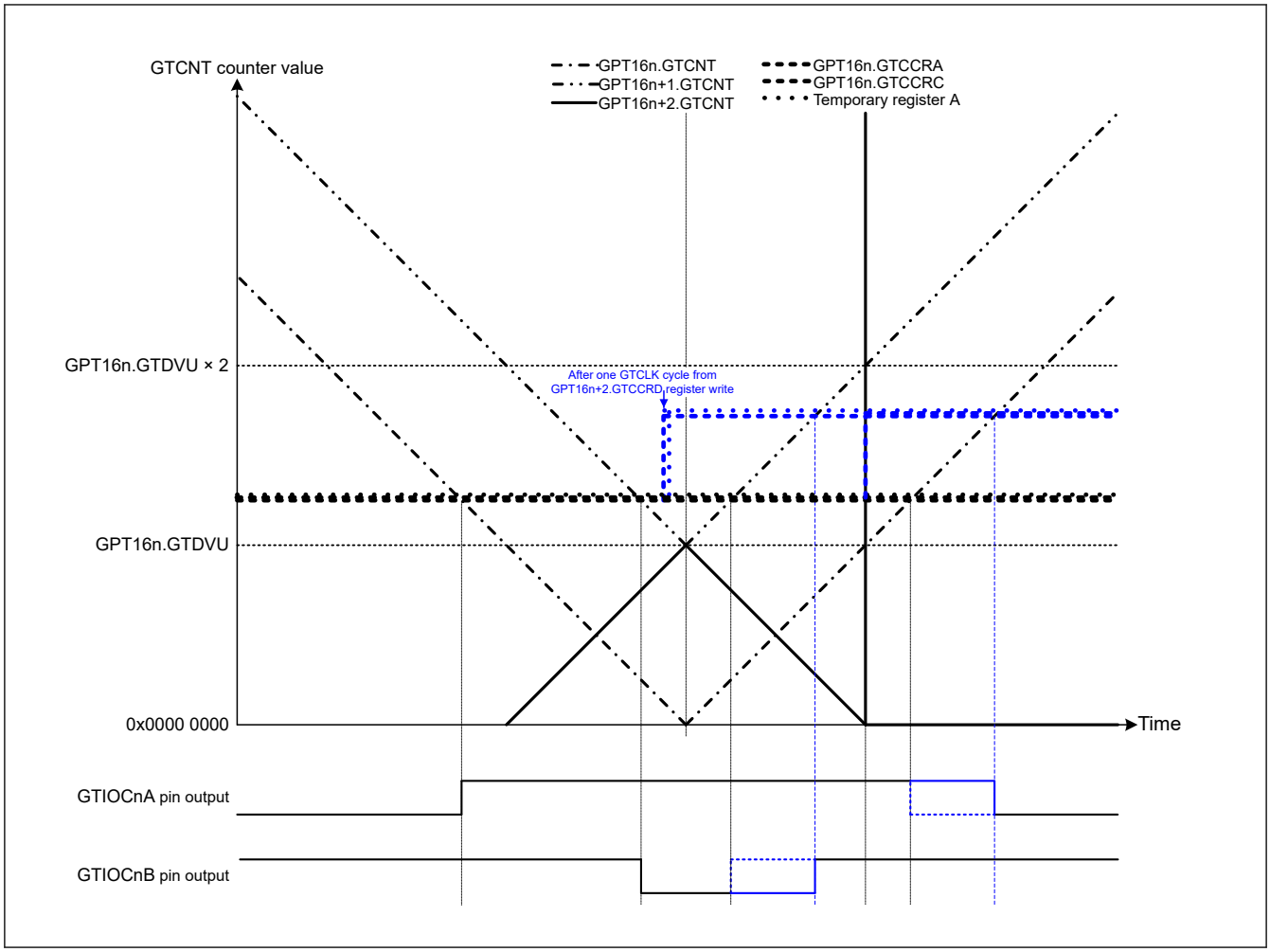


Figure 20.74 Example of complementary PWM mode 4 single buffer operation (down-counting trough section) (complementary PWM mode 4 single buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/ GTIOCnB pin = low at GTCCRA register compare match during down-counting, when a value larger than the GPT16n+2.GTCNT value is written to GTCCRD register after down-counting compare match) (n = 0)

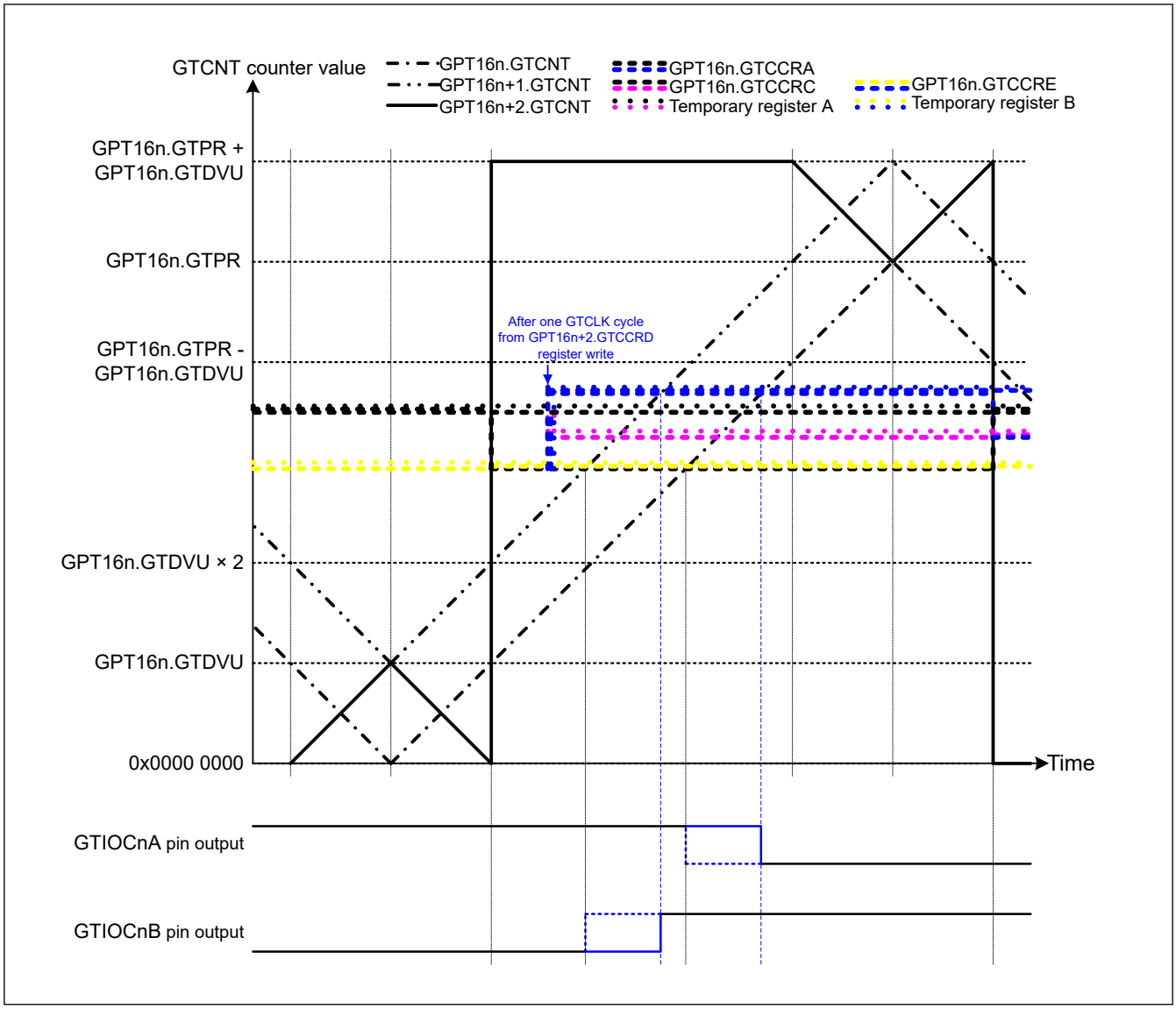


Figure 20.75 Example of complementary PWM mode 4 double buffer operation (up-counting middle section) (complementary PWM mode 4 double buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/ GTIOCnB pin = low at GTCCRA register compare match during down-counting, when a value larger than the GPT16n+1.GTCNT value is written to GTCCRF register before up-counting compare match) (n = 0)

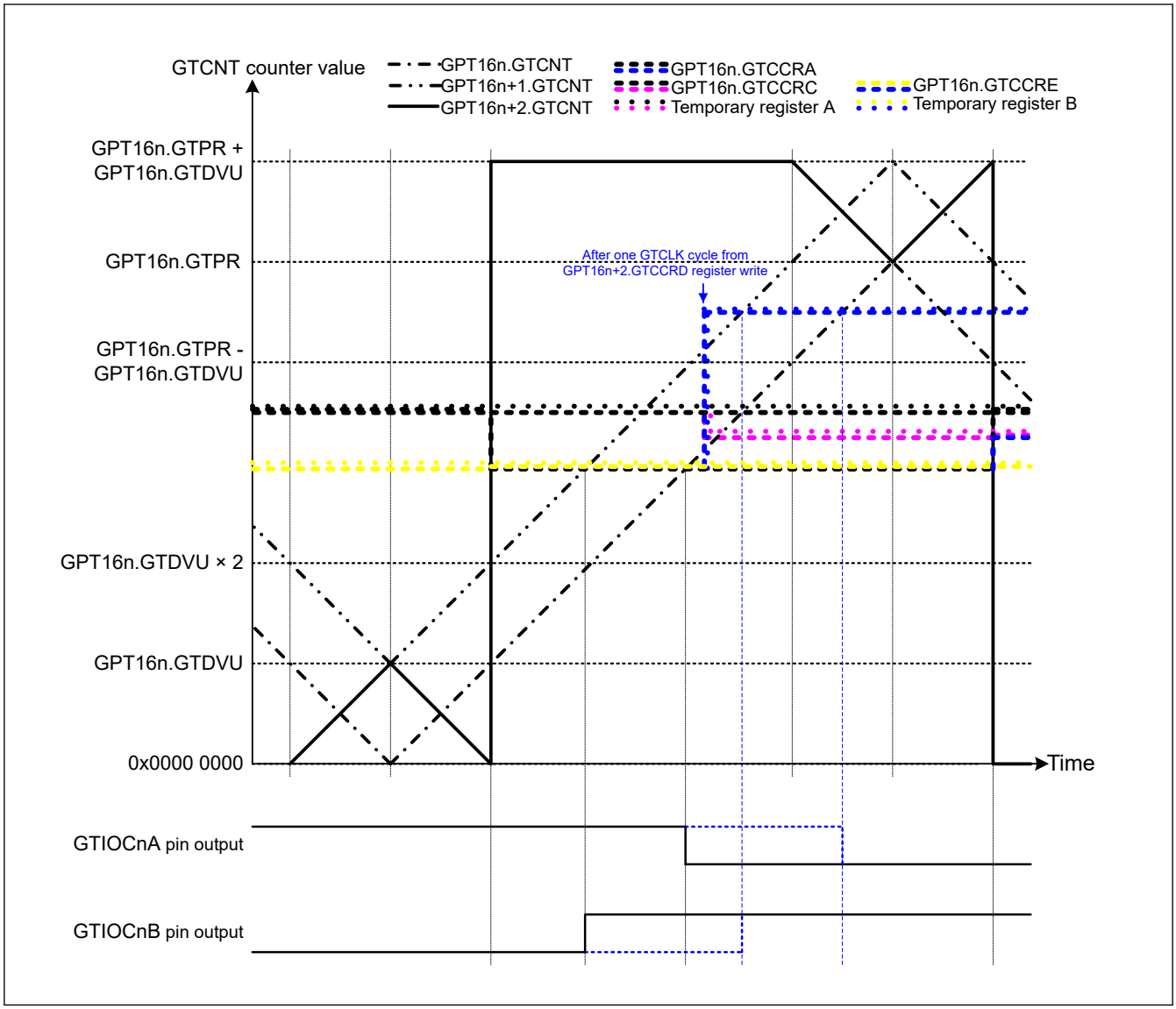


Figure 20.76 Example of complementary PWM mode 4 double buffer operation (up-counting middle section) (complementary PWM mode 4 double buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/ GTIOCnB pin = low at GTCCRA register compare match during down-counting, when a value is written to GTCCRF register after up-counting compare match) (n = 0)

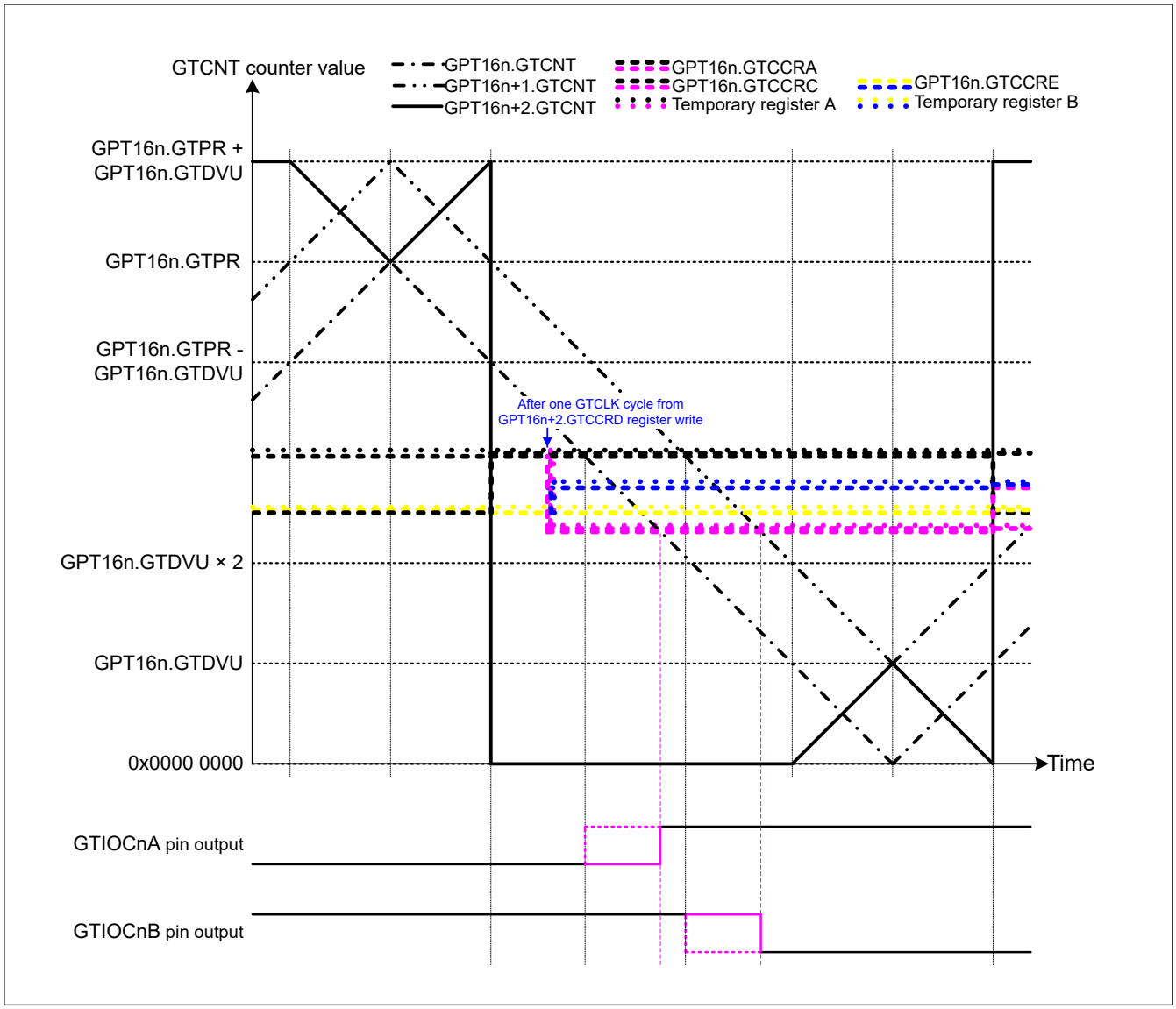


Figure 20.77 Example of complementary PWM mode 4 double buffer operation (down-counting middle section) (complementary PWM mode 4 double buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/ GTIOCnB pin = low at GTCCRA register compare match during down-counting, when a value smaller than the GPT16n.GTCNT value is written to GTCCRD register before down-counting compare match) (n = 0)

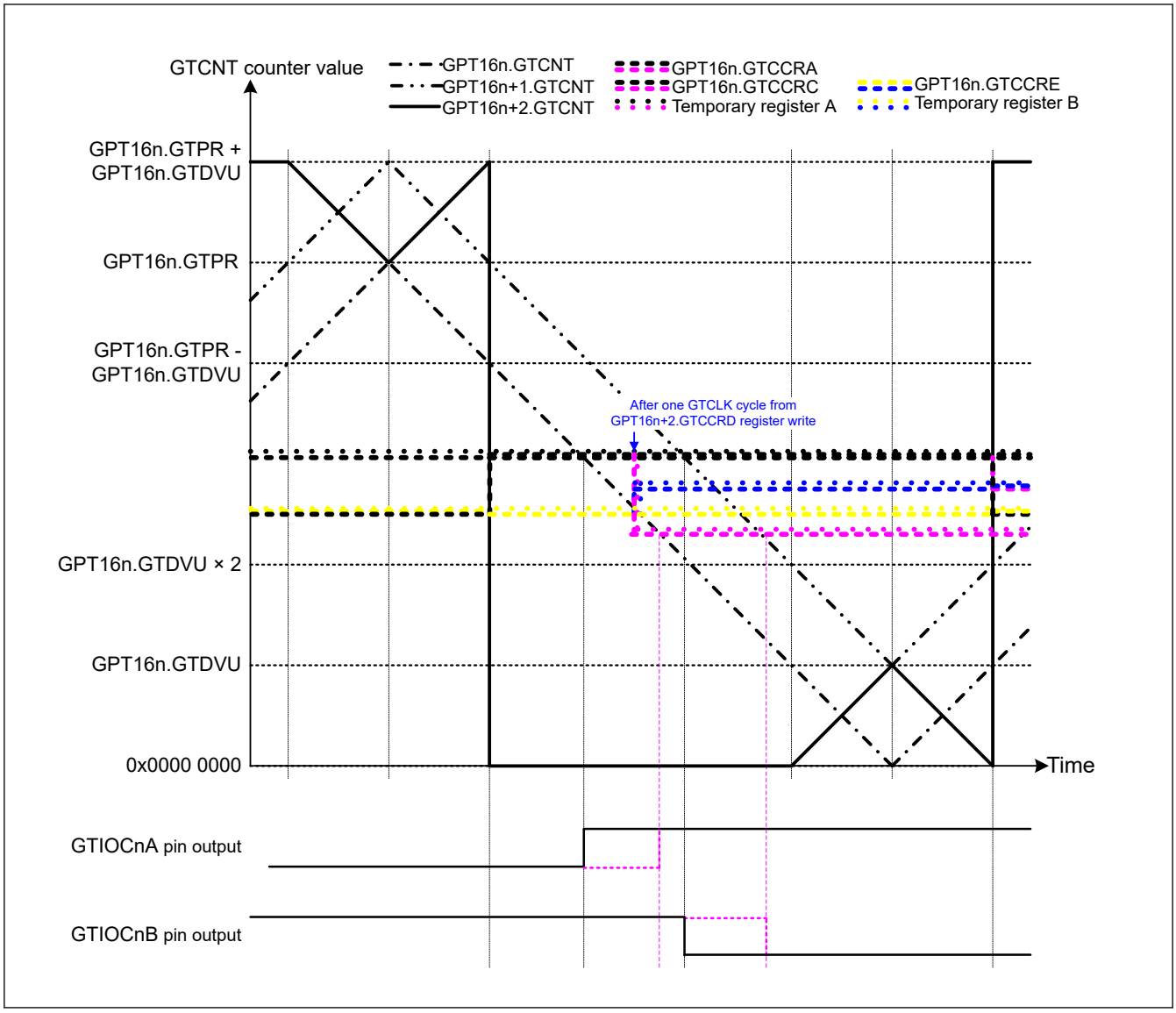


Figure 20.78 Example of complementary PWM mode 4 double buffer operation (down-counting middle section) (complementary PWM mode 4 double buffer operation, GTIOCnA pin = low/GTIOCnB pin = high at GTCCRA register compare match during up-counting, GTIOCnA pin = high/ GTIOCnB pin = low at GTCCRA register compare match during down-counting, when a value is written to GTCCRD register during the down-counting dead time) (n = 0)

Table 20.40 Example for setting complementary PWM mode 4 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode (1111b) with GTCR.MD[3:0] of GPT16n channel.
2	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits of GPT16n channel.
3	Set cycle	Set the cycle in GTPR, GTPBR.GTPDBR of GPT16n channel.
4	Set GTIOCnm /GTIOCn+1m / GTIOCn+2m pin function	Set the function of the GTIOCnm, GTIOCn+1m, and GTIOCn+2m pins with GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register of the GPT16n, GPT16n+1, and GPT16n+2 channels.
5	Enable GTIOCnm / GTIOCn+1m /GTIOCn+2m pin output	Set to enable the output from the GTIOCnm, GTIOCn+1m, and GTIOCn+2m pins with the OAE and OBE bits in the GTIOR register of the GPT16n, GPT16n+1, and GPT16n+2 channels.
6	Enable GTCPPOn pin output	Set to enable or disable PWM synchronous output from the GTCPPOn pin with the PSYE bit in the GTIOR register of GPT16n channel.
7	Set buffer operation	Set buffer operation with the GTBER2.CP3DB bit of the GPT16n, GPT16n+1, and GPT16n+2 channels.

Table 20.40 Example for setting complementary PWM mode 4 (2 of 2)

No.	Step Name	Description
8	Set compare match value	Set the output pin changing point during up-counting after count start in the GTCCRA register of the GPT16n, GPT16n+1, and GPT16n+2 channels.
9	Set buffer value	For single buffer operation, set data to be transferred for the first buffer transfer to the GTCCRA register after count start in the GTCCRD register. For double buffer operation, set data to be transferred at the first end of crest in the GTCCRD register and set data to be transferred at the first end of trough in the GTCCRF register.
10	Set dead time value	Set the dead time value in GTDVU of GPT16n channel.
11	Start count operation	Set GTCR.CST of GPT16n channel to 1 to start count operation.
12	Set compare match value to be transferred immediately	For single buffer operation, set data to be transferred immediately to the GTCCRA register in the GTCCRD register. For double buffer operation, set the compare match value for down-counting to be transferred immediately in the GTCCRD register and the compare match value for up-counting in the GTCCRF register. Finally, make settings for the GPT16n+2.GTCCRD register (data is transferred to the temporary register).

Note: n = 0
m = A, B

20.3.4 Automatic Dead Time Setting Function

By setting GTDTCR, a compare match value for a negative waveform with dead time obtained by a compare match value for a positive waveform (GTCCRA value) and specified dead time value (GTDVU and GTDVD value) can automatically be set to GTCCRB.

The automatic dead time setting function can be used in saw-wave one-shot pulse mode and all the triangle PWM modes.

Dead time for the changing point of a negative waveform is set in the GTDVU register in GPT163. Dead time can be separately set for the first half and second half of a waveform in GPT160 to GPT162. Dead time for the changing point in the first half of a negative waveform is set in the GTDVU register and that in the second half is set in GTDVD register. The same dead time can also be set for the first and second halves by setting the GTDTCR.TDFER bit to 1.

The GTDBU register can be used as a buffer register of the GTDVU register, and the GTDBD register can be used as a buffer register of the GTDVD registers.

Buffer transfer is performed at the end of the cycle. In triangle-wave mode, it is performed at the end of the cycle of a trough. In saw-wave mode, it is performed at the end of the cycle for any of the following conditions:

- An overflow of the GTCNT counter (up-counting)
- An underflow of the GTCNT counter (down-counting)
- A GTCNT counter clearing.

The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. Writing to GTCCRB is prohibited when the automatic dead time setting function is used.

Do not set the dead time that causes the change point of the waveform to exceed the count period. When any dead-time setting which would generate a dead-time error is made, adjust the change points of the positive- and negative-phase waveforms to generate waveforms with secured dead-time as shown in Table 20.41. The adjusted change point of the negative-phase waveform is automatically set in the GTCCRB register. An internal signal is used to determine the change point of the positive-phase waveform, thus the value of the GTCCRA register is not updated by the adjusted value.

In saw-wave one-shot pulse mode, if the order of the change point becomes inconsistent by adjustment of the waveform change point due to occurrence of dead time errors, or if the change point exceeds the count period even after the adjustment, the complementary relation between the positive- and negative-phases cannot be guaranteed.

In triangle-wave PWM mode, if dead time exceeds the count period by setting 0x00000000 or a value greater than or equal to the setting value of the GTPR register is set in the GTCCRA register, output change is controlled by the output protection function (see section 20.8.4. Output Protection Function for GTIOCnm Pin Output (n = 0 to 2, m = A, B)). When the GTCCRA register is greater than or equal to [GTPR register + GTDVm (m = U, D) register], [GTPR register - 1] is set in the GTCCRB register as the upper limit in GPT160 to GPT162.

Automatic setting for a dead time value to the GTCCRB register is performed at the next count clock after the register value for calculating the automatic setting value is updated. In triangle-wave mode, it also can be done at the next count clock from the current crest.

Table 20.41 Adjustment of the waveform change point when a dead-time error occurs in GPT160 to GPT162

PWM output operating mode	Count direction	Period	Condition for dead-time error	Change point of the positive-phase waveform after adjustment	Change point of the negative-phase waveform after adjustment
Saw-wave one-shot pulse mode	Up-counting	First half	GTCCRA - GTDVU < 0	GTDVU	0
		Second half	GTCCRA + GTDVD > GTPR (GTCCRA + GTDVU > GTPR)*1	GTPR - GTDVD (GTPR - GTDVU)*1	GTPR
	Down-counting	First half	GTCCRA + GTDVU > GTPR	GTPR - GTDVU	GTPR
		Second half	GTCCRA - GTDVD < 0 (GTCCRA - GTDVU < 0)*1	GTDVD (GTDVU)*1	0
Triangle-wave PWM mode 1/2/3	Up-counting	First half	GTCCRA - GTDVU ≤ 0	GTDVU + 1	1
	Down-counting	Second half	GTCCRA - GTDVD < 0 (GTCCRA - GTDVU < 0)*1	GTDVD (GTDVU)*1	0

Note 1. When GTDTCR.TDFER = 1.

Table 20.42 Adjustment of the waveform change point when a dead-time error occurs in GPT163

PWM output operating mode	Count direction	Period	Condition for dead-time error	Change point of the positive-phase waveform after adjustment	Change point of the negative-phase waveform after adjustment
Saw-wave one-shot pulse mode	Up-counting	First half	GTCCRA - GTDVU < 0	GTDVU	0
		Second half	GTCCRA + GTDVD > GTPR	GTPR - GTDVU	GTPR
	Down-counting	First half	GTCCRA + GTDVU > GTPR	GTPR - GTDVU	GTPR
		Second half	GTCCRA - GTDVD < 0	GTDVU	0
Triangle-wave PWM mode 1/2/3	Up-counting	First half	GTCCRA - GTDVU ≤ 0	GTDVU + 1	1
	Down-counting	Second half	GTCCRA - GTDVD < 0	GTDVU	0

Figure 20.79 to Figure 20.85 show examples of automatic dead time setting function operation. Table 20.43 and Table 20.44 show the setting examples.

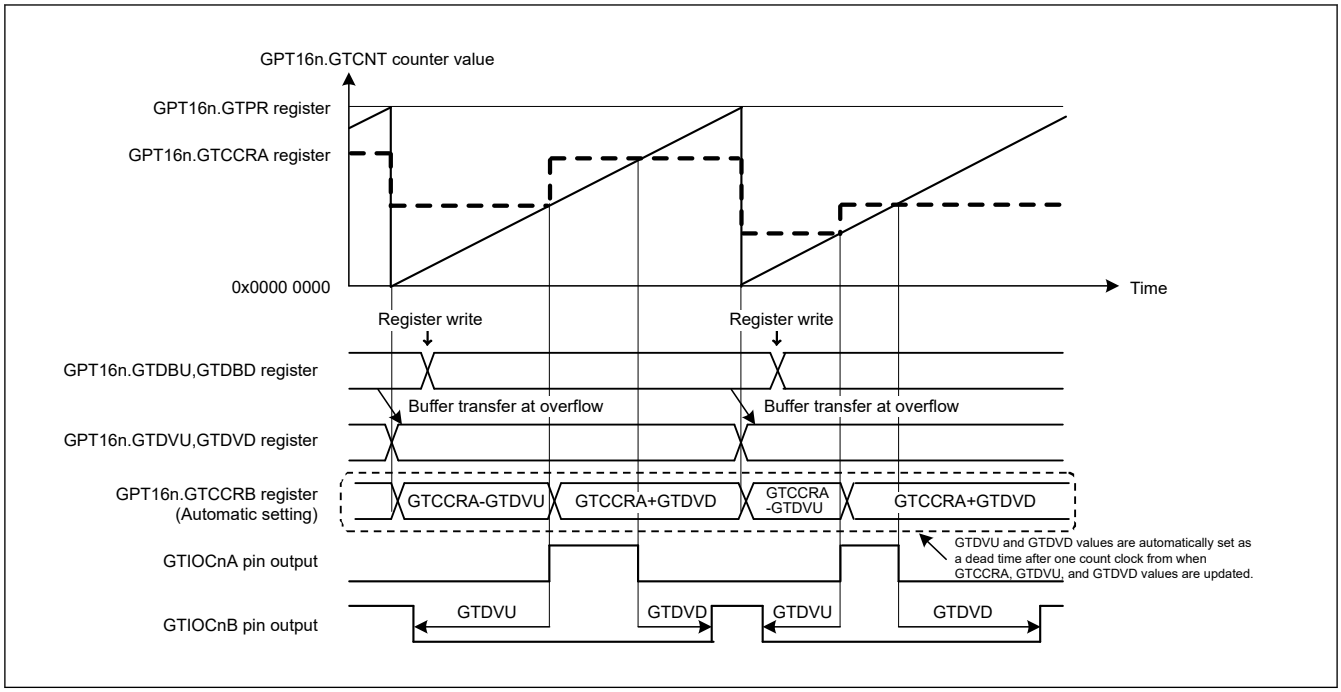


Figure 20.79 Example of automatic dead time setting function operation in saw-wave one-shot pulse mode, up-counting, GTDVU and GTDVD set to buffer operation, and active-high in GPT160 to GPT162

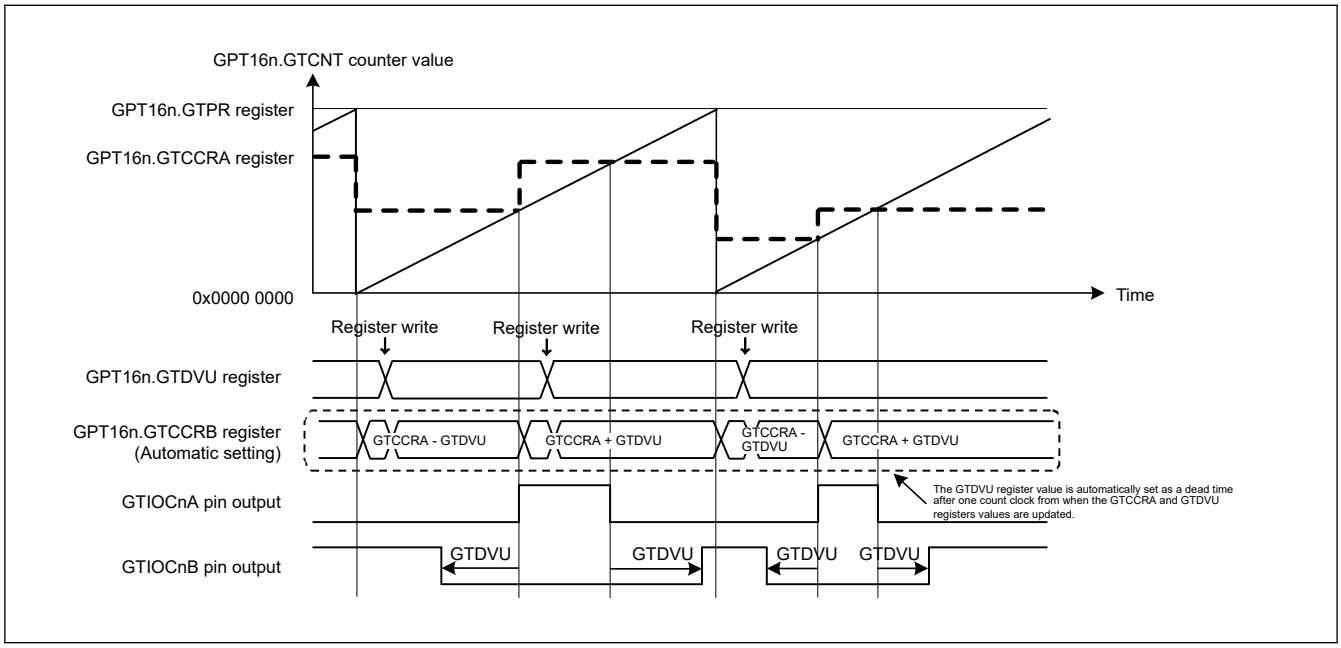


Figure 20.80 Example of automatic dead time setting function operation in saw-wave one-shot pulse mode, up-counting and active-high in GPT163

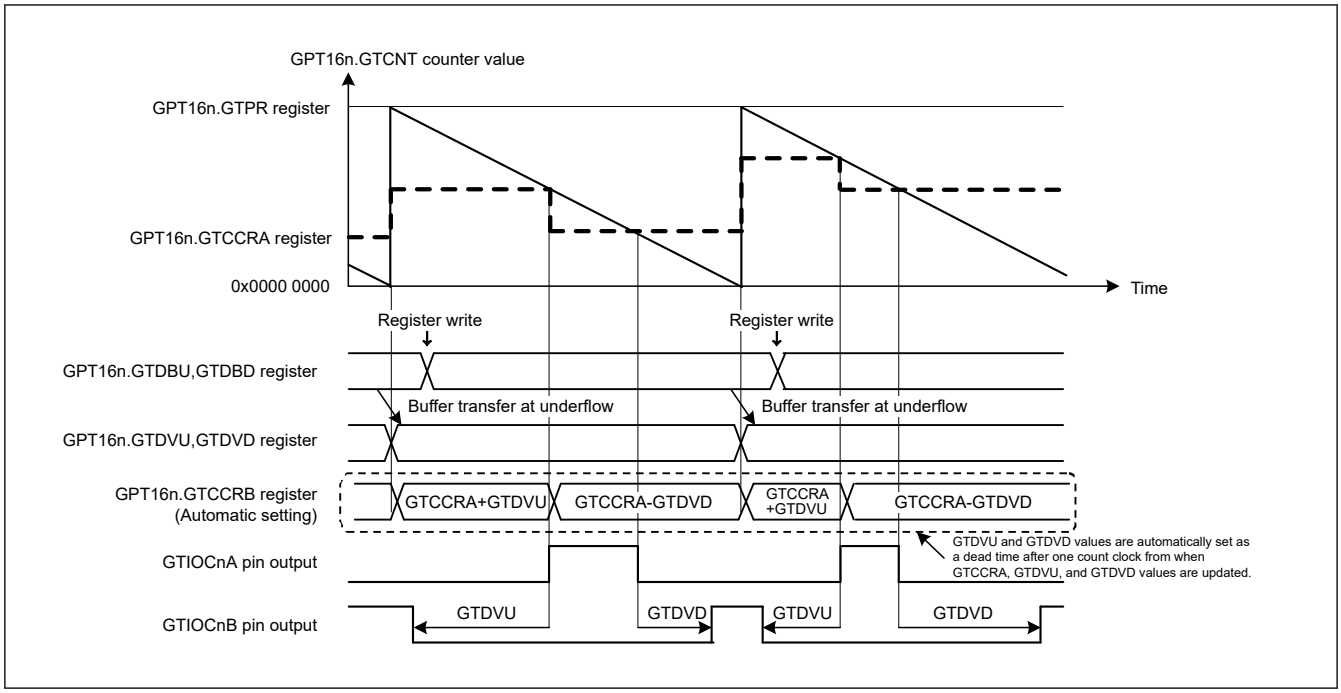


Figure 20.81 Example of automatic dead time setting function operation in saw-wave one-shot pulse mode, down-counting, GTDVU and GTDVD set to buffer operation, and active-high in GPT160 to GPT162

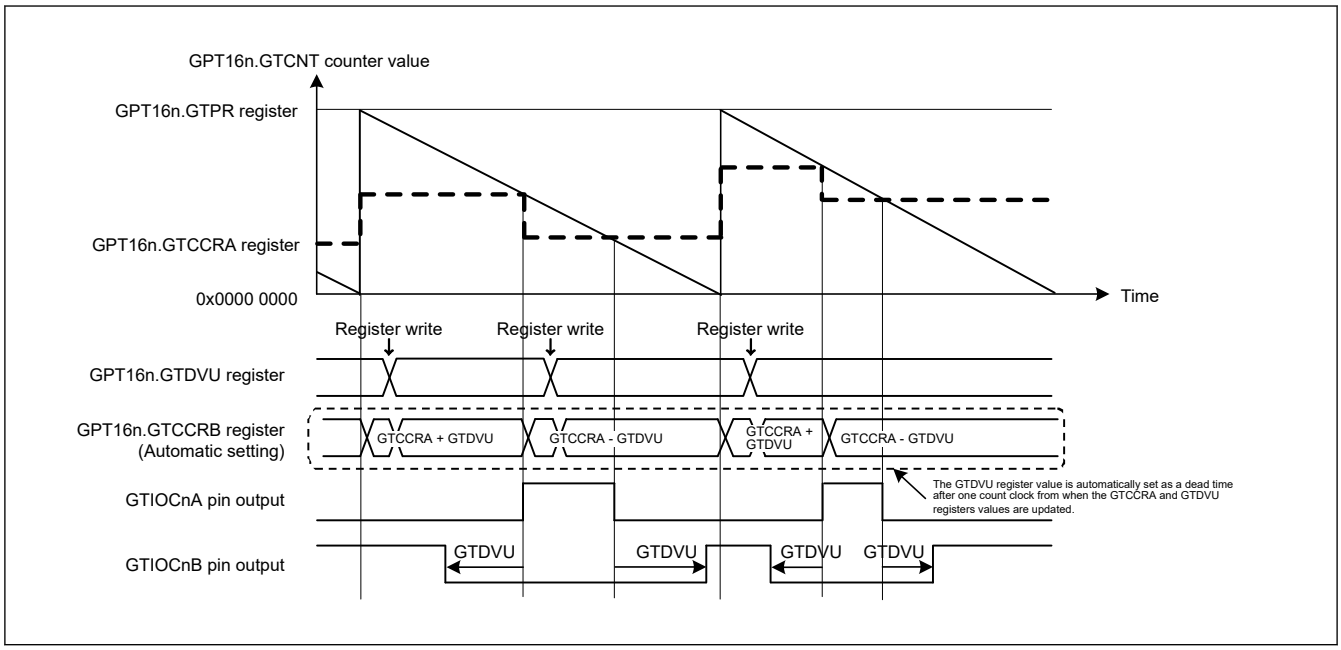


Figure 20.82 Example of automatic dead time setting function operation in saw-wave one-shot pulse mode, down-counting and active-high in GPT163

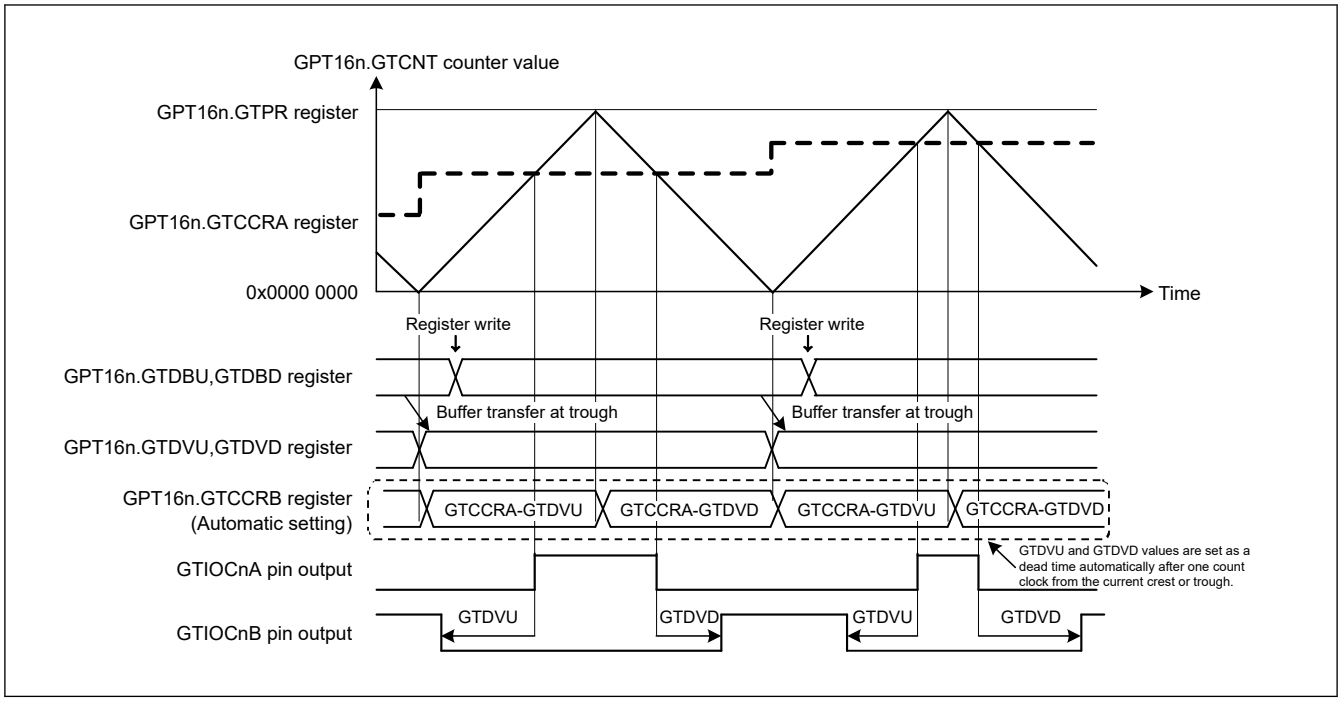


Figure 20.83 Example of automatic compare-match value setting function with dead time in triangle-wave PWM mode 1, GTDVU and GTDVD set to buffer operation, and active-high in GPT160 to GPT162

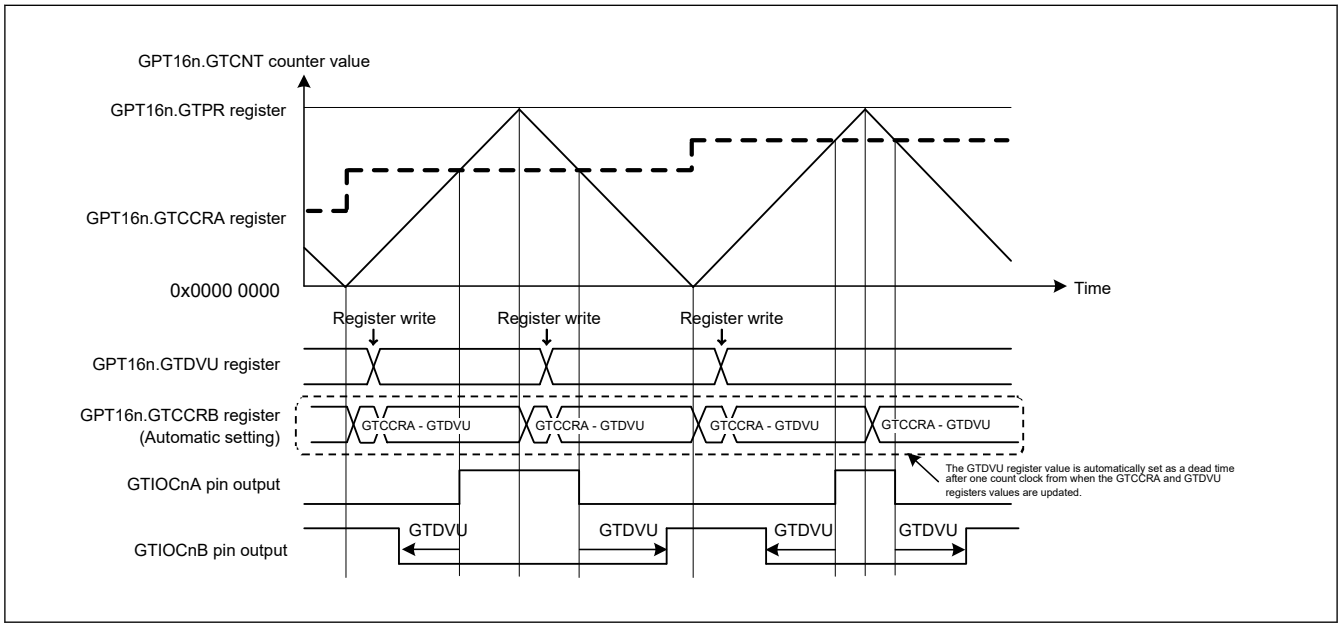


Figure 20.84 Example of automatic compare-match value setting function with dead time in triangle-wave PWM mode 1 and active-high in GPT163

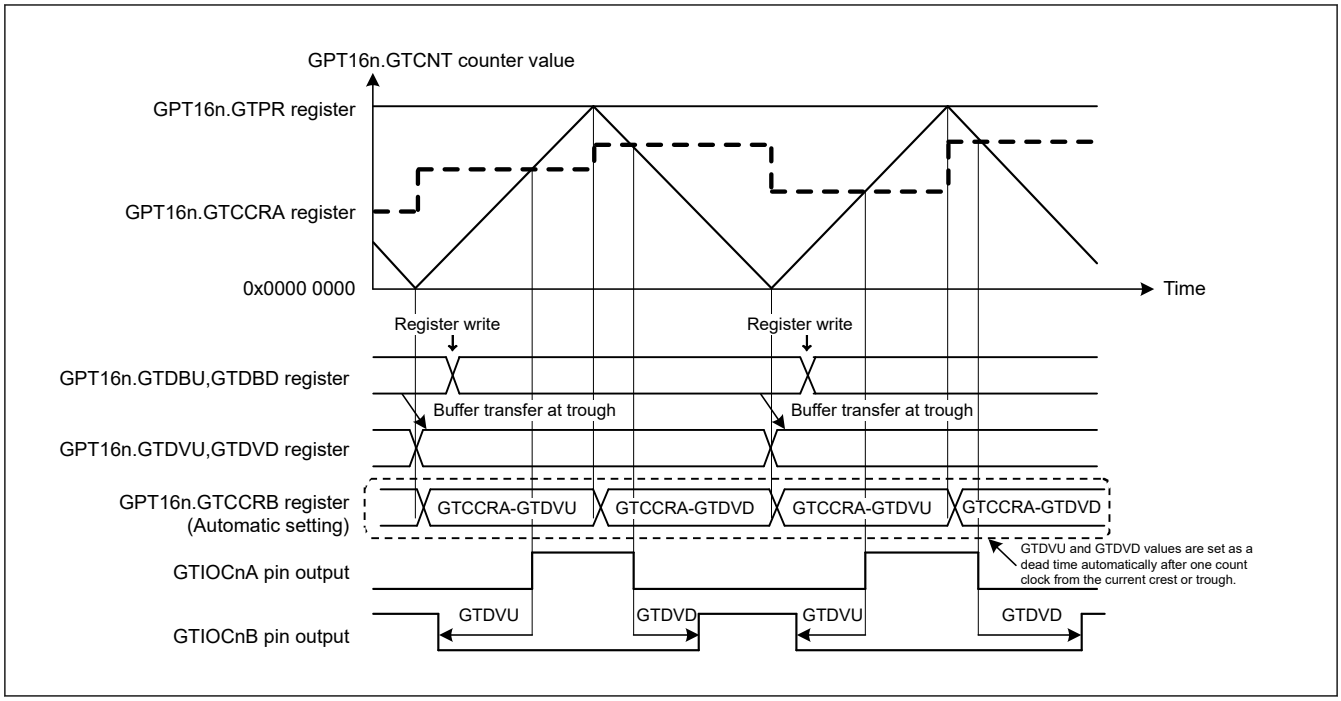


Figure 20.85 Example of automatic compare-match value setting function with dead time in triangle-wave PWM mode 2 or 3, GTDVU and GTDVD set to buffer operation, and active-high in GPT160 to GPT162

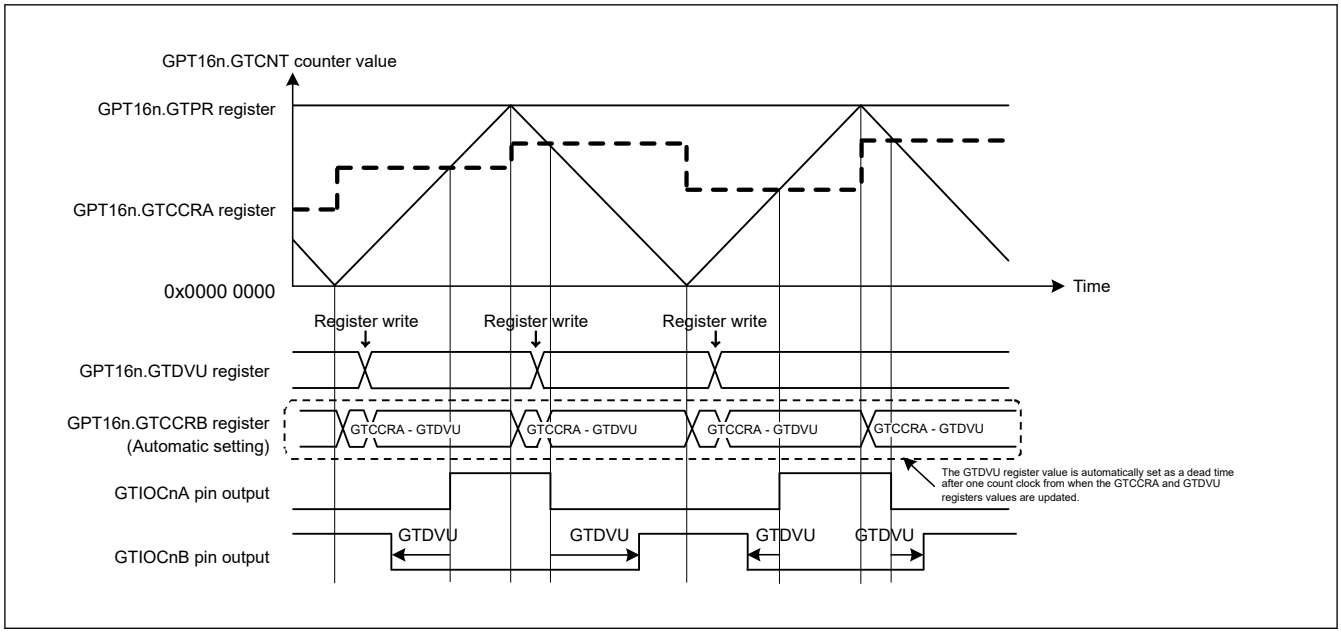


Figure 20.86 Example of automatic compare-match value setting function with dead time in triangle-wave PWM mode 2 or 3 and active-high in GPT163

Table 20.43 Example setting for automatic dead time setting function in saw-wave one-shot pulse mode, and triangle-wave PWM mode 3 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0] or GTCR.MD[3:0]. In Figure 20.79 and Figure 20.81, 001b or 0001b (saw-wave one-shot pulse mode) is set. In Figure 20.85, 110b or 0110b (triangle-wave PWM mode 3) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.79, 01b is set after 11b is set in GTUDDTYC.UDF and UD (up count). In Figure 20.81, 00b is set after 10b is set in GTUDDTYC.UDF and UD (down count).

Table 20.43 Example setting for automatic dead time setting function in saw-wave one-shot pulse mode, and triangle-wave PWM mode 3 (2 of 2)

No.	Step Name	Description
3	Select count clock	Select the count clock with GTCR.TPCS[3:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 20.79, Figure 20.81, and Figure 20.85, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
7	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with OAE and OBE in GTIOR.
8	Set buffer value for compare match*1	Set the GTIOCnA pin transition immediately after the count start in GTCCRC and GTCCRD.
9	Set forcible buffer transfer for compare match*1	Set GTBER.CCRSWT to 1 to transfer buffer register data forcibly to GTCCRA.
10	Set buffer value for compare match	Set the GTIOCnA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD.
11	Set automatic dead time setting function	Set GTDTCR.TDE to 1 to enable the automatic dead time setting function.
12	Set buffer operation for dead time setting	Set buffer operation with TDBUE and TDBDE bits in GTDTCR in GPT160 to GPT162.
13	Set dead time value	Set the dead time value in the GTDVU register in GPT163. Set the first half dead time value in GTDVU and the second half dead time in GTDVD in GPT160 to GPT162. When GTDVU is set with GTDTCR.TDFER bit set to 1, the same value is also set to GTDVD, the same dead time value can be set for the first and second halves.
14	Set buffer value for dead time	For buffer operation, set the first half dead time in one cycle after the current cycle in GTDBU and the second half dead time in GTDBD in GPT160 to GPT162.
15	Start count operation	Set GTCR.CST to 1 to start count operation.
16	Set buffer value for each cycle	Set the GTIOCnA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD. When the dead time register is used for buffer operation, set the dead time value in the first half of the next cycle from the current cycle to GTDBU and the dead time value in the second half to GTDBD in GPT160 to GPT162.

Note: n: 0 to 3
m: A, B

Note 1. When using PWM Delay Generation circuit, change the order of setting for Enable GTIOCnm pin output and setting for Set buffer value for compare match + Set forcible buffer transfer for compare match.

Table 20.44 Example setting for automatic dead time setting function in triangle-wave PWM mode 1 or 2 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0] or GTCR.MD[3:0]. In Figure 20.83, 100b or 0100b (triangle-wave PWM mode 1) is set. In Figure 20.85, 101b or 0101b (triangle-wave PWM mode 2) is set.
2	Select count clock	Select the count clock with GTCR.TPCS[3:0].
3	Set cycle	Set the cycle in GTPR.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with GTIOA[4:0] and GTIOB[4:0] in GTIOR. In Figure 20.83 and Figure 20.85, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTIOCnm pin output*1	Set to enable the GTIOCnm pin output with OAE and OBE in GTIOR.
7	Set buffer operation for compare match	Set buffer operation with CCRA in GTBER.
8	Set compare match value*1	Set the GTIOCnA pin transition in GTCCRA.

Table 20.44 Example setting for automatic dead time setting function in triangle-wave PWM mode 1 or 2 (2 of 2)

No.	Step Name	Description
9	Set buffer value for compare match	For buffer operation, set the GTIOCnA pin transition in 1 cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRC. For double buffer operation, also set the GTIOCnA pin transition in 2 cycles after the current cycle (in triangle-wave PWM mode 1) or 1 cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRD.
10	Set automatic dead time setting function	Set GTDTCR.TDE to 1 to enable the automatic dead time setting function.
11	Set buffer operation for dead time setting	Set buffer operation with TDBUE and TDBDE bits in GTDTCR in GPT160 to GPT162.
12	Set dead time value	Set the dead time value in the GTDVU register in GPT163. Set the first half dead time value in GTDVU and the second half dead time in GTDVD in GPT160 to GPT162. When GTDVU is set with GTDTCR.TDFER bit set to 1, the same value is also set to GTDVD, the same dead time value can be set for the first and second halves.
13	Set buffer value for dead time	For buffer operation, set the first half dead time in one cycle after the current cycle in GTDBU and the second half dead time in GTDBD in GPT160 to GPT162.
14	Start count operation	Set GTCR.CST to 1 to start count operation.
15	Set buffer value for each cycle	When the compare match register is used for buffer operation, set the GTIOCnA pin transition in 1 cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRC. When the compare match register is used for double-buffered operation, set the GTIOCnA pin changing point in two cycles after the current cycle (in triangle-wave PWM mode 1) or one cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRD. In the same way, set the dead time value in the first half of the cycle after current cycle in GTDBU and the dead time in the second half in GTDBD in GPT160 to GPT162.

Note: n: 0 to 3
m: A, B

Note 1. When using PWM Delay Generation circuit, change the order of setting for the GTIOCnm pin output enable and setting for a compare match value.

20.3.5 Count Direction Changing Function

The count direction of the GTCNT counter can be changed by modifying the UD bit in GTUDDTYC.

In saw-wave mode, if the UD bit in GTUDDTYC is modified during count operation, the count direction is changed at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the GTUDDTYC.UD bit is modified while the count operation stops and the GTUDDTYC.UDF bit is 0, the GTUDDTYC.UD bit modification is not reflected at the start of counting and the count direction is changed at an overflow or an underflow. If the UDF bit is set to 1 while the count operation stops, the GTUDDTYC.UD bit value at that time is reflected at the start of counting.

In triangle-wave mode, the count direction does not change even though the UD bit in GTUDDTYC is modified during the count operation. Similarly, even though the GTUDDTYC.UD bit is modified while the count operation stops and GTUDDTYC.UDF bit is 0, the GTUDDTYC.UD bit value is not reflected to the count operation. If the GTUDDTYC.UDF bit is set to 1 while the count operation is stopped, the GTUDDTYC.UD bit value at that time is reflected at the start of counting.

If the count direction changes during a saw-wave count operation, the GTPR value after the start of up-counting is reflected in the count cycle during up-counting and the GTPR value after the start of down-counting is reflected in the count cycle during down-counting.

Figure 20.87 shows an example of count direction changing function operation.

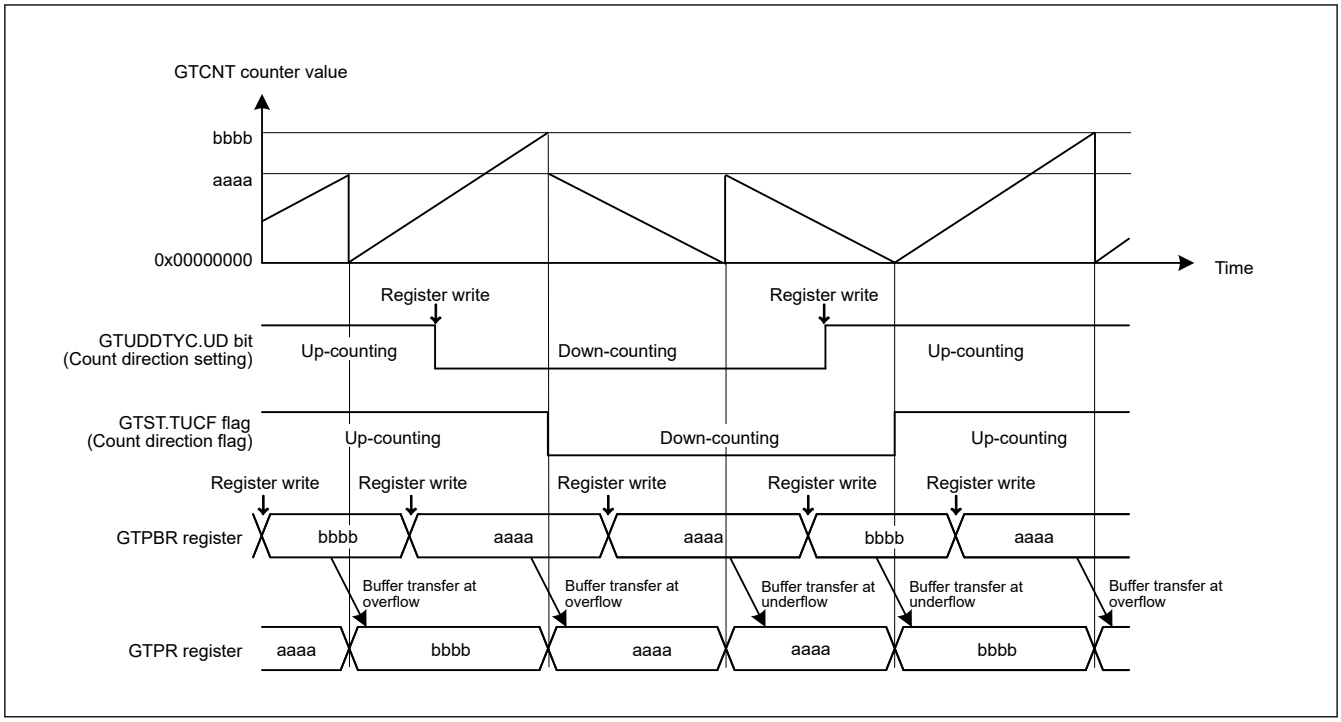


Figure 20.87 Example of a count direction changing function operation during buffer operation

20.3.6 Function of Output Duty 0% and 100%

The output duty of the GTIOCnA pin and the GTIOCnB pin (n = 0 to 3) are set to 0% or 100% by changing the GTUDDTYC.OADTY bit or GTUDDTYC.OBDTY bit.

This function is invalid in saw-wave PWM mode 2 or complementary PWM mode.

In saw-wave mode, if the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified during the count operation, the output duty setting is reflected at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 0, the output duty modification is not reflected at the start of counting. The output duty changes at an overflow or an underflow. If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 1, the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit value at that time is reflected at the start of counting.

In triangle-wave mode, if the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified during the count operation. When the GTUDDTYC.OABDTYT bit is 0, the output duty setting changed when an underflow occurs is reflected, and when the GTUDDTYC.OABDTYT bit is 1, the output duty setting changed when an underflow occurs and at the top of the triangle-wave is reflected.

If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 0, the output duty modification is not reflected at the start of counting. The output duty changes at an underflow. If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation stops and the GTUDDTYC.OADTYF or the GTUDDTYC.OBDTYF bit is 1, the output duty modification is reflected at the start of counting.

In performing 0% or 100% duty operation, GPT internally continues to:

- Perform compare match operation
- Set compare match flag
- Output interrupt
- Perform buffer operation.

When the control is changed from 0% or 100% duty setting to compare match, the output value of GTIOCnA pin at cycle end is decided by GTIOR.GTIOA[3:2] and GTUDDTYC.OADTYR. The output value of GTIOCnB pin at cycle end is decided by GTIOR.GTIOB[3:2] and GTUDDTYC.OBDTYR.

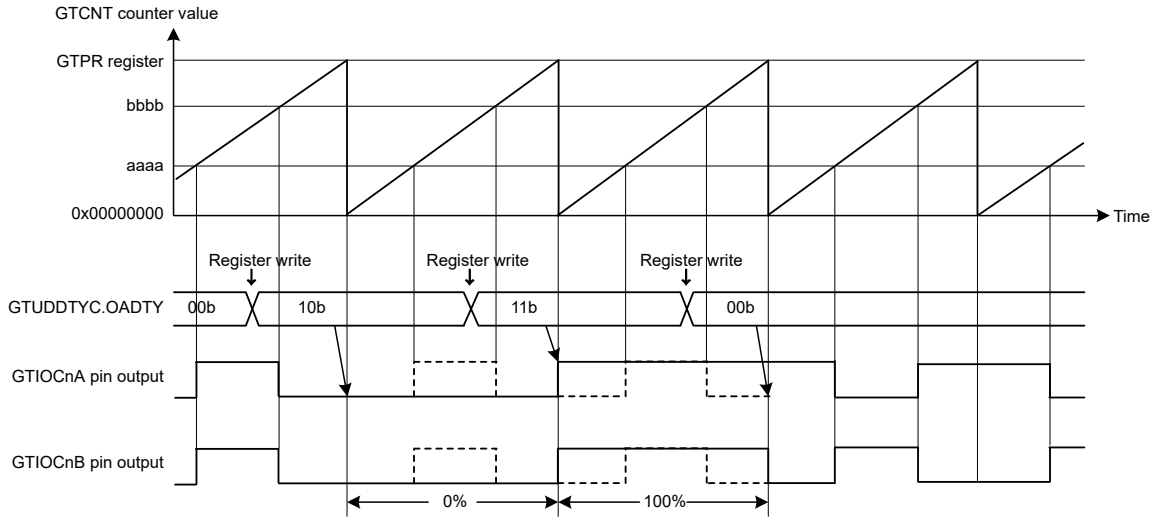
When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 01b, the output pins output low at cycle end. When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 10b, the output pins output high at cycle end.

GTUDDTYC.OADTYR selects the value that is the object of output retained/toggled at cycle end, when GTIOR.GTIOm[3:2] are set to 00b (output retained at cycle end) or when GTIOR.GTIOm[3:2] are set to 11b (output toggled at cycle end). Table 20.45 shows the values of GTIOCnA and GTIOCnB pin output at cycle end.

Table 20.45 Output values after releasing 0% or 100% duty setting (m = A, B)

GTIOR.GTIOm[3:2]	Compare match value at cycle end masked by 0% or 100% duty setting	GTUDDTYC.OmDTYR in duty 0% setting		GTUDDTYC.OmDTYR in duty 100% setting	
		0	1	0	1
00 (output retained at cycle end)	0	0	0	1	0
	1	0	1	1	1
01 (low output at cycle end)	—	0	0	0	0
10 (high output at cycle end)	—	1	1	1	1
11 (output toggled at cycle end)	0	1	1	0	1
	1	1	0	0	0

Figure 20.88 shows an example of output duty 0% and 100% function.



Note: The base address of registers are the same.

Note: [Setting examples]

GTIOR.GTIOA[4:0] bits:	00011b initial low output, output toggled at compare match, output retained at cycle end
GTUDDTYC.OADTYR bit:	0b The function selected by GTIOA[3:2] is applied to the output value when the duty cycle is set after release from 0 or 100% duty-cycle setting.
GTIOR.GTIOB[4:0] bits:	00011b initial low output, output toggled at compare match, output retained at cycle end
GTUDDTYC.OBDTYR bit:	1 The function of GTIOB[3:2] is applied to the compare match output value which is masked after 0 or 100% duty setting release.

Figure 20.88 Example of output duty 0% and 100% function

20.3.7 Hardware Count Start/Count Stop and Clear Operation

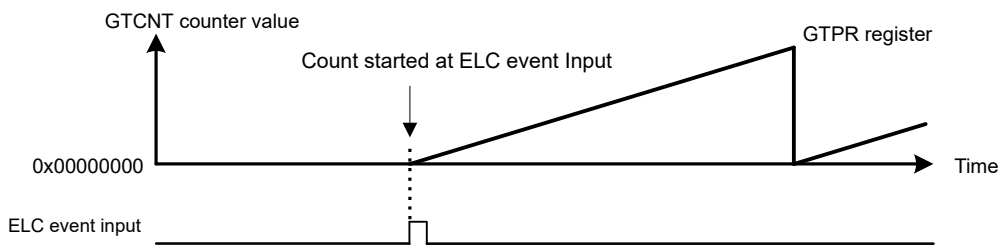
The GTCNT counter can be started, stopped, or cleared by the following hardware sources:

- External trigger input
- ELC event input
- GTIOCnA and GTIOCnB pin input (n = 0 to 3).

20.3.7.1 Hardware Start Operation

The GTCNT counter can be started by selecting a hardware source using GTSSR.

Figure 20.89 shows an example of a count start operation by a hardware source. Table 20.46 shows the setting example.



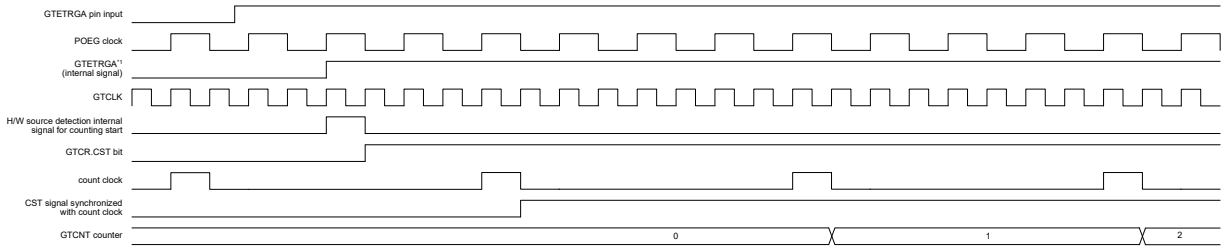
Note: ELC event input: ELC_GPTA event input

Figure 20.89 Example of count start operation by a hardware source started at the input of the signal from the ELC_GPTA event

Table 20.46 Example setting for count start operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.89, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.89, after 11b is set in the GTUDDTYC.UDF and UD bits, 01b is set in the GTUDDTYC.UDF and UD bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 20.89, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation in the GTSSR register. In Figure 20.89, GTSSR.SSELCA = 1
7	Set hardware source operation	Set operation of the hardware source selected by the GTSSR register and start counting. In Figure 20.89, the ELC_GPTA event input operation is set.

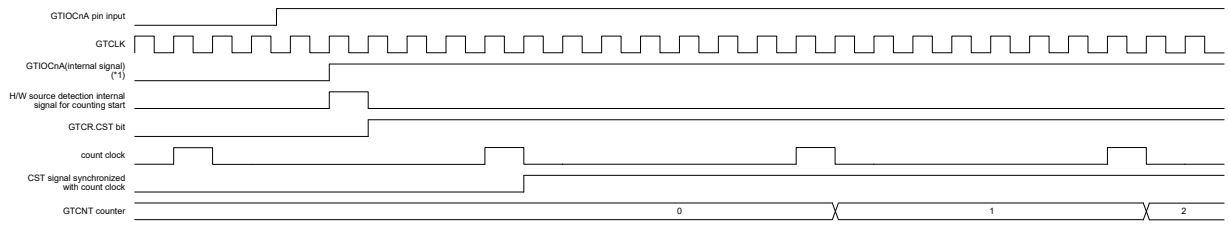
Figure 20.90 shows an example of timing of operations to start counting in response to a rising edge of the input on the GTETRGA pin.



Note: POEG clock: PCLKB
Note 1. The GTETRGA internal signal reflects the signal being input to GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 20.90 Example of timing of operations to start counting in response to a rising edge of the input on the GTETRGA pin

Figure 20.91 shows an example of timing of operations to start counting in response to a rising edge of the input on the GTIOCnA pin.



Note 1. This is the case when the digital noise filter is not in use.

Figure 20.91 Example of timing of operations to start counting in response to a rising edge of the input on the GTIOCnA pin

Figure 20.92 shows an example of the timing of operations to start counting in response to ELC_GPTA event input.

This is an example of operations to start counting by the GPT161.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPT160.GTCCRA register. This is selected as a trigger for output to the GPT161 by the ELC as ELC_GPTA.

The GPT160 compare match A signal is synchronized with GTCLK. The ELC receives the signal in synchronization with PCLKB, and outputs the GPT16 event source A signal after 1 cycle of PCLKB has elapsed.

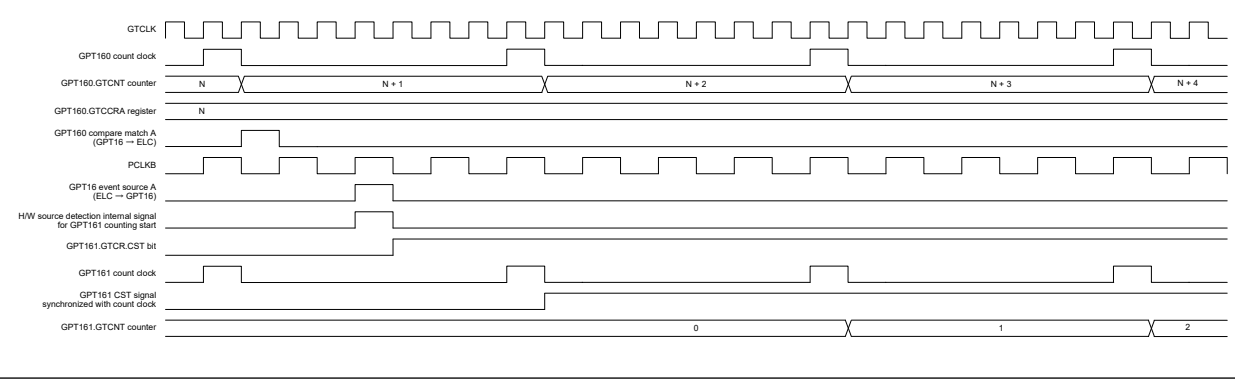
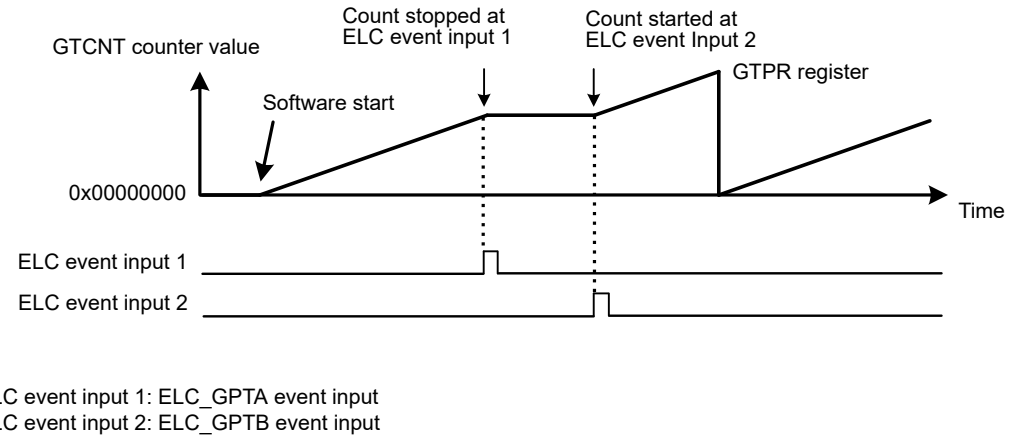


Figure 20.92 Example of timing of operations to start counting in response to event input from ELC_GPTA

20.3.7.2 Hardware Stop Operation

The GTCNT counter can be stopped by selecting a hardware source using GTPSR.

Figure 20.93 shows an example of a count stop operation by a hardware source. Table 20.47 shows the setting example. In this example, the count operation stops at the ELC_GPTA event input and restarts at the ELC_GPTB event input.



Note: ELC event input 1: ELC_GPTA event input
ELC event input 2: ELC_GPTB event input

Figure 20.93 Example of count stop operation by hardware source started by software, stopped at ELC_GPTA input, and restarted at ELC_GPTB input

Table 20.47 Example setting for count stop operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0] or GTCR.MD[3:0] bits. In Figure 20.93, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.93, after 11b is set in GTUDDTYC.UDF and UD, 01b is set in GTUDDTYC.UDF and UD (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[3:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 20.93, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation in GTSSR register, and wait for count start by the hardware source. In Figure 20.93, GTSSR.SSELCB = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation in GTPSR register and wait for count stop by the hardware source. In Figure 20.93, GTPSR.PSELCA = 1.
8	Set hardware source operation	Set operation of the hardware source selected in GTSSR register or GTPSR register, and start or stop counting. In Figure 20.93, ELC_GPTA input operation and ELC_GPTB input operation are set.

Figure 20.94 shows an example of a count start/stop operation by a hardware source. Table 20.48 shows the setting example. In this example, the counter operates during the high-level periods of the external trigger input GTETRGA.

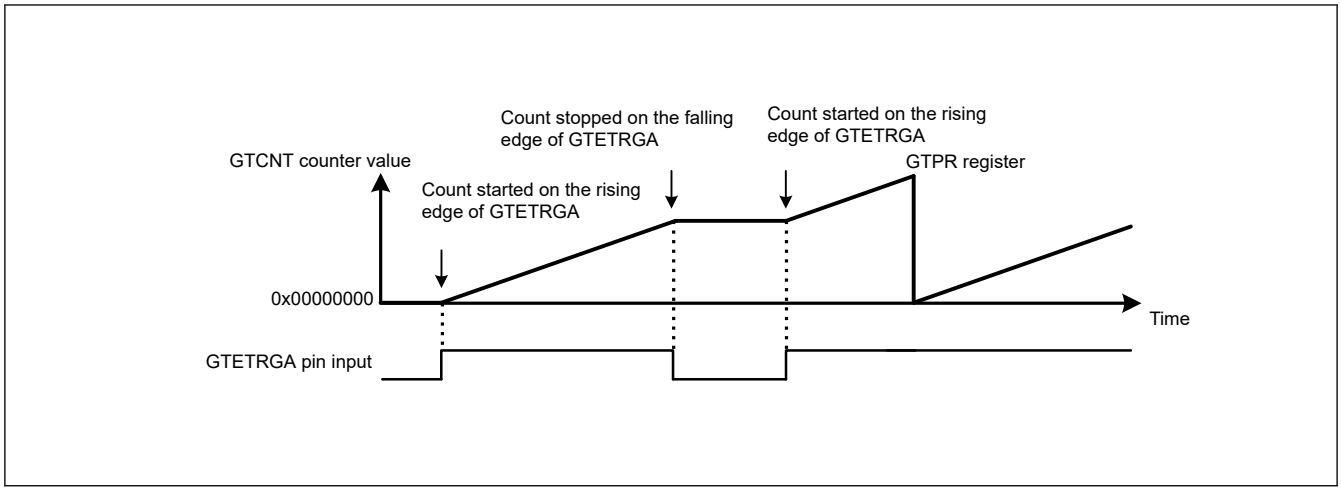
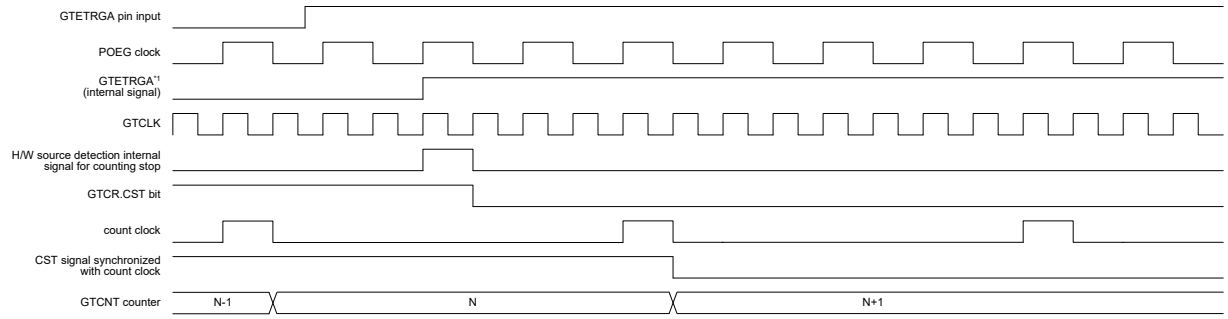


Figure 20.94 Example of count start/stop operation by a hardware source started on the rising edge of GTETRGA pin input, and stopped on the falling edge of GTETRGA pin input

Table 20.48 Example setting for count start/stop operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.94, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.94, after 11b is set in the GTUDDTYC.UDF and UD bits, 01b is set in the GTUDDTYC.UDF and UD bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 20.94, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation with the GTSSR register, and wait for count start by the hardware source. In Figure 20.94, GTSSR.SSGTRGAR = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation with the GTPSR register, and wait for count stop by the hardware source. In Figure 20.94, GTPSR.PSGTRGAF = 1.
8	Set hardware source operation	Set operation of the hardware source selected in the GTSSR register or GTPSR register and start or stop counting. In Figure 20.94, the GTETRGA pin operation is set.

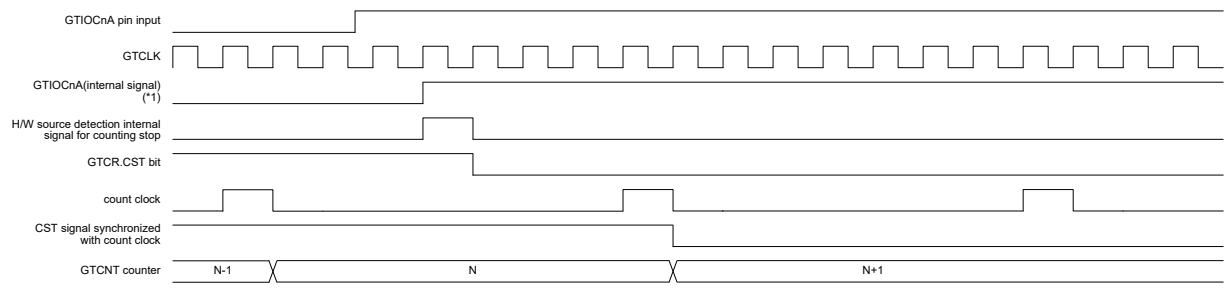
Figure 20.95 shows an example of timing of operations to stop counting in response to a rising edge of the input on the GTETRGA pin.



Note: POEG clock: PCLKB
Note 1. The GTETRGA internal signal reflects the signal being input to GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use

Figure 20.95 Example of timing of operations to stop counting in response to a rising edge of the input on the GTETRGA pin

Figure 20.96 shows an example of timing of operations to stop counting in response to a rising edge of the input on the GTIOCnA pin.



Note 1. This is the case when the digital noise filter is not in use.

Figure 20.96 Example of timing of operations to stop counting in response to a rising edge of the input on the GTIOCnA pin

Figure 20.97 shows an example of timing of operations to stop counting in response to event input from ELC_GPTA.

This is an example of operations to stop counting by the GPT161.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPT160.GTCCRA register. This is selected as a trigger for output to the GPT161 by the ELC as ELC_GPTA.

The GPT160 compare match A signal is synchronized with GTCLK. The ELC receives the signal in synchronization with PCLKB, and outputs the GPT16 event source A signal after 1 cycle of PCLKB has elapsed.

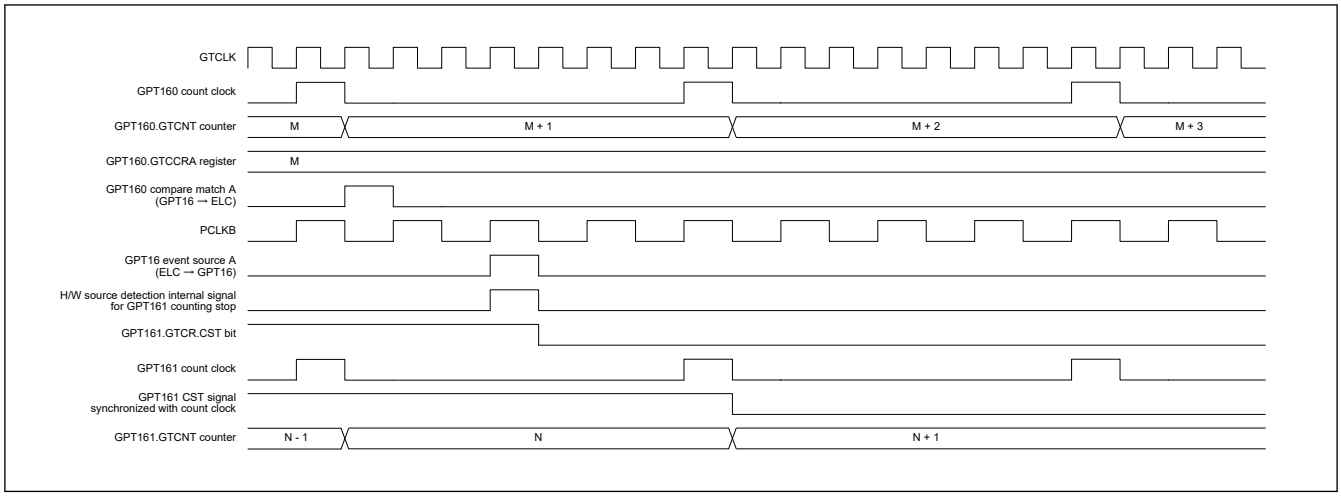


Figure 20.97 Example of timing of operations to stop counting in response to event input from ELC_GPTA

20.3.7.3 Hardware Clear Operation

The GTCNT counter can be cleared by selecting a hardware source using GTCSR. The GPTn_OVF/GPTn_UDF (n = 0 to 3) interrupt (overflow/underflow interrupt) is not generated when the GTCNT counter is cleared by a hardware source or by software.

Figure 20.98 and Figure 20.99 show examples of the GTCNT counter clearing operation by a hardware source. Table 20.49 shows the setting example. In this example, the GTCNT counter starts at the ELC_GPTA input, and the counter stops and clears at the ELC_GPTB input.

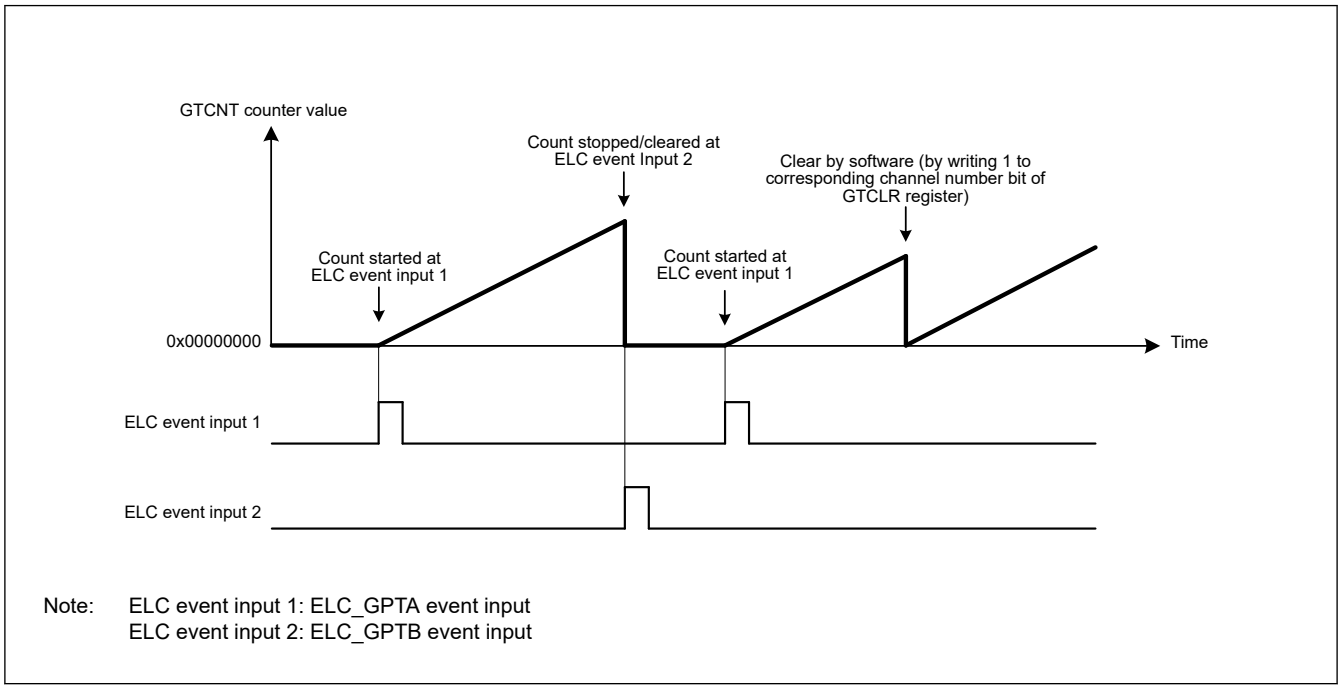
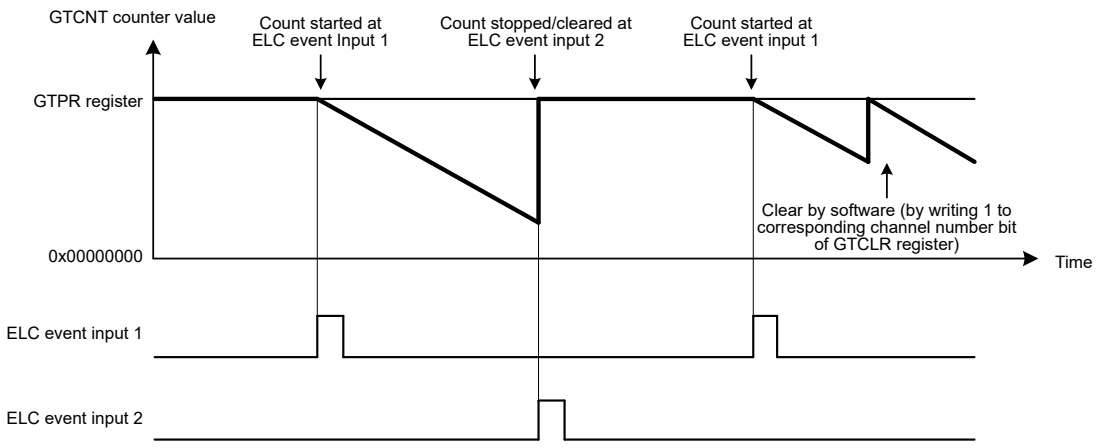


Figure 20.98 Examples of count clearing operation by hardware source in saw wave up-counting, started at ELC_GPTA input, and stopped/cleared at ELC_GPTB input



Note: ELC event input 1: ELC_GPTA event input
ELC event input 2: ELC_GPTB event input

Figure 20.99 Examples of count clearing operation by hardware source in saw wave down-counting, started at ELC_GPTA input, and stopped/cleared at ELC_GPTB input

Table 20.49 Example setting for count clearing operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.98 and Figure 20.99, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.98, after 11b is set in the GTUDDTYC.UDF and UD bits, 01b is set in the GTUDDTYC.UDF and UD bits (up-counting). In Figure 20.99, after 10b is set in the GTUDDTYC.UDF and UD bits, 00b is set in the GTUDDTYC.UDF and UD bits (down-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 20.98, 0x00000000 is set. In Figure 20.99, the GTPR register value is set.
6	Set hardware count start	Select a hardware source for starting count operation in the GTSSR register, and wait for count start by the hardware source. In Figure 20.98 and Figure 20.99, GTSSR.SSELCA = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation in the GTPSR register, and wait for count stop by the hardware source. In Figure 20.98 and Figure 20.99, GTPSR.PSELCB = 1.
8	Set hardware count clear	Select a hardware source for clearing count operation in the GTCSR register, and wait for count clear by the hardware source. In Figure 20.98 and Figure 20.99, GTCSR.CSELCB = 1.
9	Set hardware source operation	Set operation of the hardware source selected in the GTSSR register, GTPSR register or GTCSR register and start, stop or clear counting. In Figure 20.98 and Figure 20.99, the ELC_GPTA input and ELC_GPTB input are set.

The GPTn_OVF/GPTn_UDF (n = 0 to 3) interrupt (overflow/underflow interrupt) is not generated when the counter is cleared by a hardware source or by software.

Figure 20.100 shows the relationship between the counter clearing by a hardware source and the GPTn_OVF (n = 0 to 3) interrupt.

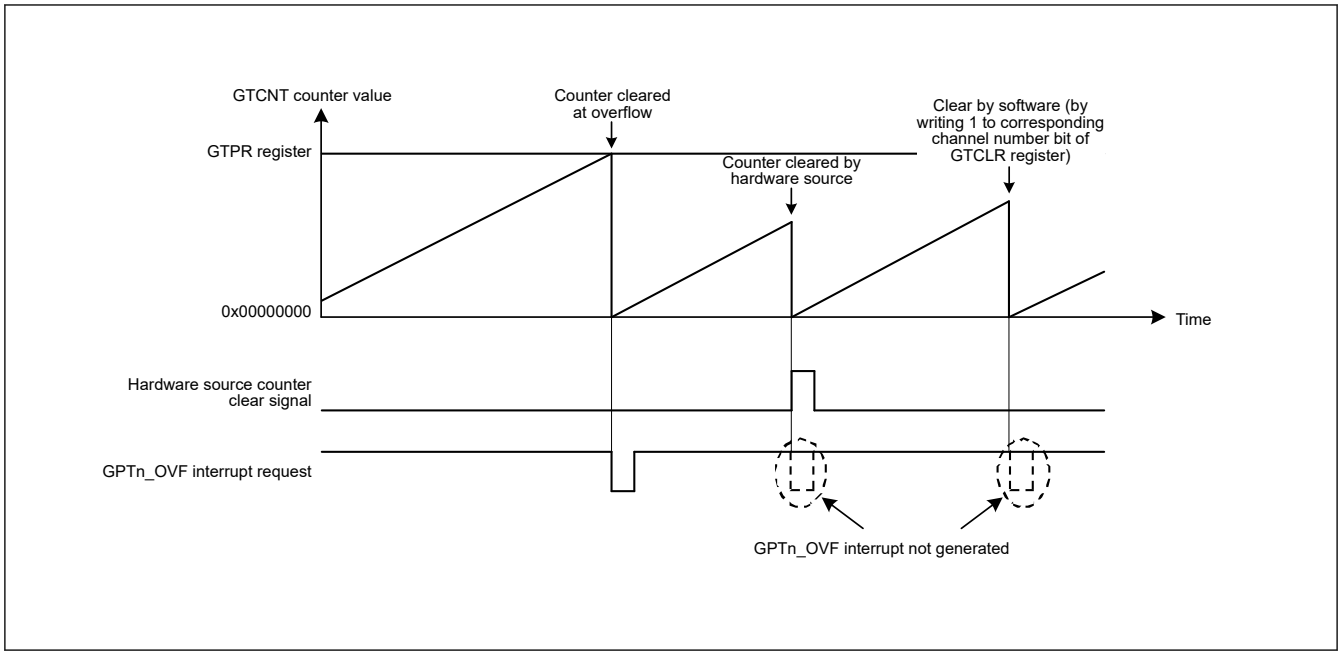
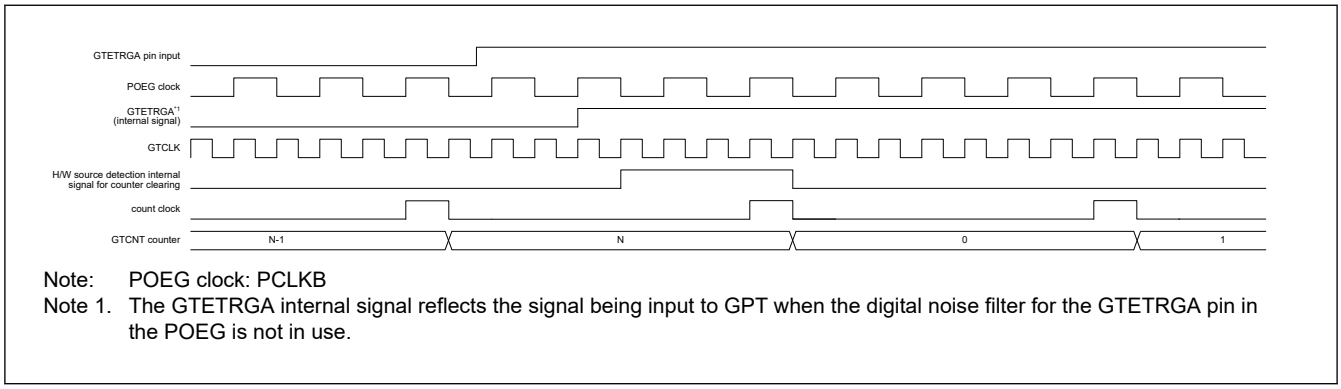


Figure 20.100 Relationship between counter clearing by hardware source and GPTn_OVF (n = 0 to 3) interrupt

Figure 20.101 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTETRGA pin when a clock signal produced by frequency-dividing the GTCLK signal is used as the counter clock for the GTCNT counter.

The GTCNT counter is cleared when counting is in progress after the GPT16 has detected the internal clearing signal.

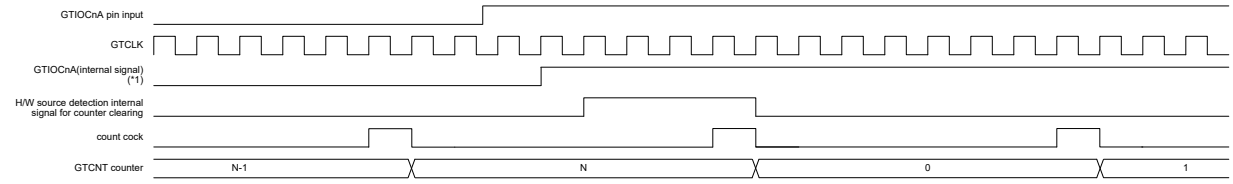


Note: POEG clock: PCLKB
Note 1. The GTETRGA internal signal reflects the signal being input to GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use.

Figure 20.101 Examples of the timing of operations for counter clearing in response to a rising edge of the input on the GTETRGA pin (during the counting of cycles of clock signal produced by dividing the GTCLK frequency)

Figure 20.102 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTIOCnA pin when a clock signal produced by frequency-dividing the GTCLK signal is used as the counter clock for the GTCNT counter.

The GTCNT counter is cleared when counting is in progress after the GPT16 has detected the internal clearing signal.



Note 1. This is the case when the digital noise filter is not in use.

Figure 20.102 Examples of timing of operations for counter clearing in response to a rising edge of the input on the GTIOCnA pin (during the counting of cycles of clock signal produced by dividing the GTCLK frequency)

Figure 20.103 shows an example of the timing of operations to clear the counter in response to event input from ELC_GPTA when a clock signal produced by frequency-dividing the GTCLK signal is used as the counter clock for the GTCNT counter.

This is an example of operations to clear counting by the GPT161.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPT160.GTCCRA register. This is selected as a trigger for output to the GPT161 by the ELC as ELC_GPTA.

The GPT160 compare match A signal is synchronized with GTCLK. The ELC receives the signal in synchronization with PCLKB, and outputs the GPT16 event source A signal after 1 cycle of PCLKB has elapsed.

The GTCNT counter is cleared when counting is in progress after the GPT16 has detected the internal clearing signal.

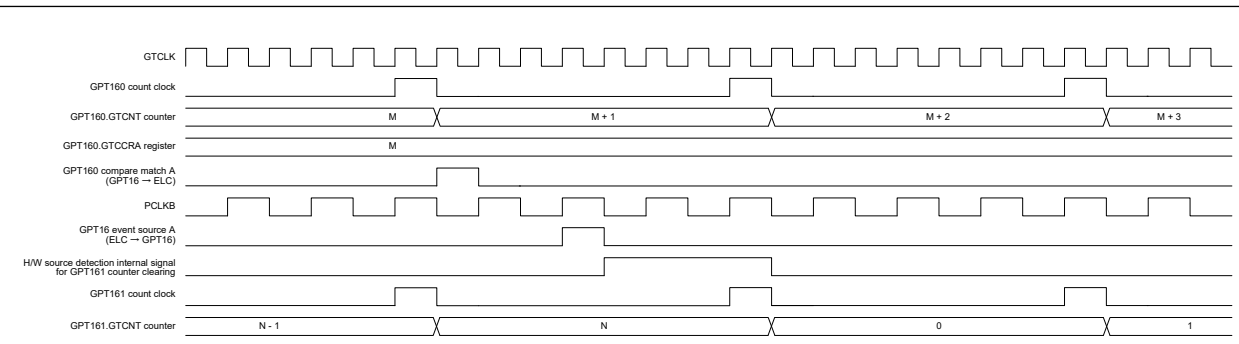
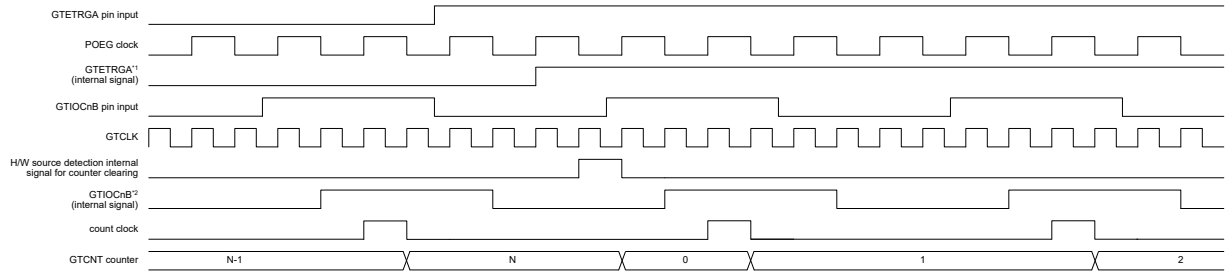


Figure 20.103 Examples of timing of operations for counter clearing in response to event input from the ELC_GPTA (during the counting of cycles of clock signal produced by dividing the GTCLK frequency)

Figure 20.104 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTETRGA pin input when counting is triggered by a hardware source.

The GTCNT counter is cleared in synchronization with GTCLK after the GPT16 has detected the internal clearing signal.

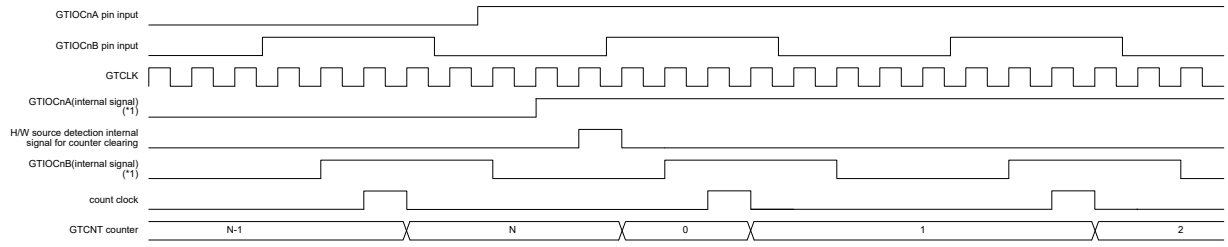


Note: POEG clock: PCLKB
Note 1. The GTETRGA internal signal reflects the signal being input to GPT when the digital noise filter for the GTETRGA pin in the POEG is not in use.
Note 2. This is the case when the digital noise filter is not in use.

Figure 20.104 Examples of the timing of operations for counter clearing in response to a rising edge of the input on the GTETRGA pin (during counting triggered by hardware source)

Figure 20.105 shows an example of the timing of operations to clear the counter in response to a rising edge of the input on the GTIOCnA pin input when counting is triggered by a hardware source.

The GTCNT counter is cleared in synchronization with GTCLK after the GPT16 has detected the internal clearing signal.



Note 1. This is the case when the digital noise filter is not in use.

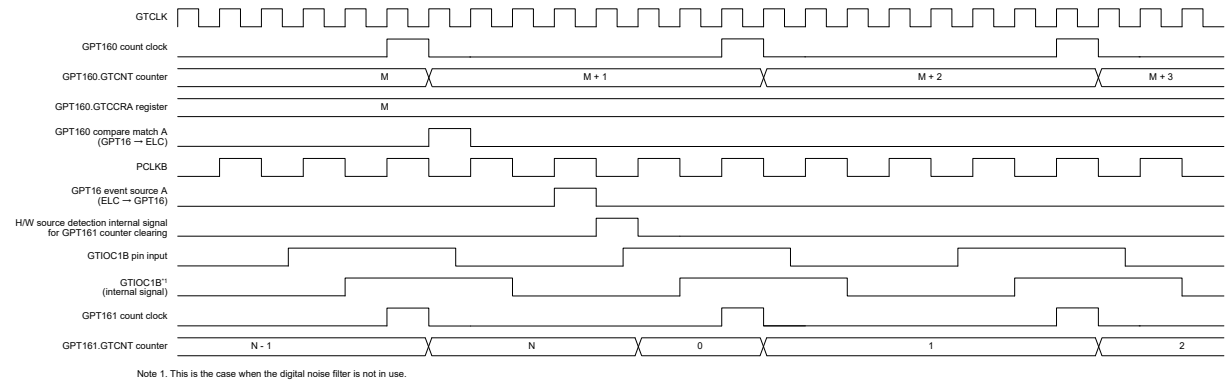
Figure 20.105 Examples of the timing of operations for counter clearing in response to a rising edge of the input on the GTIOCnA pin (during counting triggered by hardware source)

Figure 20.106 shows an example of the timing of operations for clearing the counter in response to the input of an event signal from the ELC_GPTA when counting is triggered by a hardware source.

This is an example of operations to clear counting by the GPT161.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPT160.GTCCRA register. This is selected as a trigger for output to the GPT161 by the ELC as ELC_GPTA.

The GPT160 compare match A signal is synchronized with GTCLK. The ELC receives the signal in synchronization with PCLKB, and outputs the GPT16 event source A signal after 1 cycle of PCLKB has elapsed.

The GTCNT counter is cleared in synchronization with GTCLK after the GPT16 has detected the internal clearing signal.



Note 1. This is the case when the digital noise filter is not in use.

Figure 20.106 Examples of the timing of operations for counter clearing in response to event input from the ELC_GPTA (during counting triggered by hardware source)

20.3.8 Synchronized Operation

Synchronized operation on channels such as a synchronized start, stop, and clear operation can be performed.

20.3.8.1 Synchronized Operation by Software

The GTCNT counters can be started, stopped, and cleared on multiple channels by setting the associated GTSTR, GTSTP, or GTCLR bits simultaneously to 1.

Count start with a phase difference is possible by setting the initial value in the GTCNT counter and setting the associated GTSTR bits simultaneously to 1.

When the GTCNT synchronous set/clear function is enabled on the GTCR.SSCEN bit, the GTCNT registers of the channel set in same group on the GTCR.SSCGRP [1:0] bits can be written at the same time.

The synchronous GTCNT write is invalid in complementary PWM mode.

When either the SSCE bit or the SSCD bit of the GTSECR register is set to 1, the GTCR.SSCEN bits of the channels that selected on the GTSECSR register are set to 0 or 1 and the GTCNT synchronous set/clear function of multiple channels are enabled or disabled at the same time.

Because the clock of count operation is selected by GTCR.TPCS[3:0] bits in respective channels, if the clock period of each channel that performs synchronous operation (count start/stop/clear) is different from others, the synchronous operation timings of every channels are not exact same.

Figure 20.107 shows an example of a simultaneous start, stop, and clear by software. Figure 20.108 shows an example of phase start operation by software. Figure 20.109, Figure 20.110, Figure 20.111 show an example of simultaneous start/stop/clearing with different count period.

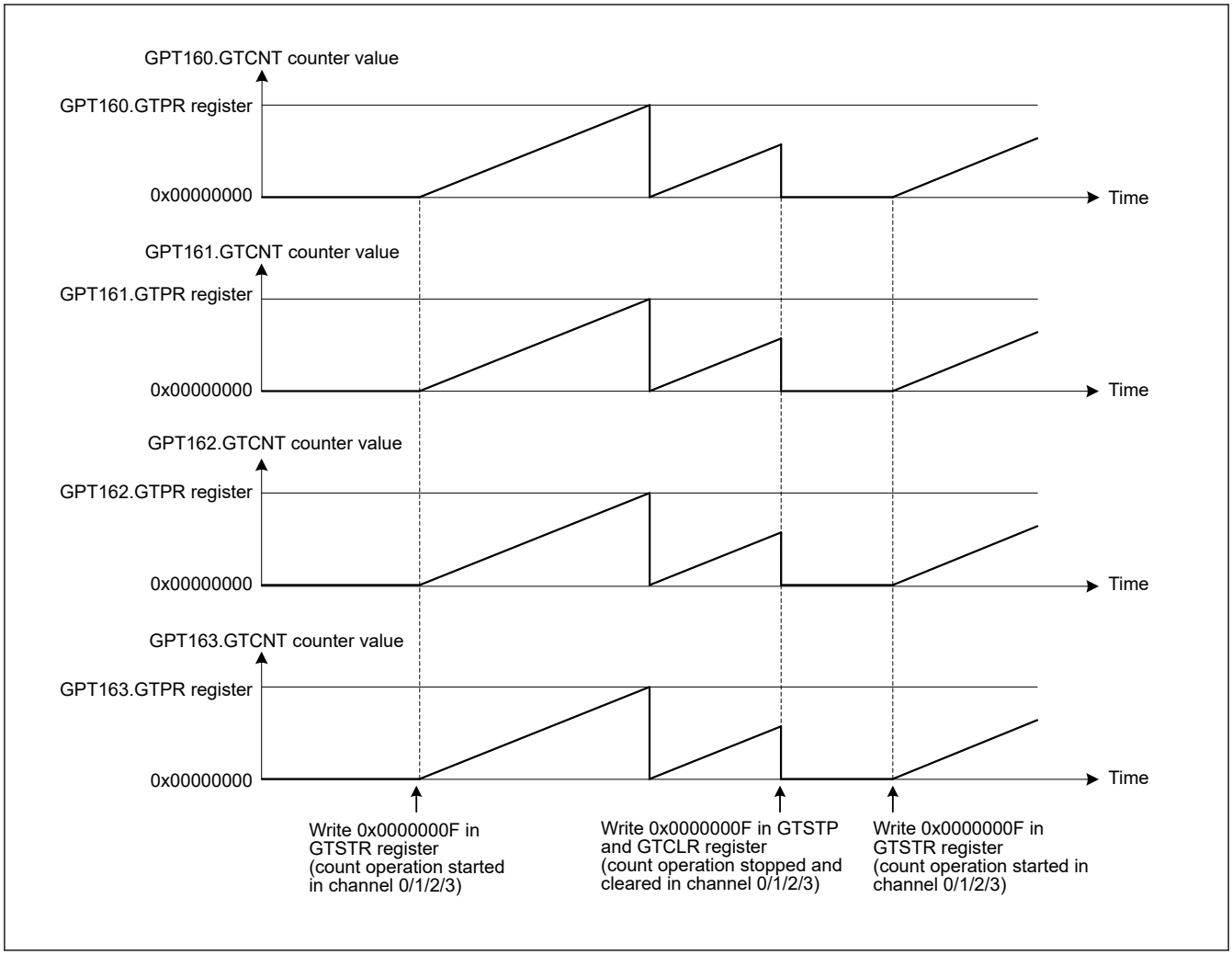


Figure 20.107 Example of a simultaneous start, stop, and clear by software with the same count cycle (GTPR register value)

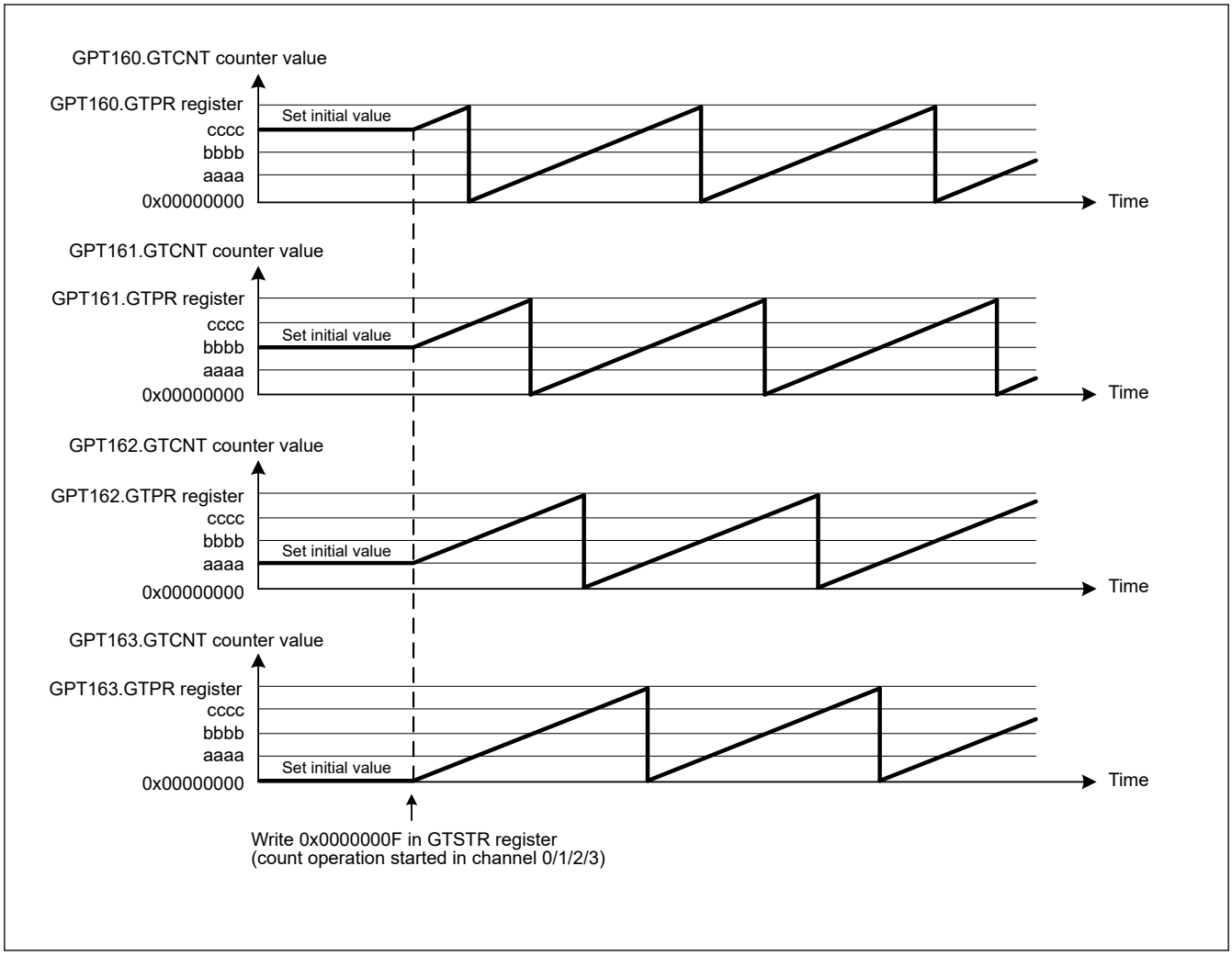


Figure 20.108 Example of software phase start with the same count cycle (GTPR register value)

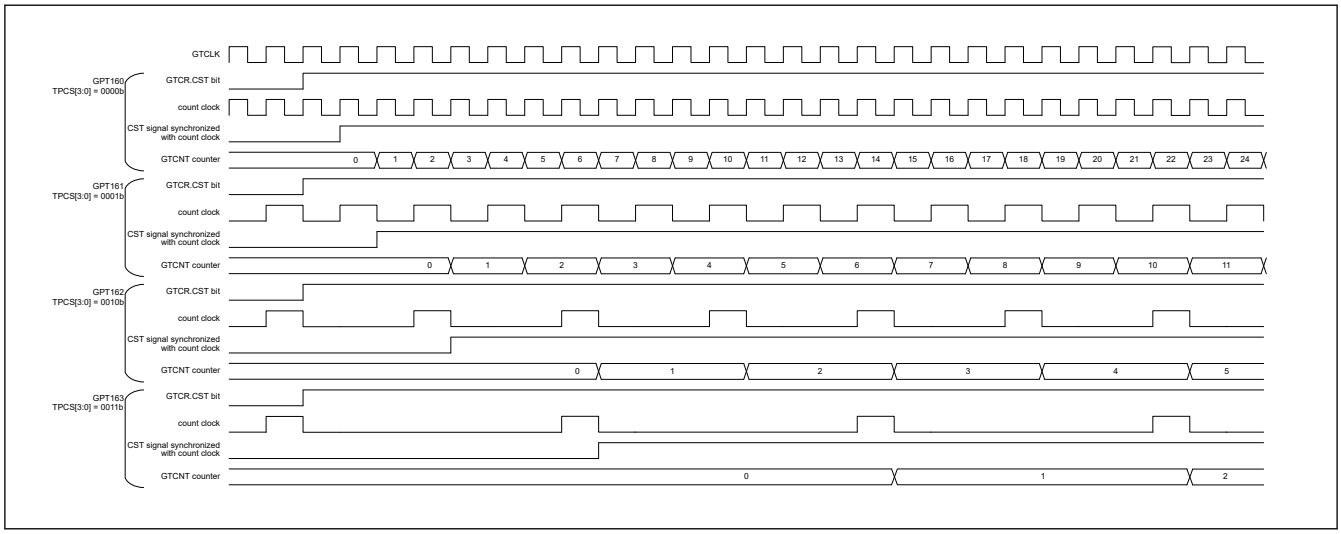


Figure 20.109 Example of simultaneous start operation by software (with different count period)

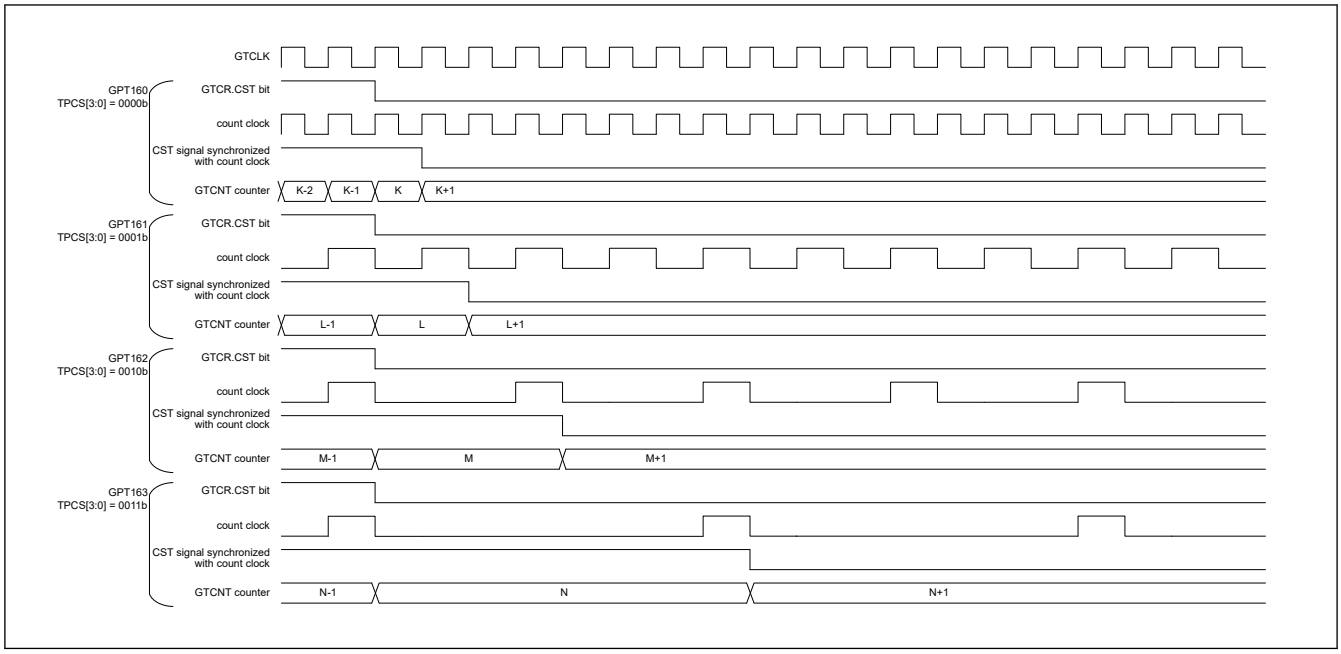


Figure 20.110 Example of simultaneous stop operation by software (with different count period)

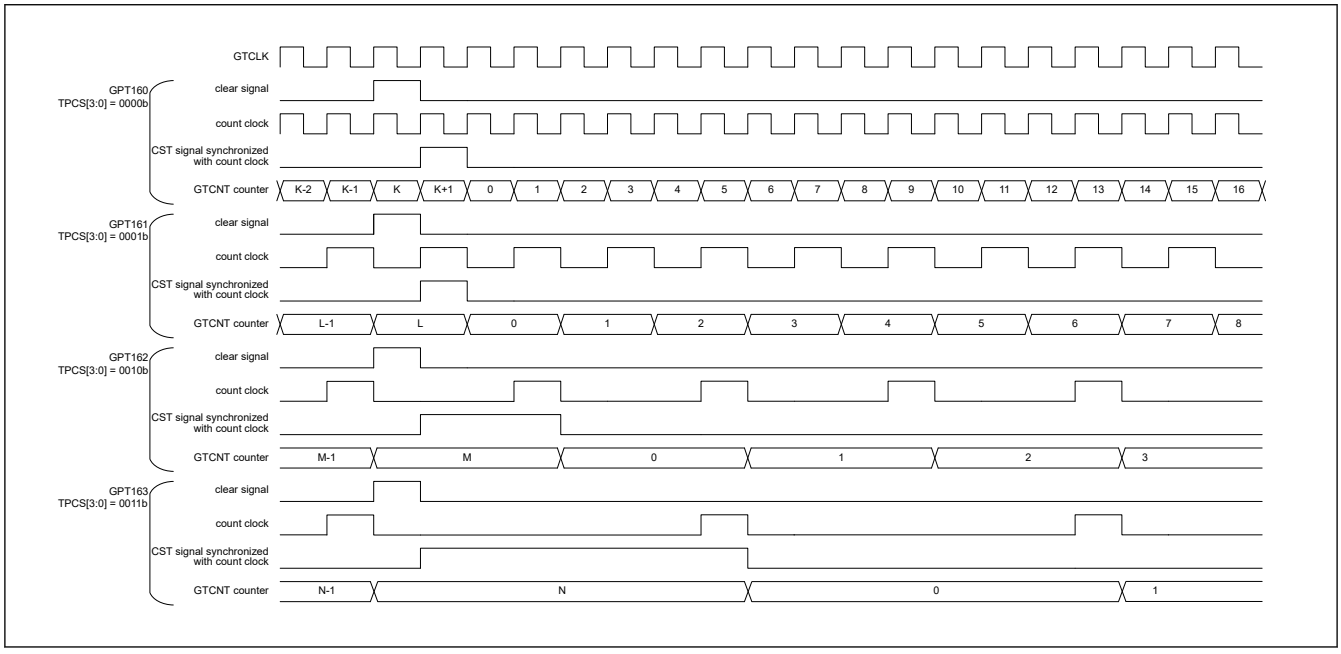
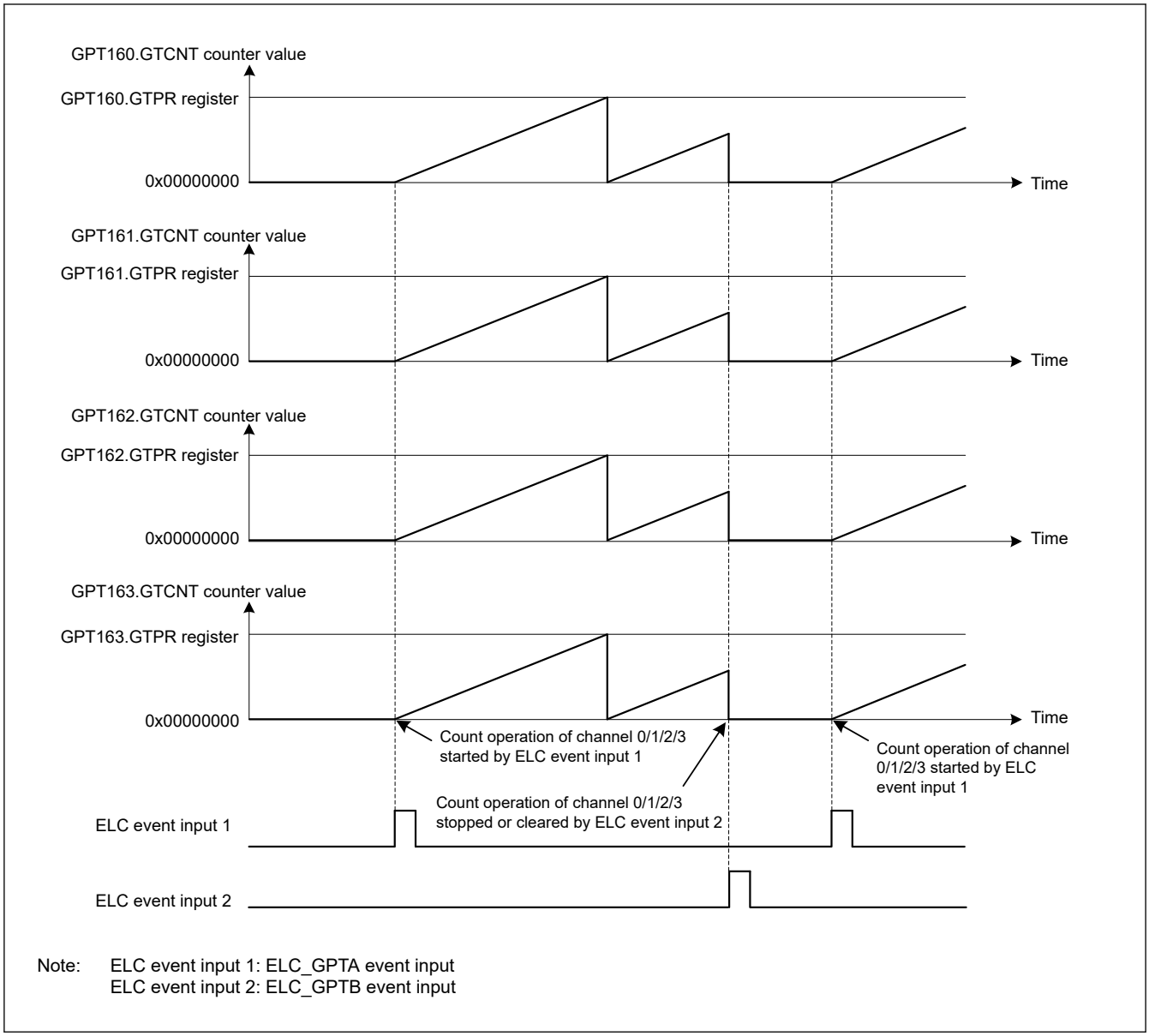


Figure 20.111 Example of simultaneous clearing operation by software (with different count period)

20.3.8.2 Synchronized Operation by Hardware

The counters for multiple channels can be started, stopped, and cleared simultaneously by the following hardware sources. Hardware sources that can cause a synchronized operation are external trigger input and ELC event input. Synchronized operation through the GTIOCnA and GTIOCnB pin inputs is possible by setting an ELC event due to input capture as a hardware source (n = 0 to 3).

Figure 20.112 shows an example of a simultaneous start, stop, and clear operation by a hardware source. Table 20.50 shows the setting example.



Note: ELC event input 1: ELC_GPTA event input
ELC event input 2: ELC_GPTB event input

Figure 20.112 Example of a simultaneous start, stop, and clear by a hardware source with the same count cycle (GTPR register value)

Table 20.50 Example setting for simultaneous start by a hardware source (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.112, 000b or 0000b (saw-wave PWM mode 1) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.112, after 11b is set in the GTUDDTYC.UDF and UD bits, 01b is set in the GTUDDTYC.UDF and UD bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 20.112, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation with the GTSSR register, and wait for count start by the hardware source. In Figure 20.112, GTSSR.SSELCA = 1.

Table 20.50 Example setting for simultaneous start by a hardware source (2 of 2)

No.	Step Name	Description
7	Set hardware count stop	Select a hardware source for stopping count operation with the GTPSR register, and wait for count stop by the hardware source. In Figure 20.112, GTPSR.PSELCB = 1.
8	Set hardware count clear	Select a hardware source for clearing count operation with the GTCSR register, and wait for count clear by the hardware source. In Figure 20.112, GTCSR.CSELCB = 1.
9	Set hardware source operation	Set operation of the hardware source selected in the GTSSR, GTPSR, or GTCSR registers, and start, stop, or clear counting. In Figure 20.112, ELC_GPTA input and ELC_GPTB input are set.

20.3.8.3 Synchronous Clear Operation by Inter Channel Cooperation

The counters of other channels can be cleared at the same time with the counter clear of some channel caused by the compare match, input capture, saw wave up-count overflow, saw wave down-count underflow, and the GTIOCnA / GTIOCnB pin input selected in the GTCSR register.

The counter clear caused by GTCLR register, external trigger input and the ELC input can perform as synchronous clear when the same counter clear factor is set to target channels of synchronous clear. Therefore, these factors are not prepared for the synchronous clear operation by inter channel cooperation.

Set the channel that generates the synchronization clear source and the channel that is synchronized clear in the same synchronous set/clear group with the GTCR.SSCGRP[1:0] bits.

Similar to the synchronous operation in section 20.3.8.1. Synchronized Operation by Software, if the count clock selected by the GTCR.TPCS [3:0] bits is different for each channel, the synchronous clear operation cannot be performed at exact same timing. Similar to the example of simultaneous clearing operation by software in Figure 20.111, if the count clock is different for each channel, the synchronous clear factor is synchronized with the count clock of each channel before counter clearing.

When either the SSCE bit or the SSCD bit of the GTSECR register is set to 1, the GTCR.SSCEN bits of the channels that selected on the GTSECSR register are set to 0 or 1 and the GTCNT synchronous set/clear function of multiple channels are enabled or disabled at the same time.

Table 20.51 shows synchronous clear factor register setting and Figure 20.113 shows an example of synchronous clear operation by inter channel cooperation.

Table 20.52 and Table 20.53 show examples for setting synchronous clear operation by inter channel cooperation.

Table 20.51 Synchronous clear factor register setting (1 of 2)

Synchronous clear factor by inter channel cooperation	Setting register	Setting value
GTCCRA register compare match/input capture	GTINTAD.SCFA	1b
	GTCSR.CSCMSC[2:0]	001b
GTCCRB register compare match/input capture	GTINTAD.SCFB	1b
	GTCSR.CSCMSC[2:0]	010b
GTCCRC register compare match	GTINTAD.SCFC	1b
	GTCSR.CSCMSC[2:0]	011b
GTCCRD register compare match	GTINTAD.SCFD	1b
	GTCSR.CSCMSC[2:0]	100b
GTCCRE register compare match	GTINTAD.SCFE	1b
	GTCSR.CSCMSC[2:0]	101b
GTCCRF register compare match	GTINTAD.SCFF	1b
	GTCSR.CSCMSC[2:0]	110b
Overflow of saw-wave up-count	GTINTAD.SCFPO	1b

Table 20.51 Synchronous clear factor register setting (2 of 2)

Synchronous clear factor by inter channel cooperation	Setting register	Setting value
Underflow of saw-wave down-count	GTINTAD.SCFPU	1b
Clear caused by GTIOCnA/GTIOCnB pin	GTCR.SCGTIOC	1b

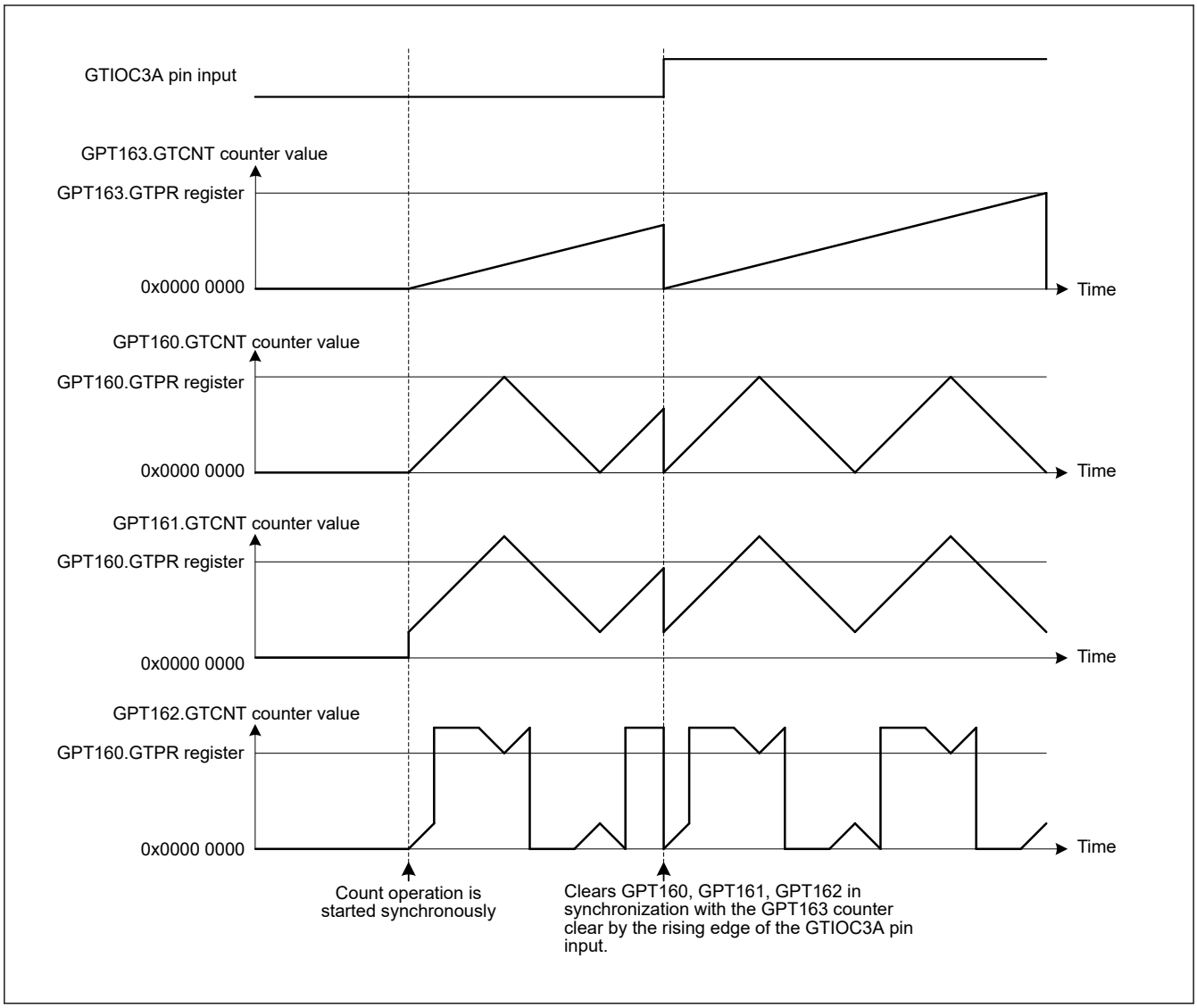


Figure 20.113 Example of synchronous clear operation by inter channel cooperation (clear caused by GTIOCnA/GTIOCnB pin) (GPT163 is saw-wave and counter is cleared by the rising edge of the GTIOC3A, GPT160, 1, 2 are complementary PWM mode. GPT160, 1, 2, 3 are the same synchronous set/clear group)

Table 20.52 Example for setting synchronous clear operation by inter channel cooperation (clear caused by GTIOCnA/GTIOCnB pin) (1 of 2)

No.	Step name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0] or GTCR.MD[3:0]. In Figure 20.113, 000b (saw-wave PWM mode 1) is set to GPT163 and 1100b (complementary PWM mode 1) is set to GPT160 (GPT161, 2).
2	Set count direction	Select the count direction (up or down) with GTUDDTYC. In Figure 20.113, for GPT163, the lower 2 bits of GTUDDTYC are set to 11b, and the lower 2 bits of GTUDDTYC are set to 01b (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits of the corresponding channel.

Table 20.52 Example for setting synchronous clear operation by inter channel cooperation (clear caused by GTIOCnA/GTIOCnB pin) (2 of 2)

No.	Step name	Description
4	Set cycle	Set the cycle in GTPR of the corresponding channel. In complementary PWM mode, set the cycle in GTPR of the master channel.
5	Set initial value for counter	Set the initial value in the GTCNT counter of the corresponding channel.
6	Inter channel cooperation synchronous clear setting (source channel)	Set the GTINTAD register, GTCSR.CSCMSC[2:0] bits, and GTCR.SCGTIOC bit in the source channel of inter channel cooperation synchronous clear to enable synchronous clear. In complementary PWM mode, set GTINTAD of the master channel. In Figure 20.113, GPT163.GTCR.SCGTIOC bit is 1.
7	Inter channel cooperation synchronous clear setting (cleared channels)	Set GTCSR.CSCMSC[2:0] bits in the cleared channels of inter channel cooperation synchronous clear to select the counter clear by synchronous counter clearing group. In complementary PWM mode, set GTCSR of the master channel. In Figure 20.113, GTCSR.CSCMSC[2:0] bits of GPT160 are 111b.
8	Set group of inter channel cooperation synchronous clear	Set the same value to GTCR.SSCGRP[1:0] bits in the source channel and cleared channels of inter channel cooperation synchronous clear and set them in the same synchronous set/clear group. In complementary PWM mode, set GTCR of the master channel.
9	Enable inter channel cooperation synchronous clear	Set GTCR.SSCEN bits in the source channel and cleared channels of inter channel cooperation synchronous clear to enable synchronous clear.

Table 20.53 Example for setting synchronous clear operation by inter channel cooperation (clear caused by the GTCCRA input capture using the GTIOCnA/GTIOCnB pin)

No.	Step name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits.
2	Set count direction	Select the count direction (up or down) with GTUDDTYC register.
3	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits of the corresponding channel.
4	Period setting	Set the cycle in GTPR of the corresponding channel. In complementary PWM mode, set the cycle in GTPR of the master channel.
5	Set initial value for counter	Set the initial value in the GTCNT counter of the corresponding channel.
6	Inter channel cooperation synchronous clear setting (source channel)	Set the input capture factor using GTIOCnA/B pin to GTICASR and set the same factor as the GTICASR to the GTCSR as the count clearing factor. In addition, set GTINTAD.SCFA bit = 1b and GTCSR.CSCMSC[2:0] bits = 001b to enable synchronous clear source by the GTCCRA input capture. In complementary PWM mode, set GTINTAD of the master channel.
7	Inter channel cooperation synchronous clear setting (cleared channels)	Set GTCSR.CSCMSC[2:0] bits in the cleared channels of inter channel cooperation synchronous clear to 111b and select the counter clear by synchronous counter clearing group. In complementary PWM mode, set GTCSR of the master channel.
8	Set group of inter channel cooperation synchronous clear	Set the same value to GTCR.SSCGRP[1:0] bits in the source channel and cleared channels of inter channel cooperation synchronous clear and set them in the same synchronous set/clear group. In complementary PWM mode, set GTCR of the master channel.
9	Enable inter channel cooperation synchronous clear	Set GTCR.SSCEN bits in the source channel and cleared channels of inter channel cooperation synchronous clear to enable synchronous clear.

20.3.8.4 Input Capture Operation by Inter Channel Cooperation

The events of compare match, input capture, saw-wave up-count overflow, saw-wave down-count underflow, the crest and trough of triangle wave and complementary PWM mode and the count clock can be used as the input capture factor for the GTCCRm register (m = A or B) of other channels. The input capture factors by inter channel cooperation can be set by the GTICCR register of the channel that generates the input capture factor. The input capture by other channel factors can be enabled by GTICmSR.mSOC bit (m = A or B) in the captured channels. The channels that generate input capture factors and the channels that are captured are set to the same input capture group by GTICCR.ICmGRP[1:0] bits (m = A or B).

Figure 20.114 shows an example of input capture operation by inter channel cooperation and Table 20.54 shows an example for setting input capture operation by inter channel cooperation.

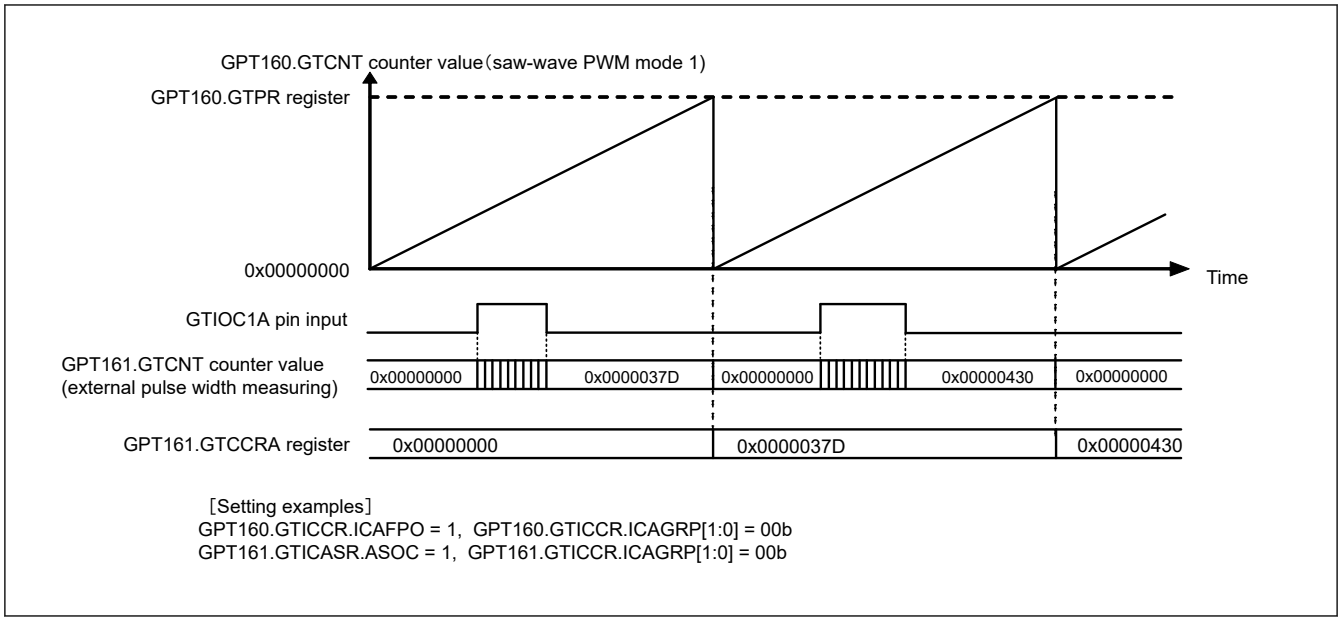


Figure 20.114 Example of input capture operation by inter channel cooperation (channel 1 is captured by the overflow of channel 0)

Table 20.54 Example for setting input capture operation by inter channel cooperation using the GTIOCnA/ GTIOCnB pin

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] or GTCR.MD[3:0] bits. In Figure 20.114, GTCR.MD[2:0] = 000b or GTCR.MD[3:0] = 0000b (saw-wave PWM mode 1) is set to GPT160.
2	Set external pulse width measuring function	Enable external pulse width measuring function with the GTUPSR.USILVL [3:0] bits for up-counting operation and the GTDNSR.DSILVL[3:0] bits for down-counting operation and select the input pin and level to measure. In Figure 20.114, GTUPSR.USILVL[3:0] = 0011b (count up when GTIOCnA pin is 1).
3	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 20.114, for GPT160, the lower 2 bits of the GTUDDTYC register is set to 11b, and the lower 2 bits of the GTUDDTYC register is set to 01b (up-counting).
4	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits of the corresponding channel.
5	Period setting	Set the cycle in GTPR of the corresponding channel. In complementary PWM mode, set a period in the GTPR of the master channel.
6	Set initial value for counter	Set the initial value in the GTCNT counter of the corresponding channel.
7	Set counter clear factor to the inter channel cooperation input capture (cleared channels)	Set the GTCSR.CSCMSC[2:0] bits in the channel to be cleared by input capture by inter channel cooperation to 001b, 010b, and select input capture as counter clear factor. In Figure 20.114, GPT161.GTCSR.CSCMSC[2:0] bits are 001b.
8	Inter channel cooperation input capture setting (source channel)	Set the input capture factor using the GTICCR register in the source channels of inter channel cooperation input capture. In Figure 20.114, GPT160.GTICCR.ICAFPO bit is 1b.
9	Inter channel cooperation input capture setting (captured channels)	Set the GTICmSR.mSOC bit (m = A or B) in the captured channels to allow input capture to the GTCCRm register (m = A or B) by other channel factors. In Figure 20.114, GPT161.GTICASR.ASOC bit is 1b.
10	Set group of inter channel cooperation input capture	Set the same value to the GTICCR.ICmGRP[1:0] bits (m = A or B) in the source channels and cleared channels of inter channel cooperation input capture and set them in the same input capture group.

20.3.9 PWM Output Operation Examples

(1) Synchronized PWM output

The GPT outputs 4 × 2 phases of linked PWM waveforms for a maximum of GPT × 4 channels.

Figure 20.115 shows an example in which four channels perform synchronized operation in saw-wave PWM mode 1 and eight phases of PWM waveforms are output. The GTIOCnA is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCnB is set so that it outputs low as the initial value, high at a GTCCRB compare match, and low at the cycle end.

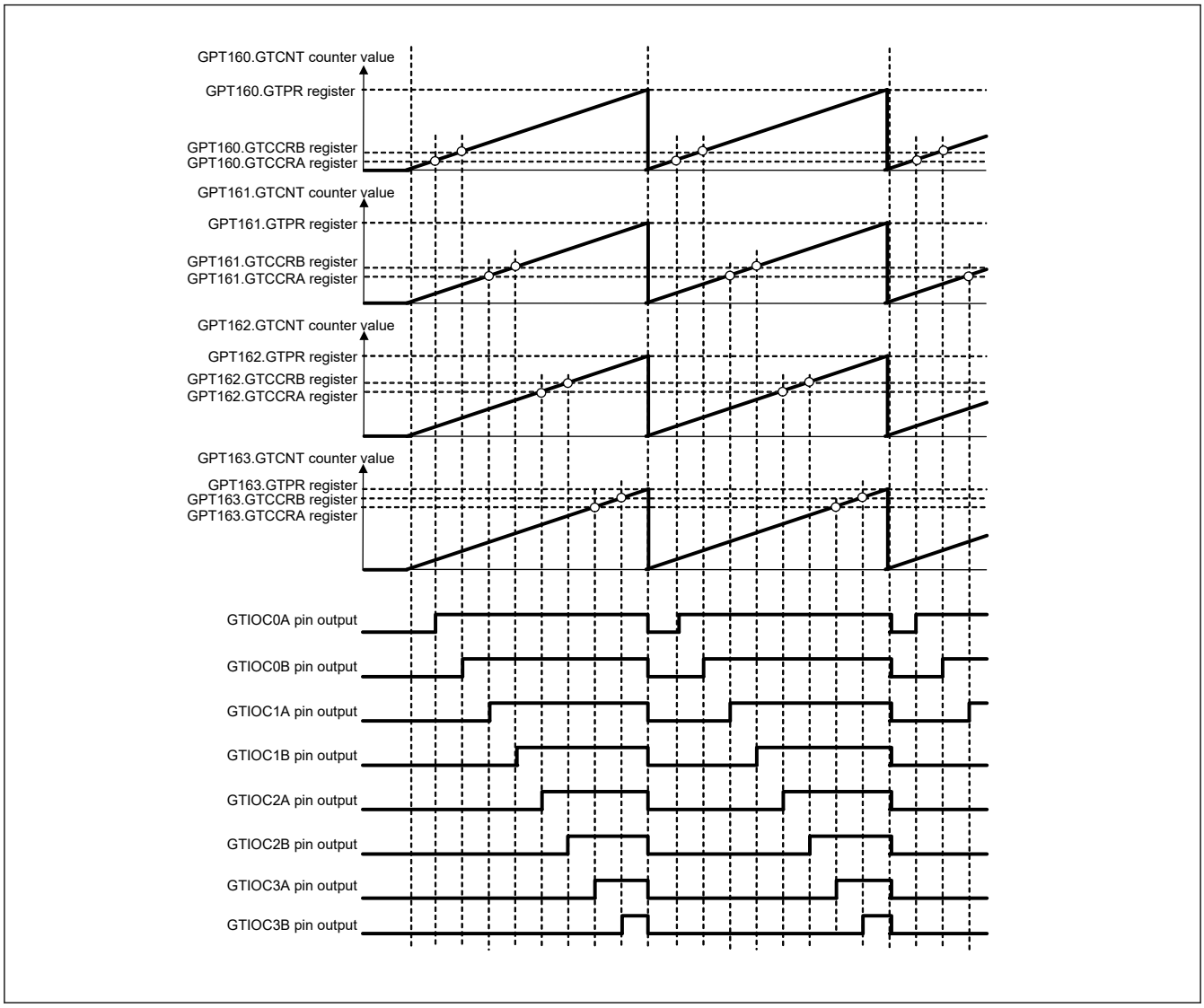


Figure 20.115 Example of synchronized PWM output

(2) 3-phase saw-wave complementary PWM output

Figure 20.116 shows an example in which three channels perform synchronized operation in saw-wave PWM mode 1 and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, low at a GTCCRB compare match, and high at the cycle end.

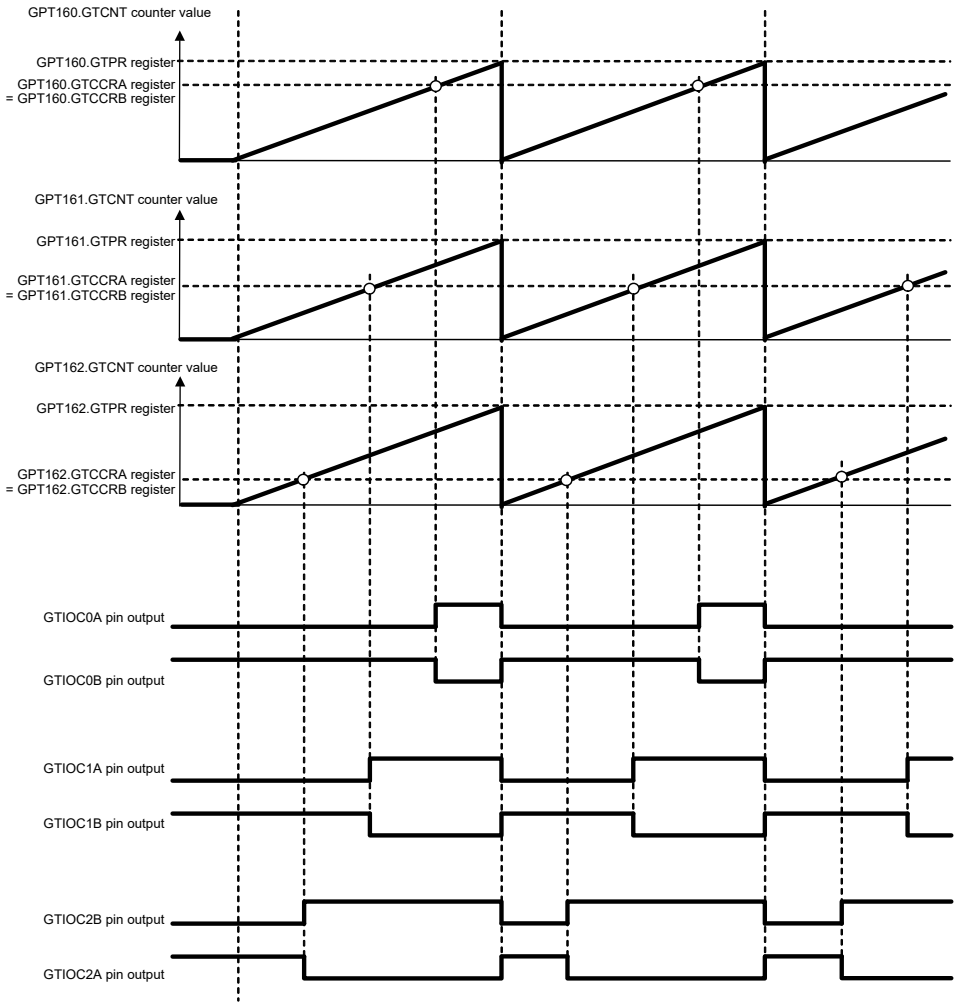


Figure 20.116 Example of 3-phase saw-wave complementary PWM output

(3) 3-phase saw-wave complementary PWM output with automatic dead time setting

Figure 20.117 shows an example in which three channels perform synchronized operation in saw-wave one-shot pulse mode with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTIOCnB compare match, and retains the output at the cycle end.

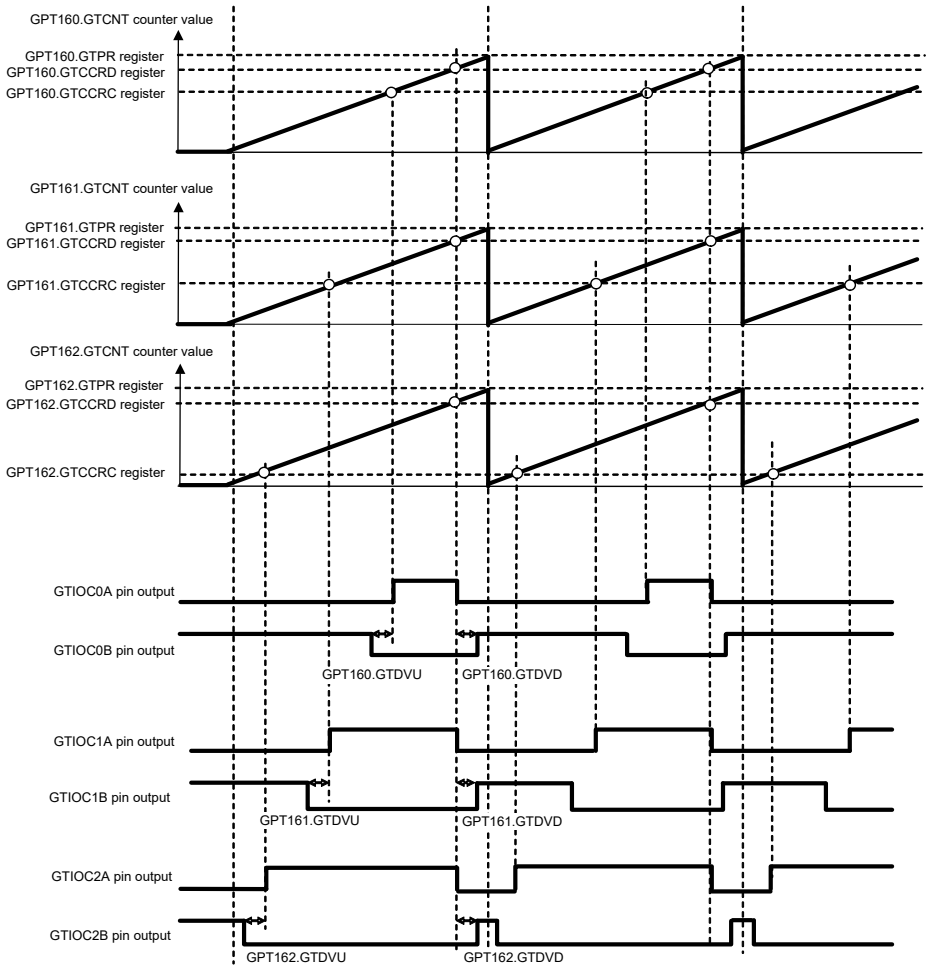


Figure 20.117 Example of 3-phase saw-wave complementary PWM output with automatic dead time setting

(4) 3-phase triangle-wave complementary PWM output

Figure 20.118 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 1 and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

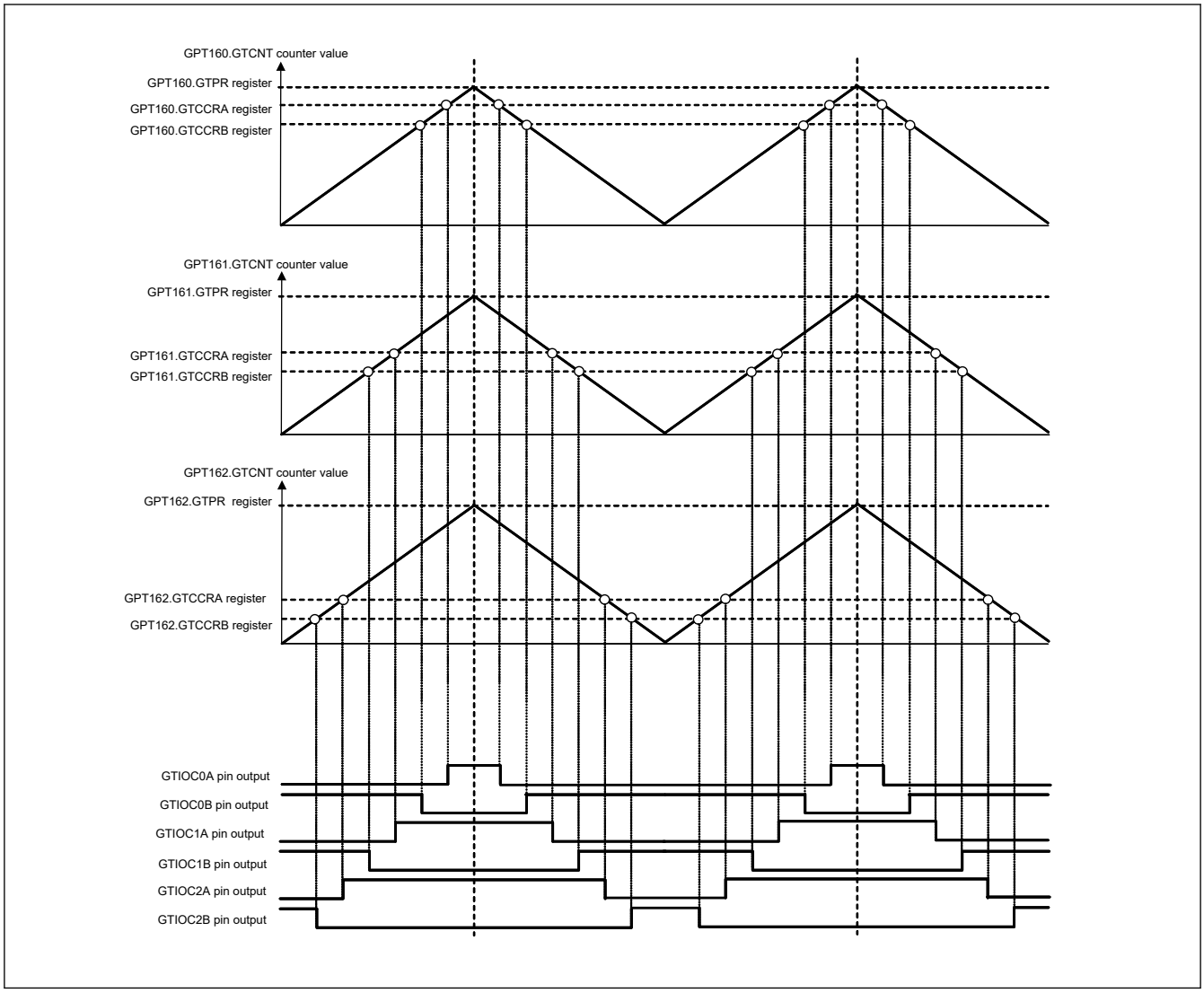


Figure 20.118 Example of 3-phase triangle-wave complementary PWM output

(5) 3-phase triangle-wave complementary PWM output with automatic dead time setting

Figure 20.119 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 1 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

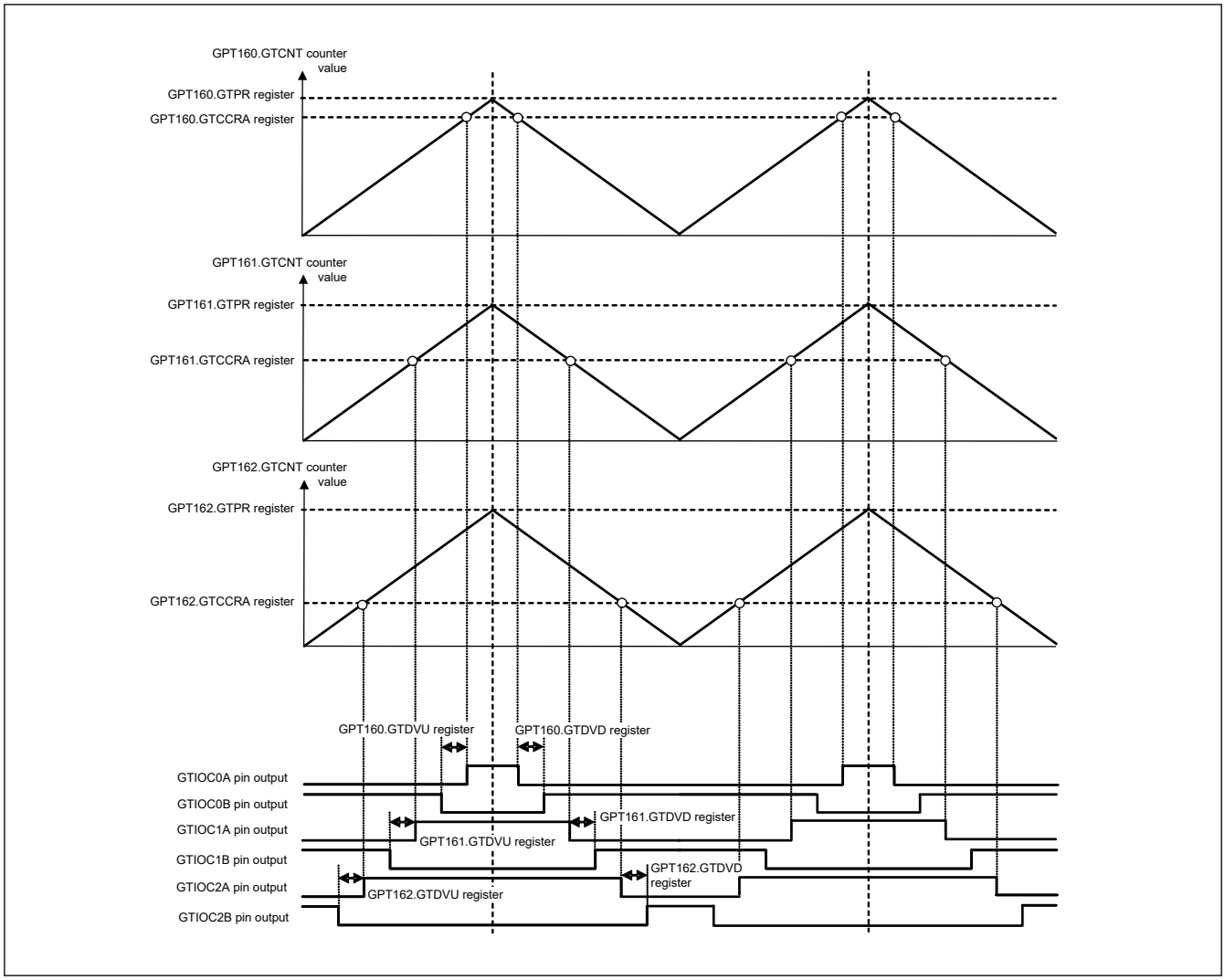


Figure 20.119 Example of 3-phase triangle-wave complementary PWM output with automatic dead time setting

(6) 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting

Figure 20.120 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 3 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.

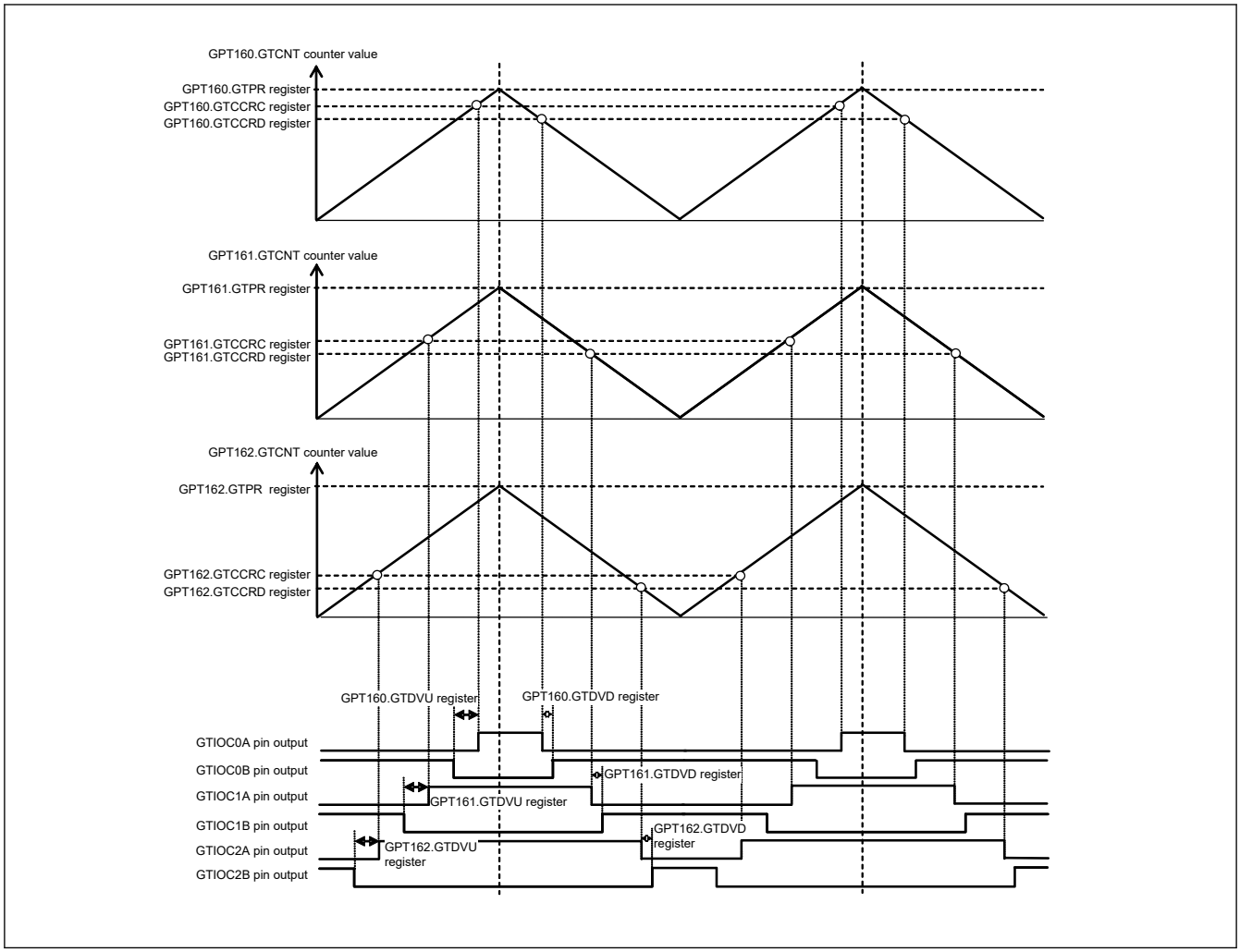


Figure 20.120 Example of 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting

20.3.10 Period Count Function

By setting the GTPC register, the end of period can be counted.

The number of period to be counted should be set into the GTPC.PCNT counter when the GTPC.PCEN bit is 0. When the PCEN bit is 1, the PCNT counter can be read, but writing is disabled. When the PCEN bit is 1, down-counting is performed at the end of period. When the PCNT counter is 1 at the end of period, it becomes 0 and counting is stopped to finish the period count function. At that time, the GTST.PCF flag is set, and the period count function finish interrupt request GPTn_PC is generated. When the GTPC.ASTP bit is 1, the GTCNT counter is also stopped at the same time that the period count function is finished.

When the GTCNT counter is stopped while period count function is enabled, the PCNT counter keeps its value. When the GTCNT counter restarts counting and the PCEN bit is 1, the PCNT counter restarts down-counting from the hold value.

When the PCEN bit is changed from 0 to 1 while the PCNT counter is 0 and the ASTP bit is 1, the GTCNT counter is stopped at the count clock immediately after that.

When either GTSECR.SPCE bit or GTSECR.SPCD bit is set to 1, the PCEN bit in the channels set to 1 by the GTSECSR register is simultaneously set the value to enable or disable the period count function for multiple channels.

Figure 20.121 and Figure 20.122 show examples of PWM cycle count function.

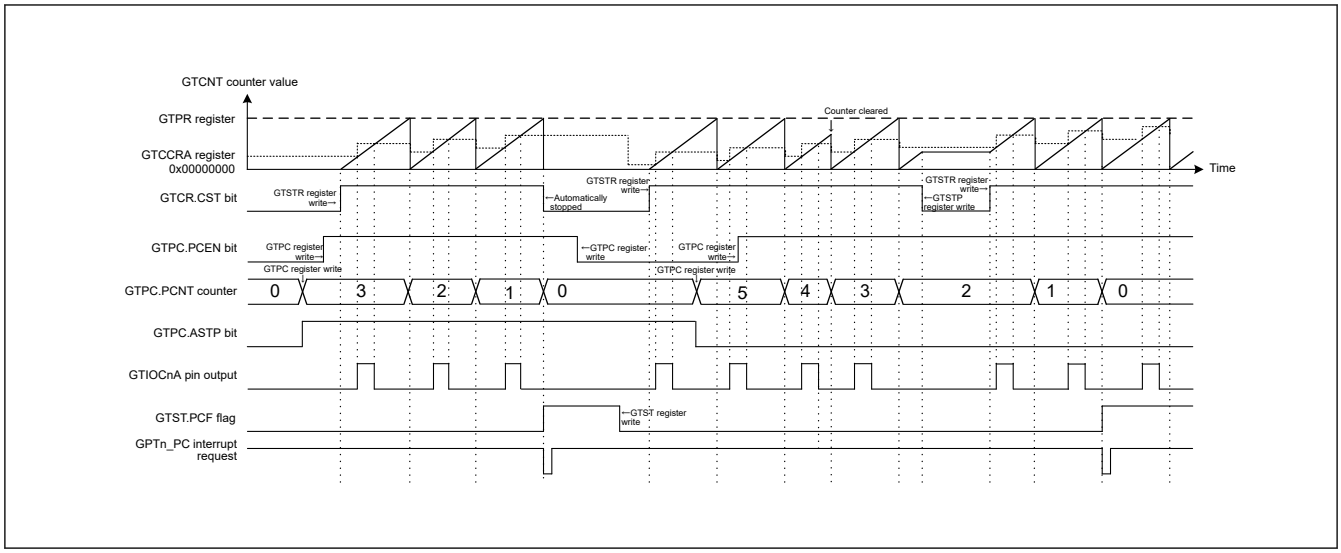


Figure 20.121 Example of PWM cycle count function (saw-wave one-shot pulse mode)

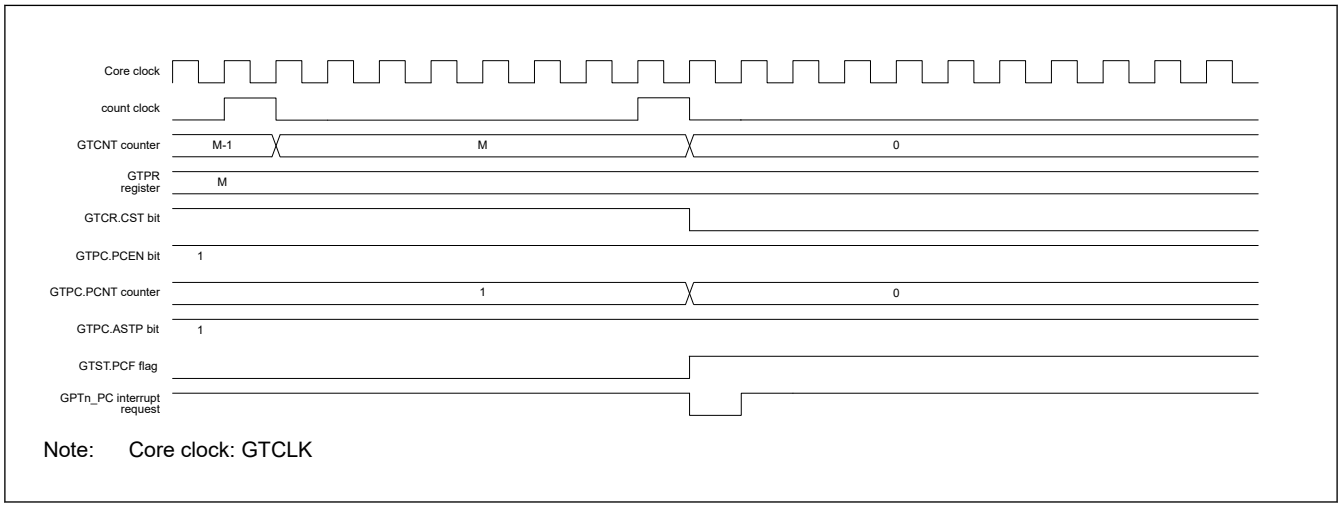


Figure 20.122 Example of the timing of operations for PWM cycle count function (saw-wave one-shot pulse mode, up-counting)

20.3.11 Phase Counting Function

The phase difference between the GTIOCnA and GTIOCnB pin (n = 3) inputs is detected and the associated GTCNT counts up or counts down. The detectable phase difference is available in any combination with the relationship between the edge and the level of GTIOCnA and GTIOCnB pin inputs being set in the GTUPSR and GTDNSR registers. For details on count operation, see section 20.3.1.1. Counter Operation.

Figure 20.123 to Figure 20.132 show an example of phase counting modes 1 to 5 operation when the GTIOCnA, GTIOCnB pins are used. Table 20.55 to Table 20.64 show conditions of up-counting or down-counting and list settings for the GTUPSR and GTDNSR registers which is corresponding to Figure 20.123 to Figure 20.132.

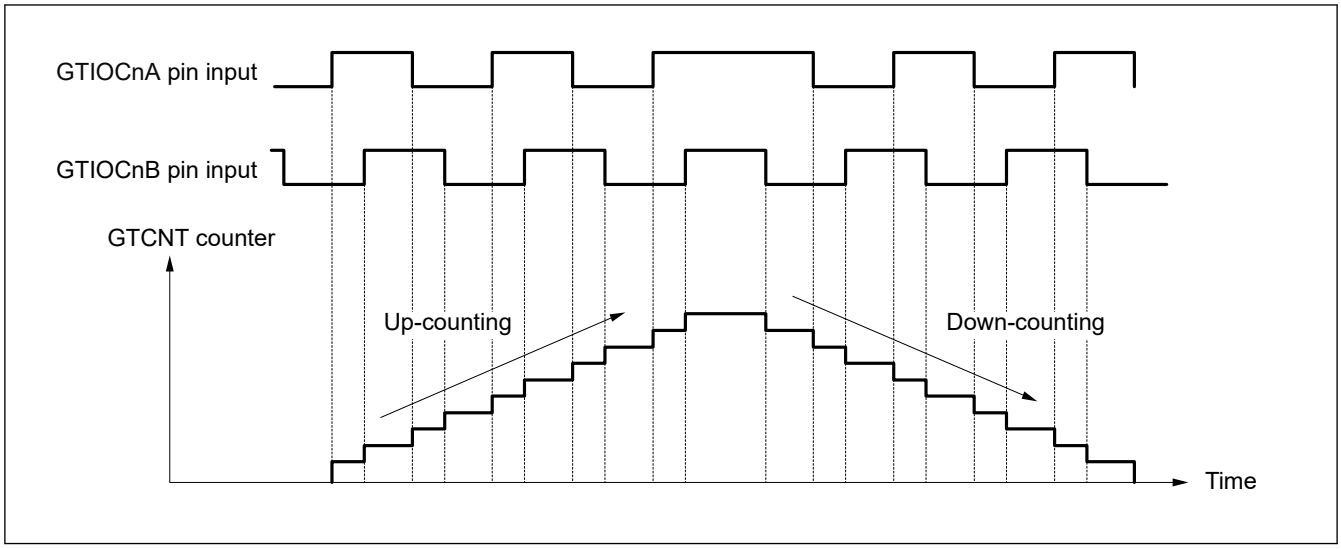


Figure 20.123 Example of phase counting mode 1

Table 20.55 Conditions of up-counting/down-counting in phase counting mode 1

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Up-counting	GTUPSR = 0x00006900 GTDNSR = 0x00009600
Low	↓		
↑	Low		
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High		
↓	Low		

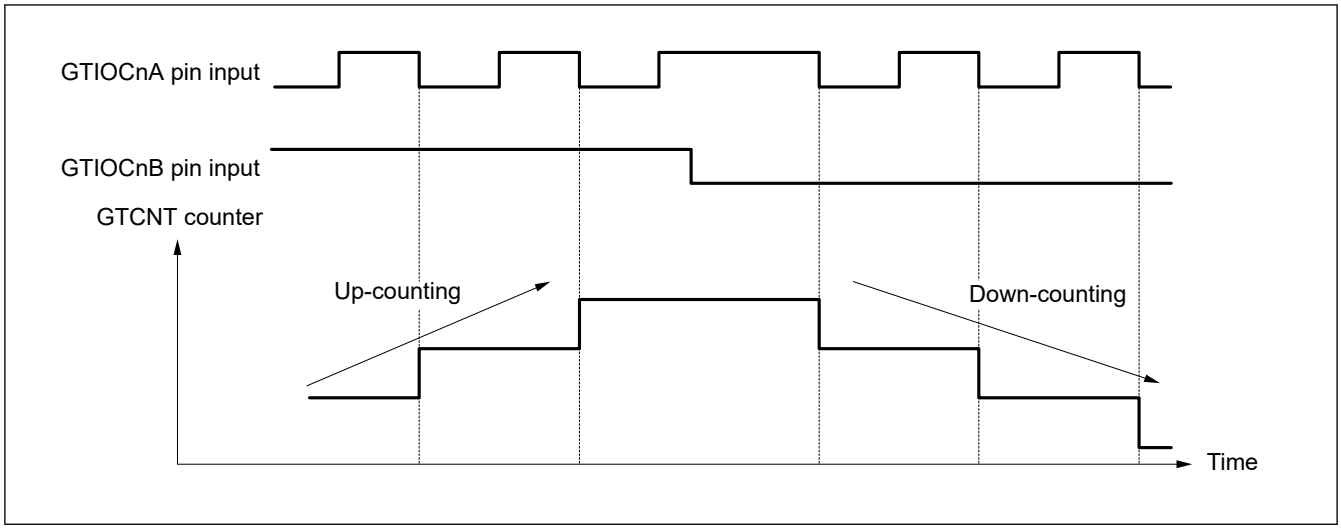


Figure 20.124 Example of phase counting mode 2 (A)

Table 20.56 Conditions of up-counting/down-counting in phase counting mode 2 (A)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000800 GTDNSR = 0x00000400
Low			
	Low		
	High	Up-counting	
High		Not counting	
Low			
	High		
	Low	Down-counting	

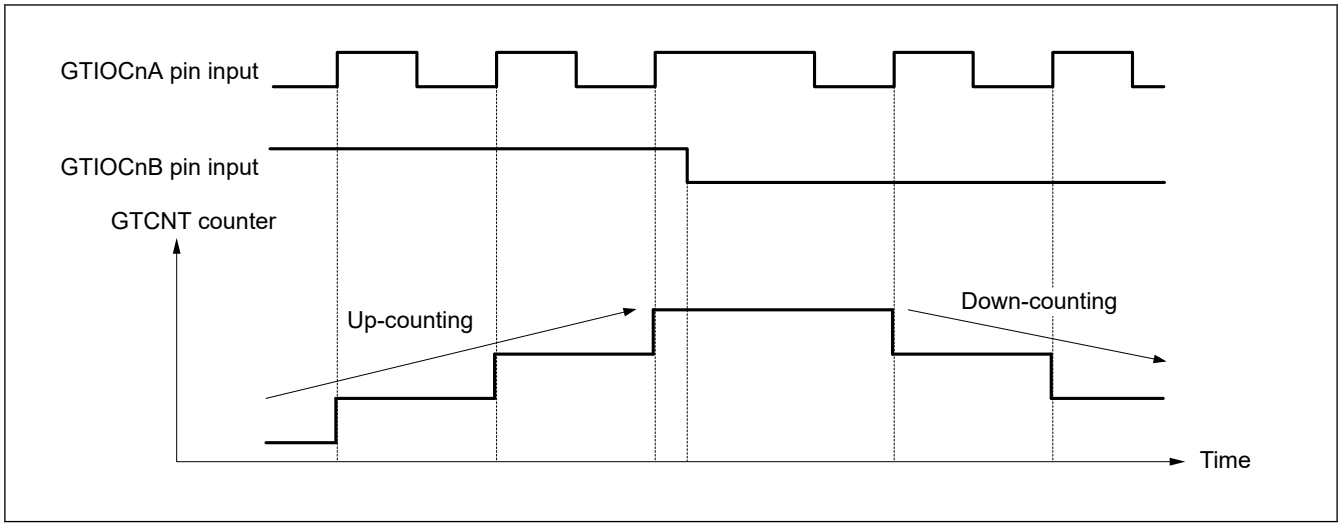


Figure 20.125 Example of phase counting mode 2 (B)

Table 20.57 Conditions of up-counting/down-counting in phase counting mode 2 (B)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Not counting	GTUPSR = 0x00000200 GTDNSR = 0x00000100
Low	↓		
↑	Low	Down-counting	
↓	High	Not counting	
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Not counting	

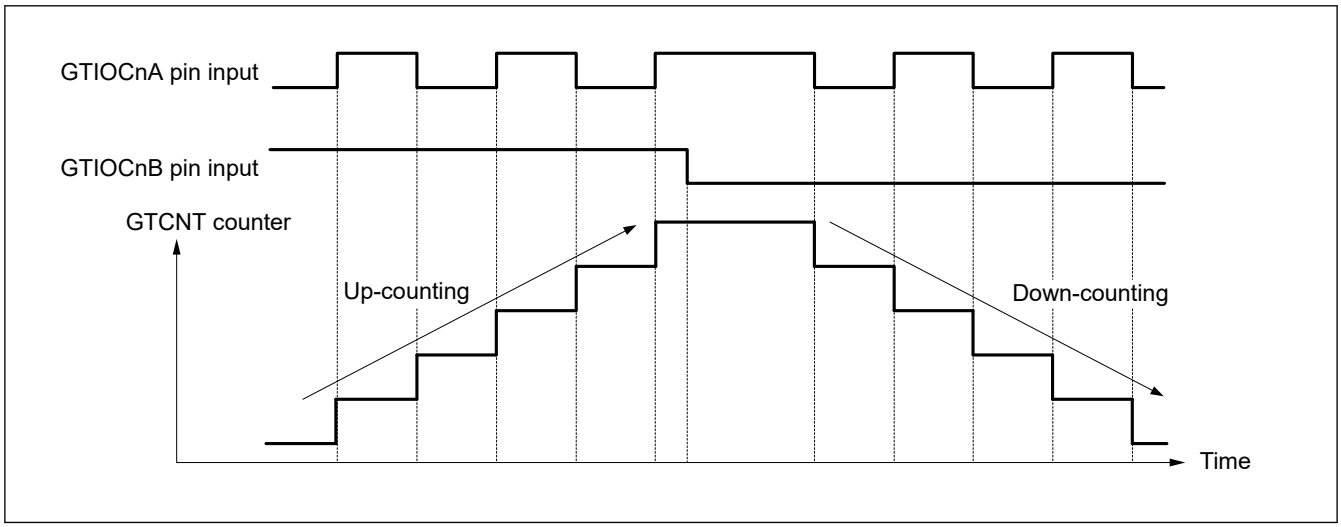


Figure 20.126 Example of phase counting mode 2 (C)

Table 20.58 Conditions of up-counting/down-counting in phase counting mode 2 (C)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Not counting	GTUPSR = 0x00000A00 GTDNSR = 0x00000500
Low	↓		
↑	Low	Down-counting	
↓	High	Up-counting	
High	↓	Not counting	
Low	↑		
↑	High	Up-counting	
↓	Low	Down-counting	

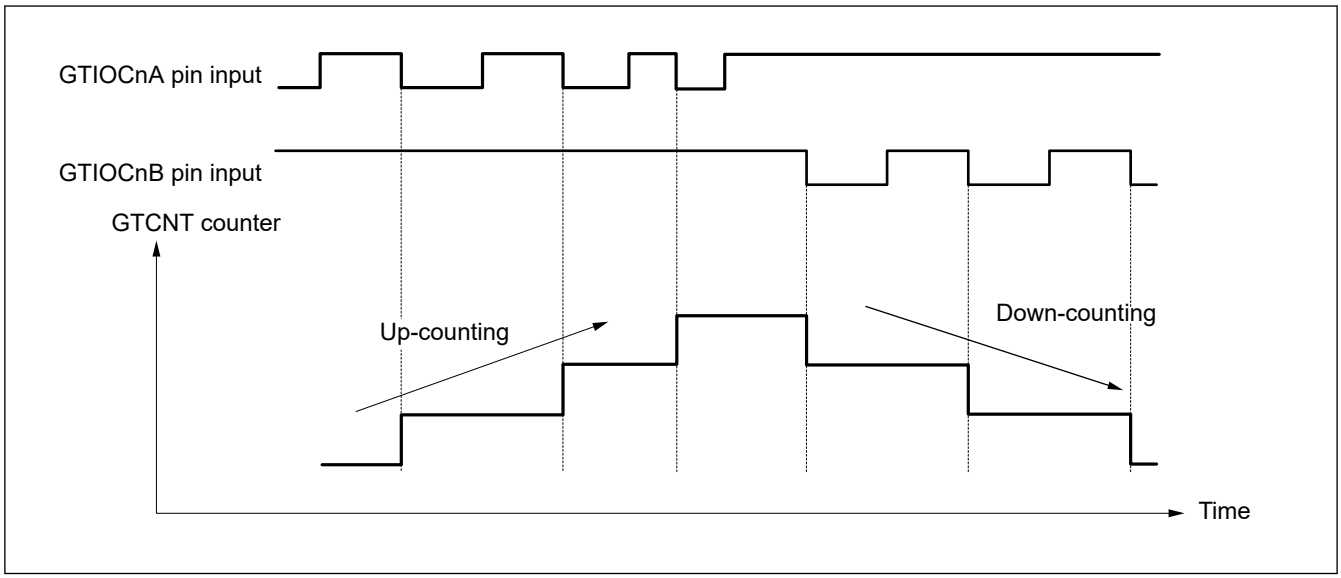


Figure 20.127 Example of phase counting mode 3 (A)

Table 20.59 Conditions of up-counting/down-counting in phase counting mode 3 (A)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Not counting	GTUPSR = 0x00000800 GTDNSR = 0x00008000
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Not counting	
↑	High		
↓	Low		

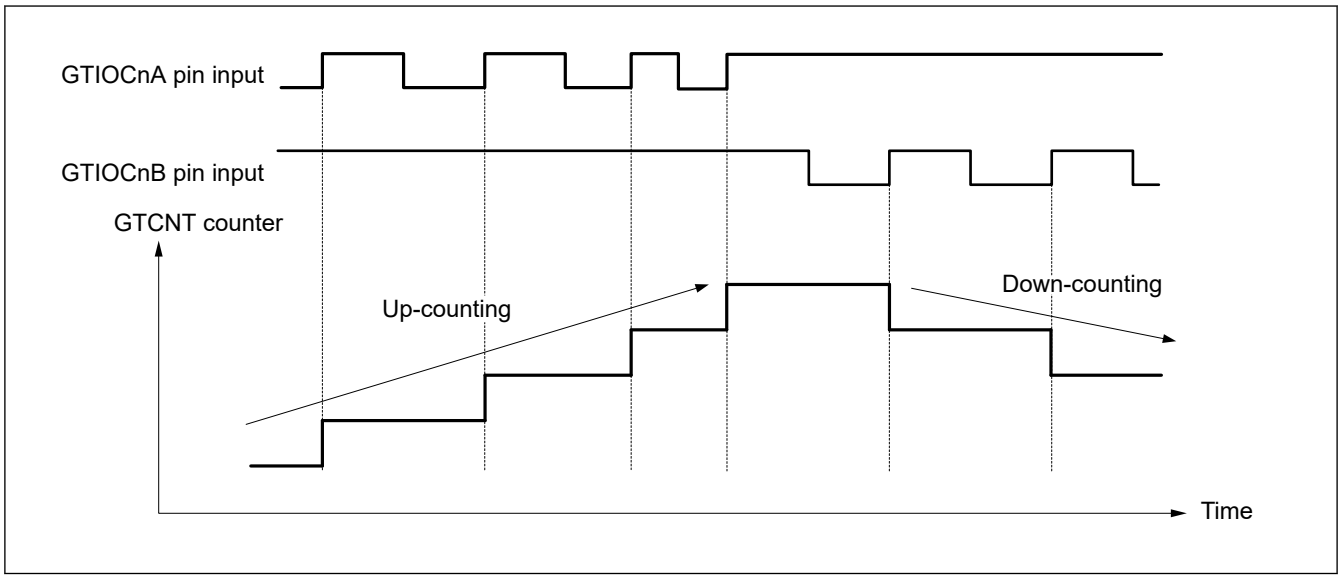


Figure 20.128 Example of phase counting mode 3 (B)

Table 20.60 Conditions of up-counting/down-counting in phase counting mode 3 (B)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Down-counting	GTUPSR = 0x00000200 GTDNSR = 0x00002000
Low	↓	Not counting	
↑	Low		
↓	High		
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Not counting	

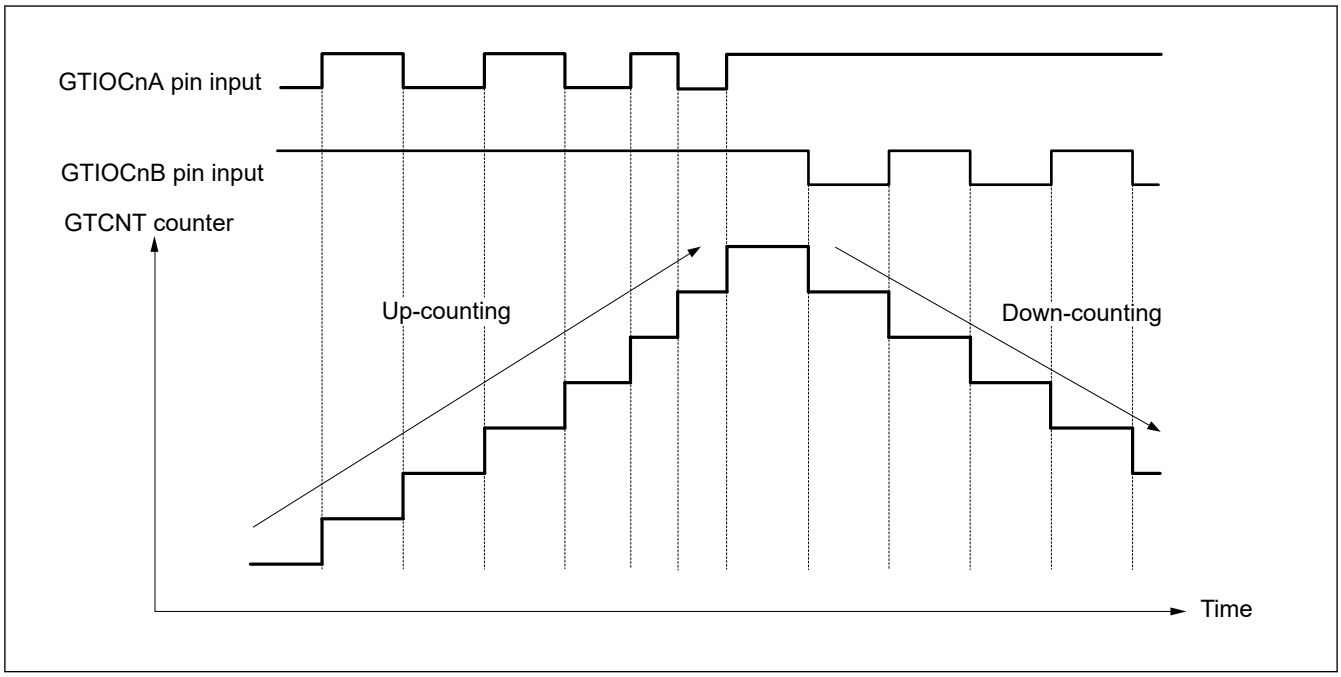


Figure 20.129 Example of phase counting mode 3 (C)

Table 20.61 Conditions of up-counting/down-counting in phase counting mode 3 (C)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Down-counting	GTUPSR = 0x00000A00 GTDNSR = 0x0000A000
Low	↓	Not counting	
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Not counting	
↑	High	Up-counting	
↓	Low	Not counting	

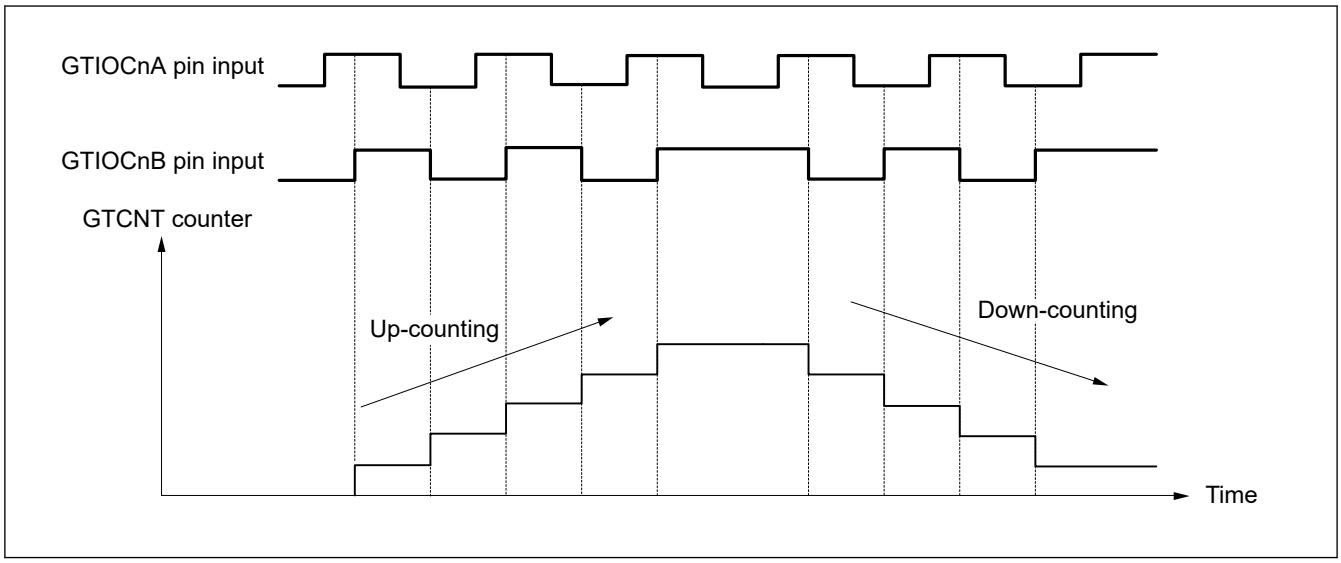


Figure 20.130 Example of phase counting mode 4

Table 20.62 Conditions of up-counting/down-counting in phase counting mode 4

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Up-counting	GTUPSR = 0x00006000 GTDNSR = 0x00009000
Low	↓		
↑	Low	Not counting	
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High	Not counting	
↓	Low		

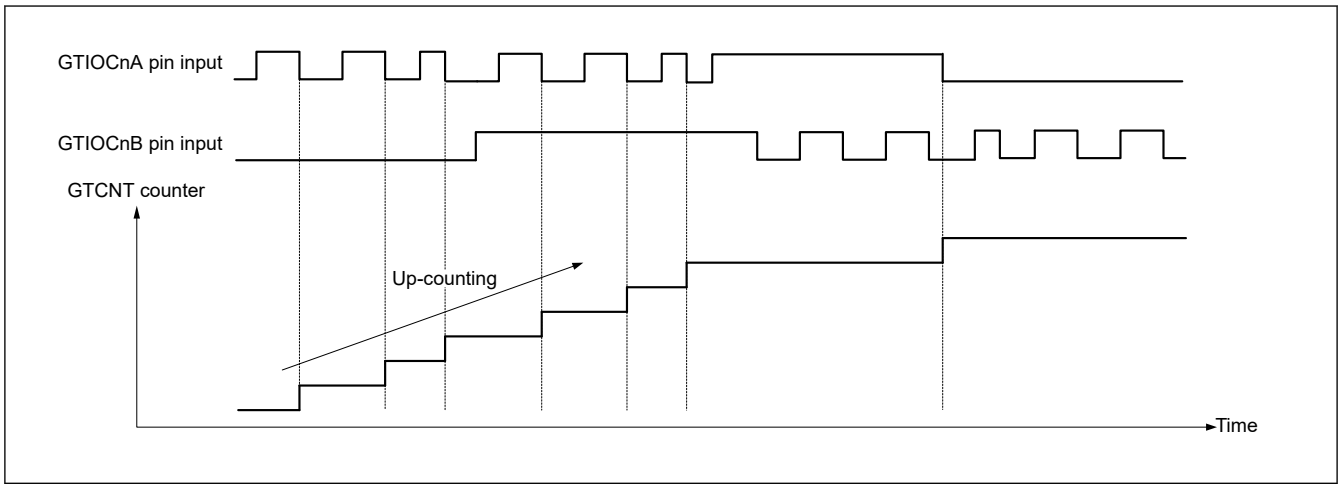


Figure 20.131 Example of phase counting mode 5 (A)

Table 20.63 Conditions of up-counting/down-counting in phase counting mode 5 (A)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000C00 GTDNSR = 0x00000000
Low			
	Low		
	High	Up-counting	
High		Not counting	
Low			
	High		
	Low	Up-counting	

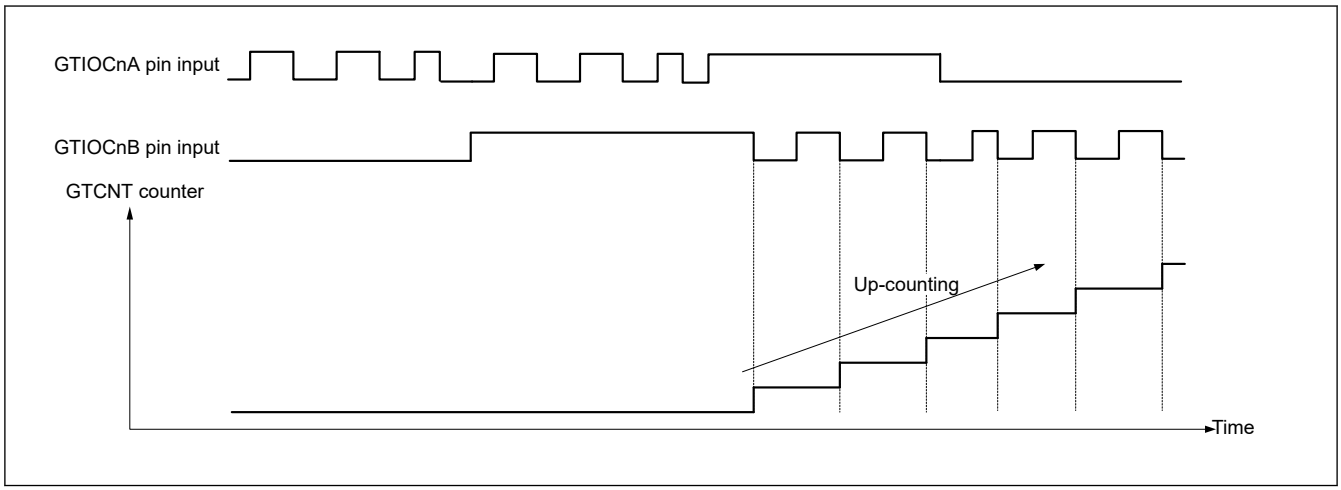


Figure 20.132 Example of phase counting mode 5 (B)

Table 20.64 Conditions of up-counting/down-counting in phase counting mode 5 (B)

: Rising edge

: Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High	↑	Not counting	GTUPSR = 0x0000C000 GTDNSR = 0x00000000
Low	↓	Up-counting	
↑	Low	Not counting	
↓	High		
High	↓	Up-counting	
Low	↑	Not counting	
↑	High		
↓	Low		

20.3.12 External Pulse Width Measuring Function

The pulse width of the GTIOCnA and GTIOCnB pin (n = 3) inputs, and GTETRGA/GTETRGB pin inputs can be measured.

The setting to enable or disable up-counting of the GTCNT counter and the input pin and level which measured pulse width are selected by the USILVL[3:0] bits of the GTUPSR register.

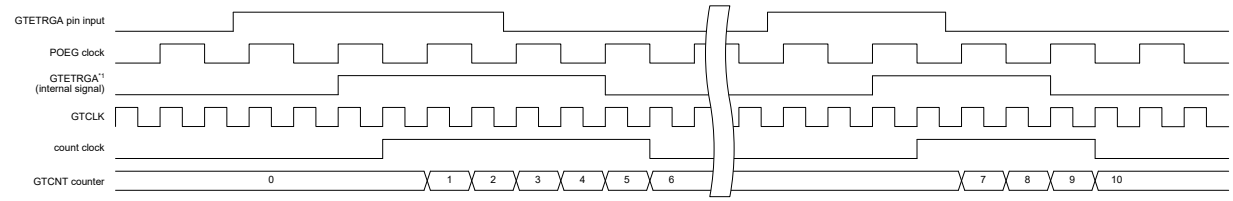
The setting to enable or disable down-counting of the GTCNT counter and the input pin and level which measured pulse width are selected by the DSILVL[3:0] bits of the GTDNSR register.

The setting to enable both up-counting and down-counting of the GTCNT counter at the same time is prohibited.

The counting operation performs periodic counting with the period of the GTPR register.

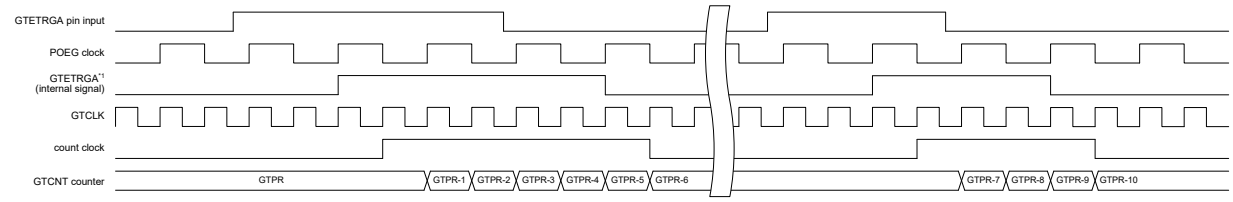
If the phase counting function and the pulse width measuring function are enabled at the same time, the pulse width measuring function does not work and the phase counting function works.

Figure 20.133, Figure 20.134 show examples of external pulse width measuring function and Table 20.65 shows example for setting external pulse width measuring function.



Note: POEG clock: PCLKB
Note 1. The input signal to GPT when the digital noise filter of GTETRGA pin is not used in POEG is shown.

Figure 20.133 Example of external pulse width measuring function (up-counting)



Note: POEG clock: PCLKB
Note 1. The input signal to GPT when the digital noise filter of GTETRGA pin is not used in POEG is shown.

Figure 20.134 Example of external pulse width measuring function (down-counting)

Table 20.65 Example for setting external pulse width measuring function

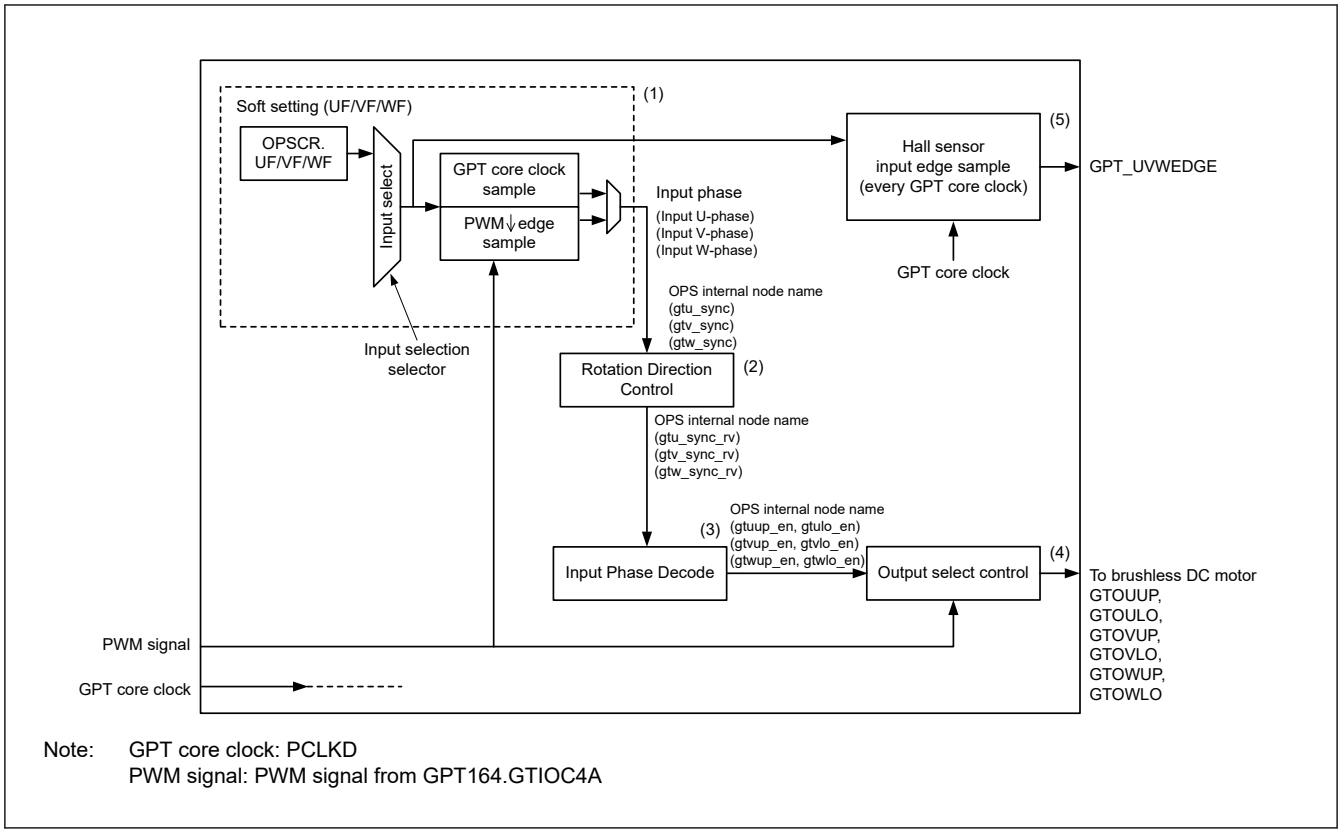
No.	Step Name	Description
1	Set external pulse width measuring function	Enable external pulse width measuring function with the GTUPSR.USILVL[3:0] bits for up-counting operation and the GTDNSR.DSILVL[3:0] bits for down-counting operation and select the input pin and level to measure. In Figure 20.133, GTUPSR.USILVL[3:0] = 1001b (count up when GTETRGA pin is high) In Figure 20.134, GTDNSR.DSILVL[3:0] = 1001b (count down when GTETRGA pin is high)
2	Select count clock	Select the count clock with GTCR.TPCS[3:0] bits of the corresponding channel.
3	Set cycle	Set the cycle in GTPR of the corresponding channel.
4	Set initial value for counter	Set the initial value in the GTCNT counter of the corresponding channel. In Figure 20.133, 0000 0000h is set. In Figure 20.134, the GTPR register value is set.
5	Start count operation	Input a pulse to the input pin to be measured to start count operation.

20.3.13 Output Phase Switching (GPT_OPS)

GPT_OPS provides a function for easy control of brushless DC motor operation using the Output Phase Switching Control Register (OPSCR).

GPT_OPS outputs a PWM signal to be used for chopper control or level signal for each phase (U-positive phase/negative phase, V-positive phase/negative phase, W-positive phase/negative phase) of the 6-phase motor control. This function uses a soft setting value (OPSCR.UF, VF, WF) set by software and a PWM waveform of GPT164.GTIOC4A.

Figure 20.135 shows the conceptual diagram of GPT_OPS control flow.

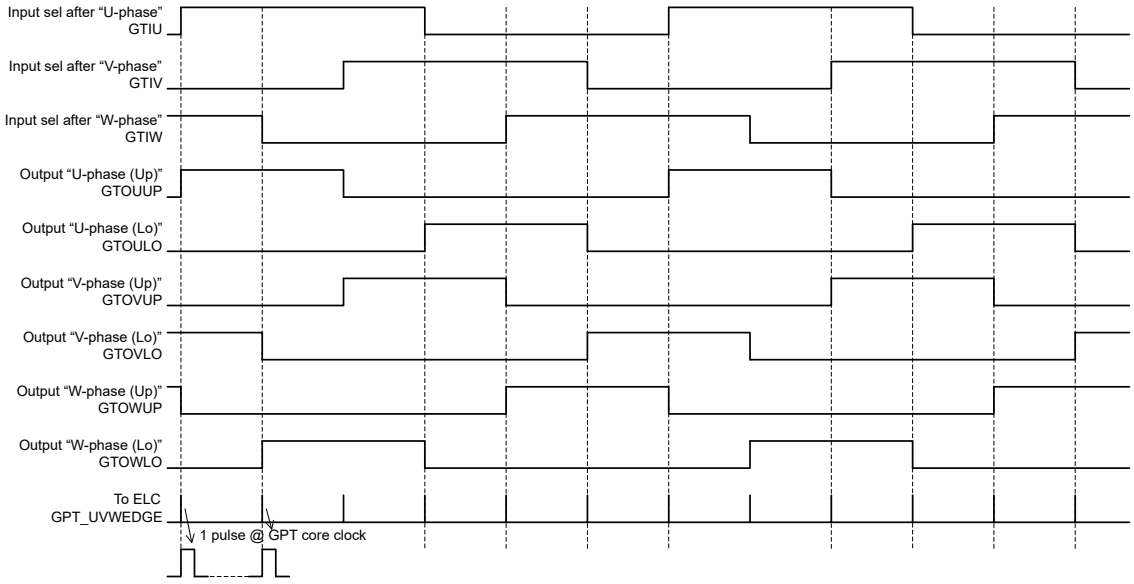


Note: GPT core clock: PCLKD
PWM signal: PWM signal from GPT164.GTIOC4A

Figure 20.135 Conceptual diagram of GPT_OPS control flow

Figure 20.136 shows a 6-phase level signals output example of a GPT_OPS operation.

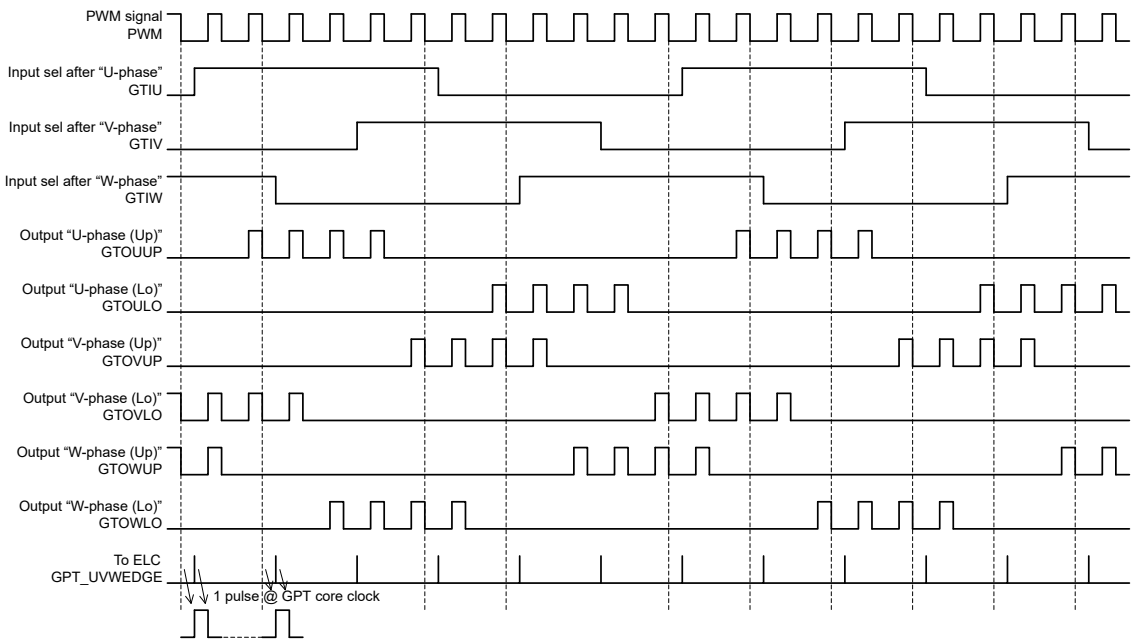
The GPT_UVWEDGE signal in Figure 20.136 is the input phase edge that outputs to the ELC.



Note: Register settings : OPSCR.ALIGN = 0, OPSCR.EN = 1, OPSCR.P = 0, OPSCR.N = 0, OPSCR.INV = 0, OPSCR.RV = 0
Note: GPT core clock: PCLKD

Figure 20.136 Example of 6-phase level output operation

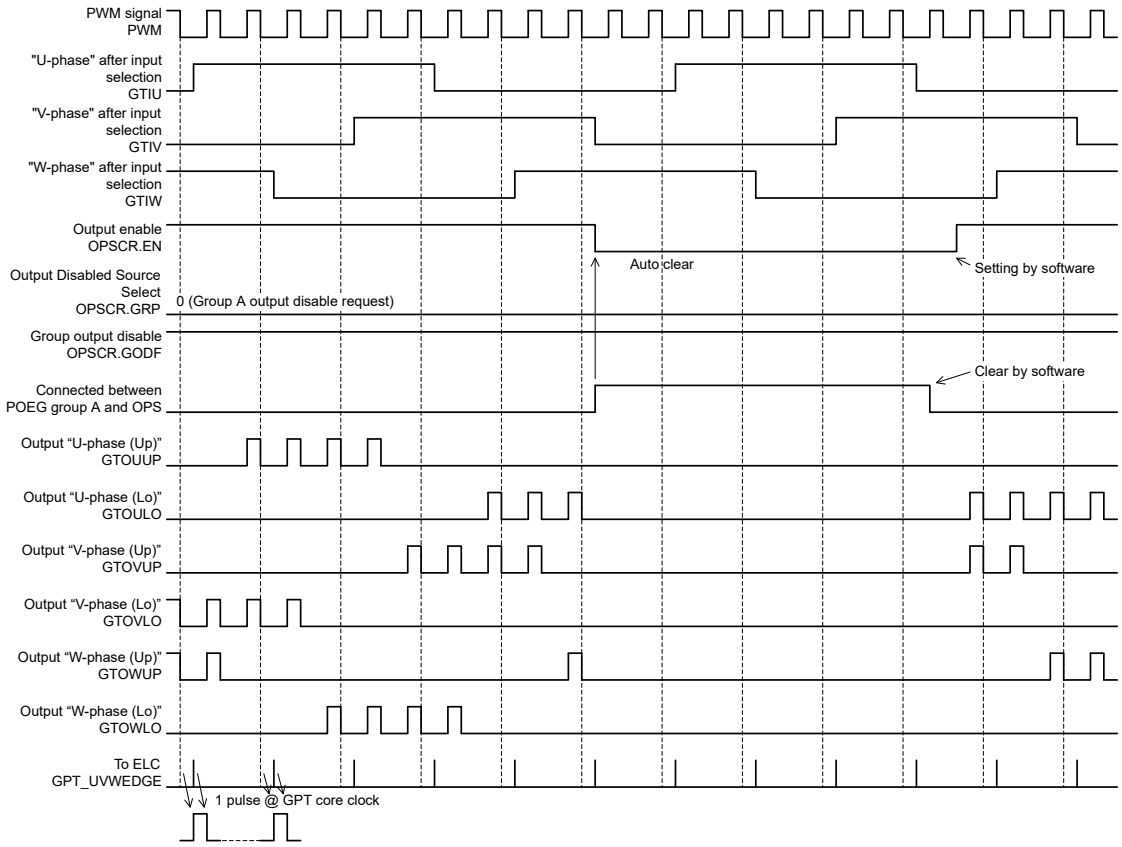
Figure 20.137 shows a 6-phase PWM output example of a GPT_OPS operation with chopper control.



Note: Register settings : OPSCR.ALIGN = 1, OPSCR.EN = 1, OPSCR.P = 1, OPSCR.N = 1, OPSCR.INV = 0, OPSCR.RV = 0
Note: GPT core clock: PCLKD
PWM signal: PWM signal from GPT164.GTIOC4A

Figure 20.137 Example of 6-phase PWM output operation with chopper control

Figure 20.138 shows a 6-phase PWM output example of an output disable control operation.



Note: Register settings : OPSCR.P = 1, OPSCR.N = 1, OPSCR.INV = 0, OPSCR.RV = 0
Note: GPT core clock: PCLKD
PWM signal: PWM signal from GPT164.GTIOC4A

Figure 20.138 Example of group output disable control operation

20.3.13.1 Input Selection and Synchronization of External Input Signal

In the GPT_OPS control flow conceptual diagram shown in Figure 20.135, (1) is a selection of input phase from the software settings by the OPSCR.FB bit.

When OPSCR.FB bit is 1, select the soft setting (OPSCR.UF, VF, WF) with the value of the input phase of PWM (PWM of GPT164.GTIOC4A) using falling edge sampling with OPSCR.ALIGN bit set to 1.

When OPSCR.ALIGN bit is 0, GPT_OPS operates with the input phase of PCLKD synchronization. However, there are cases where the PWM pulse width of the output U/V/W phases (PWM output mode) of switch timing (just before/just after) is shortened.

Table 20.66 shows the input selection process and setting of associated OPSCR bits.

Table 20.66 Input selection processing method

Register OPSCR		Selection of input phase sampling method (U/V/W-phase)	Synchronization input/output selection process (GPT_OPS internal node name)
FB bit	ALIGN bit		
1	1	Software Settings at PWM Falling Edge Sampling (OPSCR.UF, VF, WF of falling edge sample)	Input Phase Input U-Phase (gtu_sync) Input V-Phase (gtv_sync) Input W-Phase (gtw_sync)
	0	Software Setting Value Selection (= OPSCR.UF/VF/WF value) (= PCLKD synchronization)	

20.3.13.2 Rotation Direction Control

In the GPT_OPS control flow conceptual diagram shown in Figure 20.135, (2) controls the direction of rotation of a 3-phase motor using the OPSCR.RV bit.

When the rotation direction is reverse (RV bit = 1), the input phase is inverted.

Table 20.67 shows the assigned output phases based on the OPSCR.RV bit setting (before and after rotation direction control).

Table 20.67 Rotation Direction Control Method

Reversal of Direction of Rotation Using Output Phases as Specified in OPSCR Register	Output of Rotation Direction Control [U/V/W (Positive/Negative)] (GPT_OPS Internal Node Name after Control)					
OPSCR.RV bit	(gtuup_ren)	(gtulo_ren)	(gtvup_ren)	(gtvlo_ren)	(gtwup_ren)	(gtwlo_ren)
0	U-phase (Up) (gtuup_en)	U-phase (Lo) (gtulo_en)	V-phase (Up) (gtvup_en)	V-phase (Lo) (gtvlo_en)	W-phase (Up) (gtwup_en)	W-phase (Lo) (gtwlo_en)
1	U-phase (Up) (gtuup_en)	U-phase (Lo) (gtulo_en)	W-phase (Up) (gtwup_en)	W-phase (Lo) (gtwlo_en)	V-phase (Up) (gtvup_en)	V-phase (Lo) (gtvlo_en)

20.3.13.3 Input Phase Decode

In the GPT_OPS control flow conceptual diagram shown in Figure 20.135, (3) enables the 6-phase signals by decoding the input phase.

Table 20.68 shows the decode table of input phase when OPSCR.RV bit is 0.

Table 20.68 Decode table of input phase (OPSCR.RV = 0)

Input phase (U/V/W) (GPT_OPS internal node name)			6-phase enable {U/V/W (Up/Lo)} by decoding input phase (GPT_OPS internal node name)					
Input U-Phase	Input V-Phase	Input W-Phase	U-phase (Up)	U-phase (Lo)	V-phase (Up)	V-phase (Lo)	W-phase (Up)	W-phase (Lo)
(gtu_sync)	(gtv_sync)	(gtw_sync)	(gtuup_en)	(gtulo_en)	(gtvup_en)	(gtvlo_en)	(gtwup_en)	(gtwlo_en)
1	0	1	1	0	0	1	0	0
1	0	0	1	0	0	0	0	1
1	1	0	0	0	1	0	0	1
0	1	0	0	1	1	0	0	0
0	1	1	0	1	0	0	1	0
0	0	1	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0

Table 20.69 Decode table of input phase (OPSCR.RV = 1) (1 of 2)

Input phase (U/V/W) (GPT_OPS internal node name)			6-phase enable {U/V/W (Up/Lo)} by decoding input phase (GPT_OPS internal node name)					
Input U-Phase	Input V-Phase	Input W-Phase	U-phase (Up)	U-phase (Lo)	V-phase (Up)	V-phase (Lo)	W-phase (Up)	W-phase (Lo)
(gtu_sync)	(gtv_sync)	(gtw_sync)	(gtuup_en)	(gtulo_en)	(gtvup_en)	(gtvlo_en)	(gtwup_en)	(gtwlo_en)
1	0	1	0	1	1	0	0	0
1	0	0	0	1	0	0	1	0
1	1	0	0	0	0	1	1	0
0	1	0	1	0	0	1	0	0

Table 20.69 Decode table of input phase (OPSCR.RV = 1) (2 of 2)

Input phase (U/V/W) (GPT_OPS internal node name)			6-phase enable {U/V/W (Up/Lo)} by decoding input phase (GPT_OPS internal node name)					
Input U-Phase	Input V-Phase	Input W-Phase	U-phase (Up)	U-phase (Lo)	V-phase (Up)	V-phase (Lo)	W-phase (Up)	W-phase (Lo)
(gtu_sync)	(gtv_sync)	(gtw_sync)	(gtuup_en)	(gtulo_en)	(gtvup_en)	(gtvlo_en)	(gtwup_en)	(gtwlo_en)
0	1	1	1	0	0	0	0	1
0	0	1	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0

20.3.13.4 Output Selection Control

In the GPT_OPS control flow conceptual diagram in Figure 20.135, (4) represents the selection of the output waveform by setting the OPSCR register bit.

For output selection, the following bits are relevant:

- The OPSCR.EN bit controls whether to output the 6-phase output, or to stop
- The OPSCR.P and OPSCR.N bits can select from the level signal or PWM signal (chopper output) for the output phase
- The polarity of the output phase can be set to a positive logic or negative logic by the OPSCR. INV bit.

Table 20.70 and Table 20.71 show the output selection control method using the OPSCR register bit.

Table 20.70 Output selection control method (positive phase)

Enable-phase output control	Positive-phase output (P) control	Invert-phase output control	Output port name (positive phase = up) (output selection internal node allocation)	
OPSCR.EN	OPSCR.P	OPSCR.INV	GTOUUP GTOVUP GTOWUP	Mode
0	x	x	0	Output Stop (External pin: Hi-Z) GPT_OPS → 0 output
1	0	0	Level signal (gtuup_ren) (gtvup_ren) (gtwup_ren)	Level Output Mode (Positive phase) (Positive logic)
1	0	1	Level signal (~gtuup_ren) (~gtvup_ren) (~gtwup_ren)	Level Output Mode (Positive phase) (Negative logic)
1	1	0	PWM signal (PWM & gtuup_ren) (PWM & gtvup_ren) (PWM & gtwup_ren)	PWM Output Mode (Positive phase) (Positive logic)
1	1	1	PWM signal (~(PWM & gtuup_ren)) (~(PWM & gtvup_ren)) (~(PWM & gtwup_ren))	PWM Output Mode (Positive phase) (Negative logic)

Table 20.71 Output selection control method (negative phase)

Enable-phase output control	Positive-phase output (N) control	Invert-phase output control	Output port name (negative phase = Lo) (output selection internal node allocation)	
OPSCR.EN	OPSCR.N	OPSCR.INV	GTOULO GTOVLO GTOWLO	Mode
0	x	x	0	Output Stop (External pin: Hi-Z) GPT_OPS → 0 output
1	0	0	Level signal (gtulo_ren) (gtvlo_ren) (gtwlo_ren)	Level Output Mode (Negative phase) (Positive logic)
1	0	1	Level signal (~gtulo_ren) (~gtvlo_ren) (~gtwlo_ren)	Level Output Mode (Negative phase) (Negative logic)
1	1	0	PWM signal (PWM & gtulo_ren) (PWM & gtvlo_ren) (PWM & gtwlo_ren)	PWM Output Mode (Negative phase) (Positive logic)
1	1	1	PWM signal (~(PWM & gtulo_ren)) (~(PWM & gtvlo_ren)) (~(PWM & gtwlo_ren))	PWM Output Mode (Negative phase) (Negative logic)

20.3.13.5 Output Selection Control (Group Output Disable Function)

When OPSCR.GODF is 1 and the signal value selected by the OPSCR.GRP bit is high (output disable request), the group output-disable function asynchronously sets the output to Hi-Z. When an output-disable request is generated, the OPSCR.EN bit is cleared to 0. For the return, set the OPSCR.EN bit to 1 after clearing the output disable request by software.

To ensure output-disable control, use the POEG_GROUPn (n = A, B) interrupt to clear the flag in the POE or check that the OPSCR.EN bit is 0 and then clear the flag. For an example of the operation for group output disable control, see Figure 20.138.

20.3.13.6 Event Link Controller (ELC) Output

In the GPT_OPS control flow conceptual diagram shown in Figure 20.135, (5) outputs the UVW input signal edge to the ELC.

The UVW input edge signal is the logical OR of the rising and falling edge signals of each U-phase/V-phase/W-phase input sampled at PCLKD. That is, if the high period of each of the U-phase/V-phase/W-phase of the input phase is short in duration, the Hall sensor edge input signal is not output at that time.

When OPSCR.FB bit is 1, the UVW input edge signal is the logical OR of the edge of the soft setting (OPSCR.UF, VF, WF) sampled at PCLKD.

See Figure 20.136 to Figure 20.138 for examples of the output signal to the ELC.

20.3.13.7 GPT_OPS Start Operation Setting Flow

Table 20.72 Example setting of GPT_OPS start operation (1 of 2)

No.	Step Name	Description
1	GPT164 operation mode setting	GPT164.GTIOC4A set the PWM output operation mode of the saw-wave or triangle-wave. For details, see section 20.3.3. PWM Output Operating Mode.
2	Counting of GPT164	Start the count operation of GPT164, and outputs a PWM waveform.
3	GPT_OPS input data set	Set software setting to OPSCR.UF, VF, and WF bits.

Table 20.72 Example setting of GPT_OPS start operation (2 of 2)

No.	Step Name	Description
4	GPT_OPS input phase enable setting/input phase alignment setting	Setting the OPSCR.FB = 1 to enable software setting for the input phase. Select the alignment of the input phase by OPSCR.ALIGN bit.
5	Setting the GPT_OPS output phase	Set the level output/PWM output of the positive/negative phase output by OPSCR.P/OPSCR.N bit. Set the positive logic/negative logic of the output phase by OPSCR.INV bit. Set the rotation direction by OPSCR.RV bit
6	GPT_OPS setting the group output disable function	Set the selection of output disable source by OPSCR.GRP bit. Perform the setting of on/off of the group output disable function by OPSCR.GODF bit.
7	GPT_OPS Working	Setting the OPSCR.EN = 1 outputs the 6-phase output to drive the brushless DC motor from the GPT_OPS.

20.3.14 GTIOCnA/B Input/Output Pin Polarity Reversal Function

The polarity of the GTIOCnA, GTIOCnB input pins and the GTIOCnA, GTIOCnB output pins can be inverted according to the polarity of the external device.

When GTCR.AINV is 1, the input and output of GTIOCnA are inverted.

When GTCR.BINV is 1, the input and output of GTIOCnB are inverted.

GTIOCnA and GTIOCnB are inverted at the final output stage immediately after input. When GTIOCnA and GTIOCnB are used as count start/stop/clear factors, countdown/up factors, and input capture factors, or when low/high/toggle output is used as PWM output, external input and external output are reversed. It is necessary to make settings that take this into consideration.

Set GTCR.AINV and GTCR.BINV after setting the GTIOCnm pin functions during initial settings.

20.4 Interrupt Sources

20.4.1 Interrupt Sources

The GPT provides the following interrupt sources:

- GTCCR input capture/compare match
- GTADTR compare match
- GTCNT counter overflow (GTPR compare match)/underflow.
- period count function finish

Each interrupt source has its own status flag and control bit for generating interrupt request signals, where the generation of interrupt request signals can be enabled or disabled individually. When an interrupt source signal is generated, the associated status flag in GTST is set to 1, and an interrupt request is generated if the corresponding interrupt request enable or disable bit in the GTINTAD register is 1. The associated status flag in GTST can be cleared by writing 0. If flag set and flag clear occur at the same time, flag clear takes priority over flag set. These flags are automatically updated by the internal state.
The Interrupt Controller Unit can change the relative channel priorities. However, the priority within a channel is fixed. For details, see section 12, Interrupt Controller Unit (ICU).

Table 20.73 lists the GPT interrupt sources.

Table 20.73 Interrupt sources

Channel	Name	Interrupt source	Interrupt flag	DTC activation
n = 0 to 3	GPTn_CCMPA	GPT16n.GTCCRA input capture/compare match	GTST[0] (TCFA)	Possible
	GPTn_CCMPB	GPT16n.GTCCRB input capture/compare match	GTST[1] (TCFB)	Possible
	GPTn_CMPC	GPT16n.GTCCRC compare match	GTST[2] (TCFC)	Possible
	GPTn_CMPD	GPT16n.GTCCRD compare match	GTST[3] (TCFD)	Possible
	GPTn_CMPE	GPT16n.GTCCRE compare match	GTST[4] (TCFE)	Possible
	GPTn_CMPF	GPT16n.GTCCRF compare match	GTST[5] (TCFF)	Possible
	GPTn_OVF	GPT16n.GTCNT overflow (GPT16n.GTPR compare match)	GTST[6] (TCFPO)	Possible
	GPTn_UDF	GPT16n.GTCNT underflow	GTST[7] (TCFPU)	Possible
	GPTn_ADTRGA	GPT16n.GTADTRA compare match	GTST[17:16] (ADTRADF, ADTRAUF)	Possible
	GPTn_ADTRGB	GPT16n.GTADTRB compare match	GTST[19:18] (ADTRBDF, ADTRBUF)	Possible
	GPTn_PC	Period count function finish	GTST[31] (PCF)	Possible

(1) GPTn_CCMPA interrupt (n = 0 to 3)

An interrupt request is generated under the following conditions:

- When the GTCCRA register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRA register
- When the GTCCRA register functions as an input capture register, the input-capture signal causes transfer of the GTCNT counter value to the GTCCRA register. In complementary PWM mode, GTCCRA register does not function as an input capture register.

(2) GPTn_CCMPB interrupt (n = 0 to 3)

An interrupt request is generated under the following conditions:

- When the GTCCRB register functions as a compare match register, the GTCNT counter value matches with the GTCCRB register. In complementary PWM mode, GTCCRB register does not function as a compare match register.
- When the GTCCRB register functions as an input capture register, the input-capture signal causes transfer of the GTCNT counter value to the GTCCRB register. In complementary PWM mode, GTCCRB register does not function as an input capture register.

(3) GPTn_CMPC interrupt (n = 0 to 3)

An interrupt request is generated under the following condition:

- When the GTCCRC register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, GTCNT counter value of master channel) matches with the GTCCRC register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[3:0] = 0001b (saw-wave one-shot pulse mode)
- GTCR.MD[3:0] = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (buffer operation with the GTCCRC register).

(4) GPTn_CMPD interrupt (n = 0 to 3)

An interrupt request is generated under the following condition:

- When the GTCCRD register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRD register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[3:0] = 0001b (saw-wave one-shot pulse mode)
- GTCR.MD[3:0] = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 10b, 11b (buffer operation with the GTCCRD register).

(5) GPTn_CMPE interrupt (n = 0 to 3)

An interrupt request is generated under the following condition:

- When the GTCCRE register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRE register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[3:0] = 0001b (saw-wave one-shot pulse mode)
- GTCR.MD[3:0] = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (buffer operation with the GTCCRE register).

(6) GPTn_CMPF interrupt (n = 0 to 3)

An interrupt request is generated under the following condition:

- When the GTCCRF register functions as a compare match register, the GTCNT counter value (in complementary PWM mode, the GTCNT counter value of master channel) matches with the GTCCRF register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[3:0] = 0001b (saw-wave one-shot pulse mode)
- GTCR.MD[3:0] = 0110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 10b, 11b (buffer operation with the GTCCRF register).

(7) GPTn_OVF interrupt (n = 0 to 3)

An interrupt request is generated in the following conditions:

- In saw-wave PWM mode 1 and saw-wave one-shot pulse mode, interrupt requests are enabled at overflows (when the GTCNT counter value changes from GTPR to 0 during up-counting)
- In saw-wave PWM mode 2, interrupt requests are enabled at overflows (GTCNT counter value changes from GTCCRm (m = A to F) register value selected with GTCSR.CSCMSC[2:0] bits to 0) or the time when GTCNT counter value matches GTPR register value.
- In triangle-wave mode, interrupt requests are enabled at crests (the GTCNT changes from GTPR to GTPR-1)
- In complementary PWM mode, interrupt requests are enabled at crests (GTCNT counter value of master channel changes from GTPR register value to GTPR register value minus 1).
- In counting by hardware sources (include external pulse width measuring function), an overflow (GTCNT changes from GTPR to 0 in up count) has occurred.

(8) GPTn_UDF interrupt (n = 0 to 3)

An interrupt request is generated in the following conditions.

- In saw-wave PWM mode 1 and saw-wave one-shot pulse mode, interrupt requests are enabled at underflows (when the GTCNT counter value changes from 0 to GTPR during down-counting)
- In triangle-wave mode, interrupt requests are enabled at troughs (the GTCNT changes from 0 to 1)
- In complementary PWM mode, interrupt requests are enabled at troughs (GTCNT counter value of master channel changes from 0 to 1).
- In counting by hardware sources (include external pulse width measuring function), underflow (GTCNT changes from 0 to GTPR in down count) has occurred.

About Interrupt signals and interrupt status flags, see section 20.2.16. GTST : General PWM Timer Status Register.

(9) GPTn_ADTRGA interrupt (n = 0 to 3)

When the GTCNT counter value matches with GTADTRA, an interrupt request is generated under the following condition.

- In Up-counting, the interrupt enable bit (ADTRAUEN) in GTINTAD is 1.
- In Down-counting, the interrupt enable bit (ADTRADEN) in GTINTAD is 1.
  In event count operation performing, this interrupt request isn't generated.

(10) GPTn_ADTRGB interrupt (n = 0 to 3)

When the GTCNT counter value matches with GTADTRB, an interrupt is generated under the following condition.

- In Up-counting, the interrupt enable bit (ADTRBUEN) in GTINTAD is 1.
- In Down-counting, the interrupt enable bit (ADTRBDEN) in GTINTAD is 1.
  In event count operation performing, this interrupt request isn't generated.

(11) GPTn_PC Interrupt (n = 0 to 3)

When the GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 1, an interrupt request is generated at the end of cycle.

20.4.2 DTC Activation

The DTC can be activated by the interrupt in each channel. For details, see section 12, Interrupt Controller Unit (ICU) and section 15, Data Transfer Controller (DTC).

20.4.3 Interrupt and A/D Conversion Start Request Skipping Function

20.4.3.1 Interrupt Skipping Function by GTITC Register

By setting the GTITC register, the GTCNT counter overflow (GTPR register compare match) interrupt (GPTn_OVF) and underflow interrupt (GPTn_UDF) can be skipped. Other interrupts and A/D conversion start request signals can be skipped in coordination with the GPTn_OVF/GPTn_UDF skipping function. When the interrupt is skipped, the updating of relevant status flag is also skipped. The interrupt skipping is continued even if the status flag is set to 1.

The interrupt skipping function is related only to the GTITC register setting, and is not related to setting of the GTINTAD register interrupt enable bit. The interrupt skipping is continued even if the interrupt is disabled with GTINTAD register setting.

When both troughs and crests are counted and skipped in triangle-wave mode, if the number of times of skipping is odd, GPTn_OVF/GPTn_UDF interrupt requests cannot be generated at troughs only or at crests only depending on the skipping counter start timing. Therefore, in order to count both troughs and crests and generate the GPTn_OVF/GPTn_UDF interrupts at troughs only or crests only in triangle-wave mode, the number of times of skipping should be even.

Similarly, in saw-wave mode, when both overflows and underflows are counted and skipped with the count direction changed, GPTn_OVF/GPTn_UDF interrupt requests are sometimes not generated at overflows only or at underflows only. Therefore, in order to count both overflows and underflows with the count direction changed and generate the GPTn_OVF/GPTn_UDF interrupts at overflows only or underflows only in saw-wave mode, the skipping state should be carefully checked before using.

When changing the skipping count, be sure to release the skipping count setting (GTITC.IVTC[1:0] bits = 00b).

Figure 20.139 to Figure 20.144 show examples of skipping function operation.

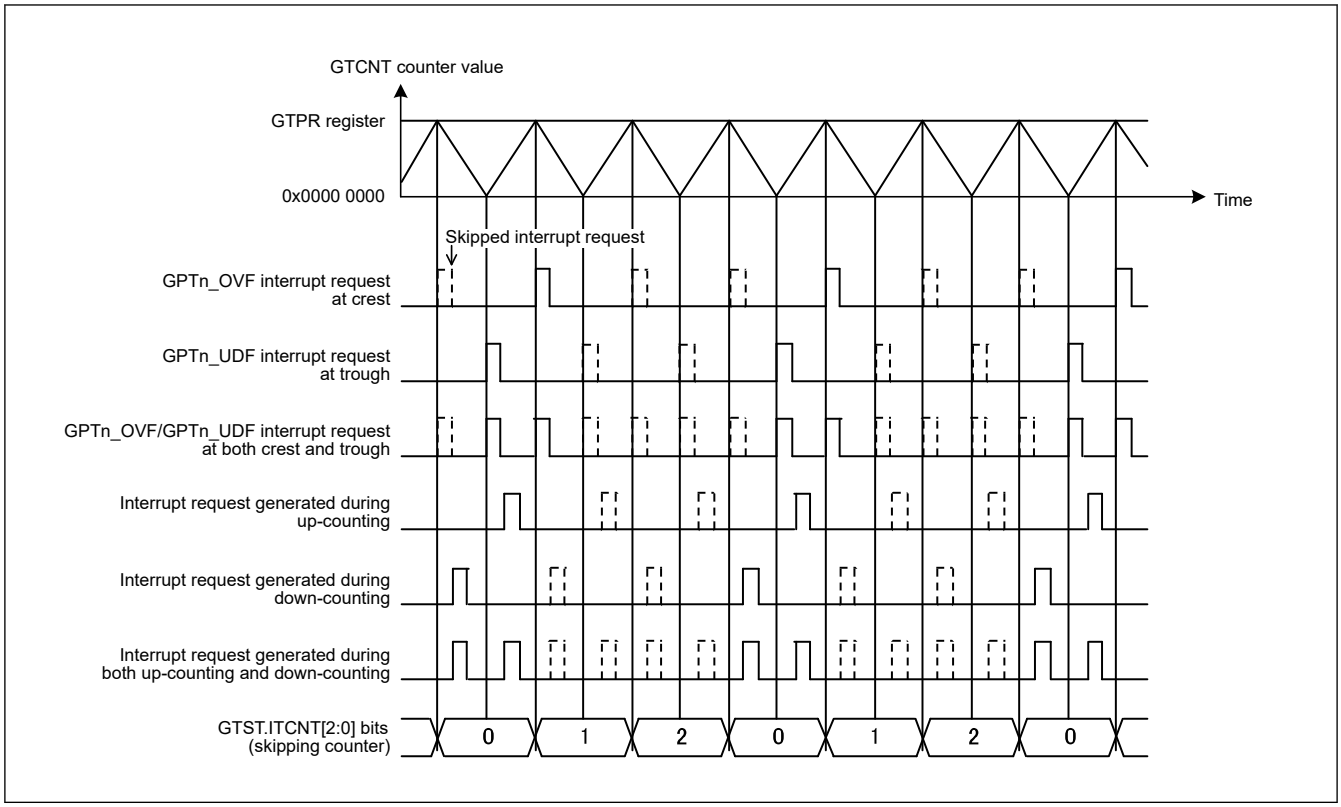


Figure 20.139 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Crests, Skipping Count: 2)

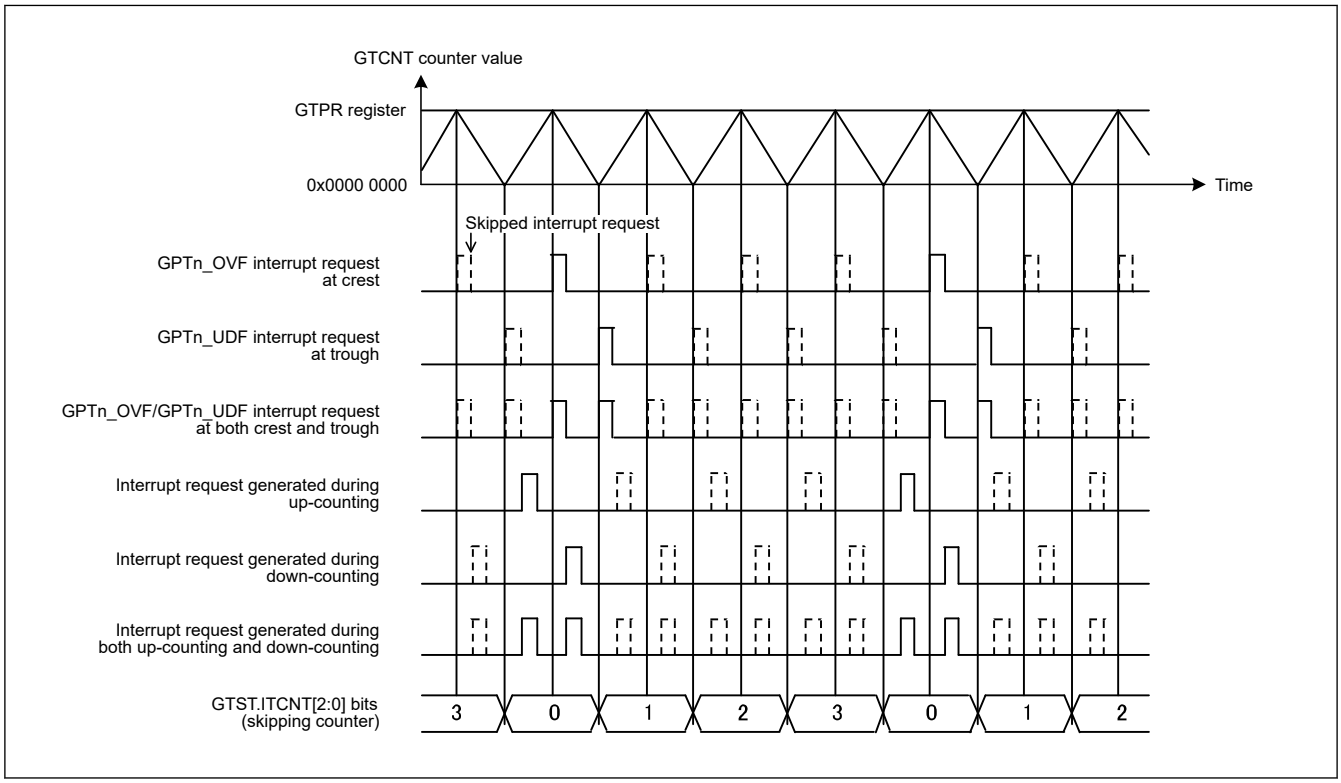


Figure 20.140 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Troughs, Skipping Count: 3)

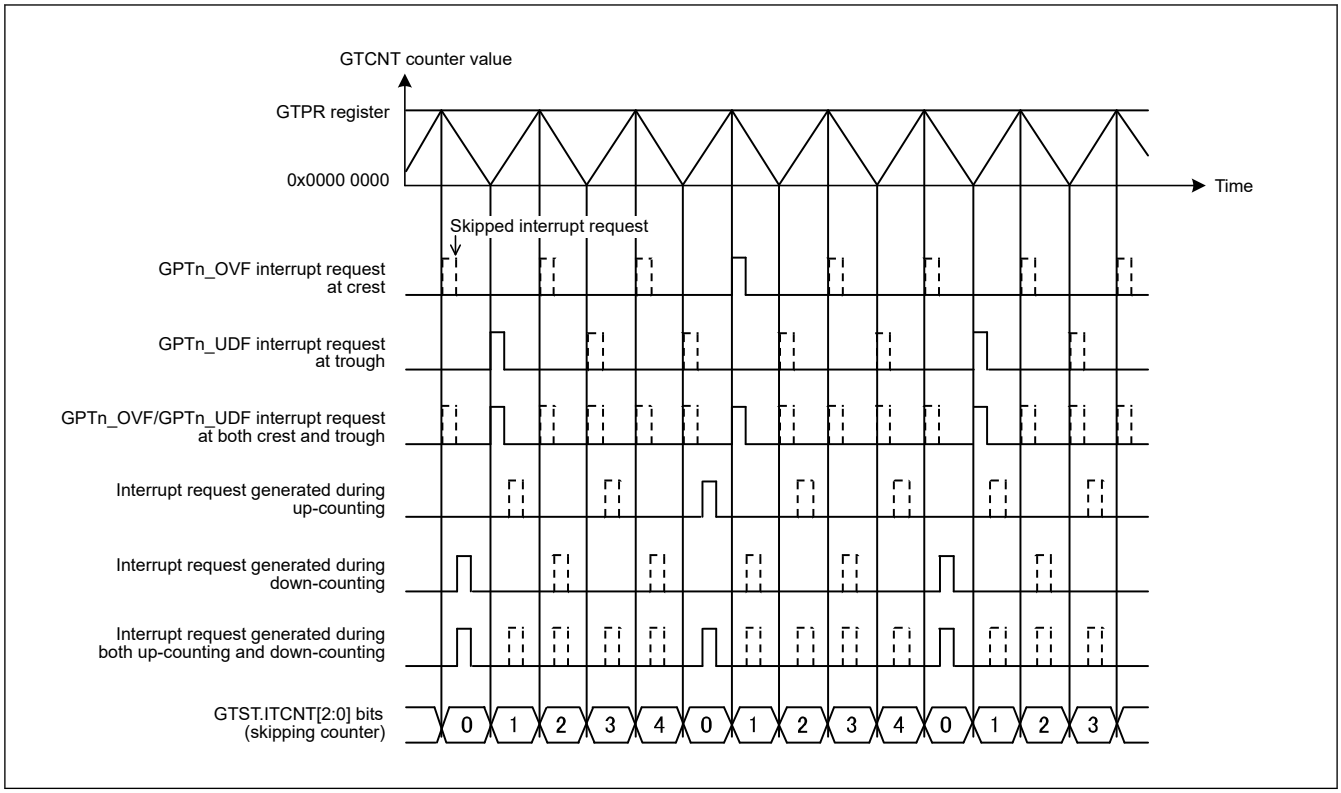


Figure 20.141 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 4)

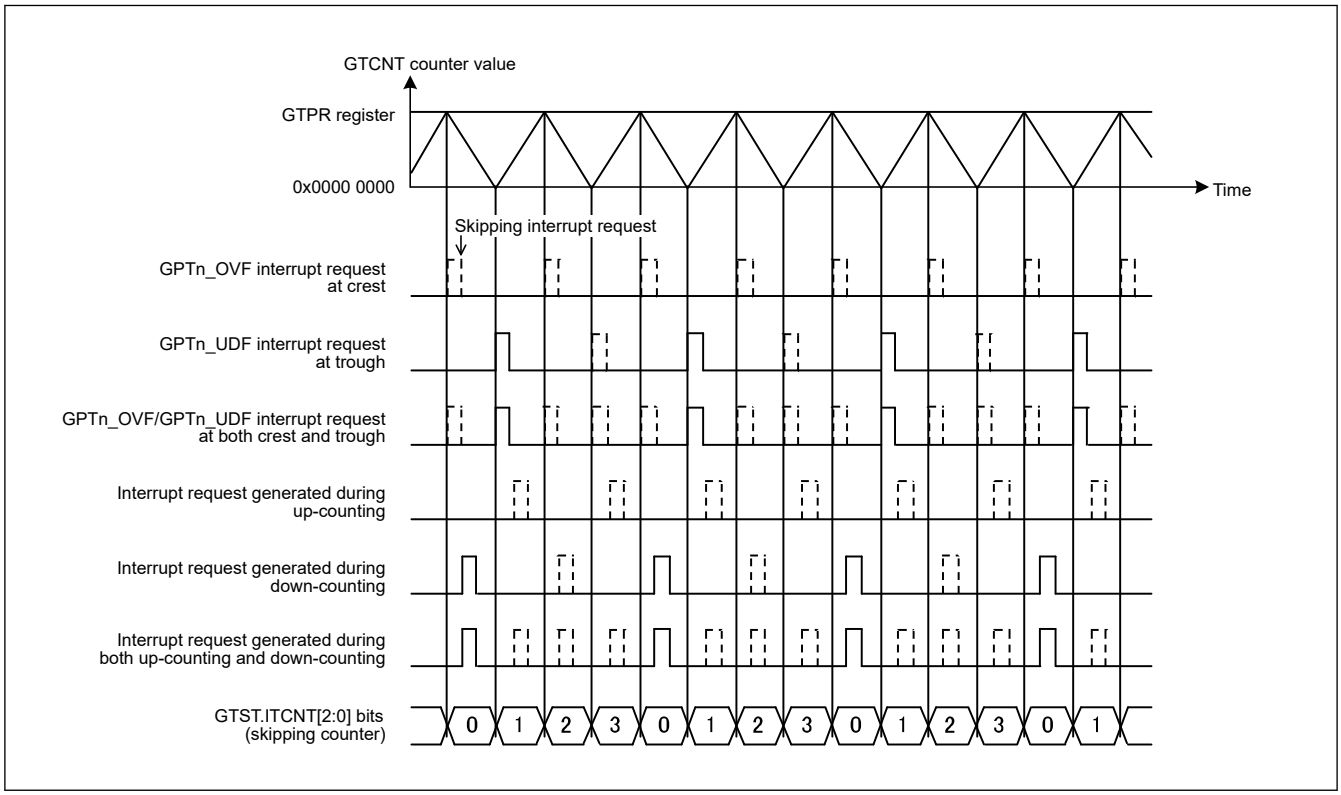


Figure 20.142 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 3, Skipping Started at Up-Counting)

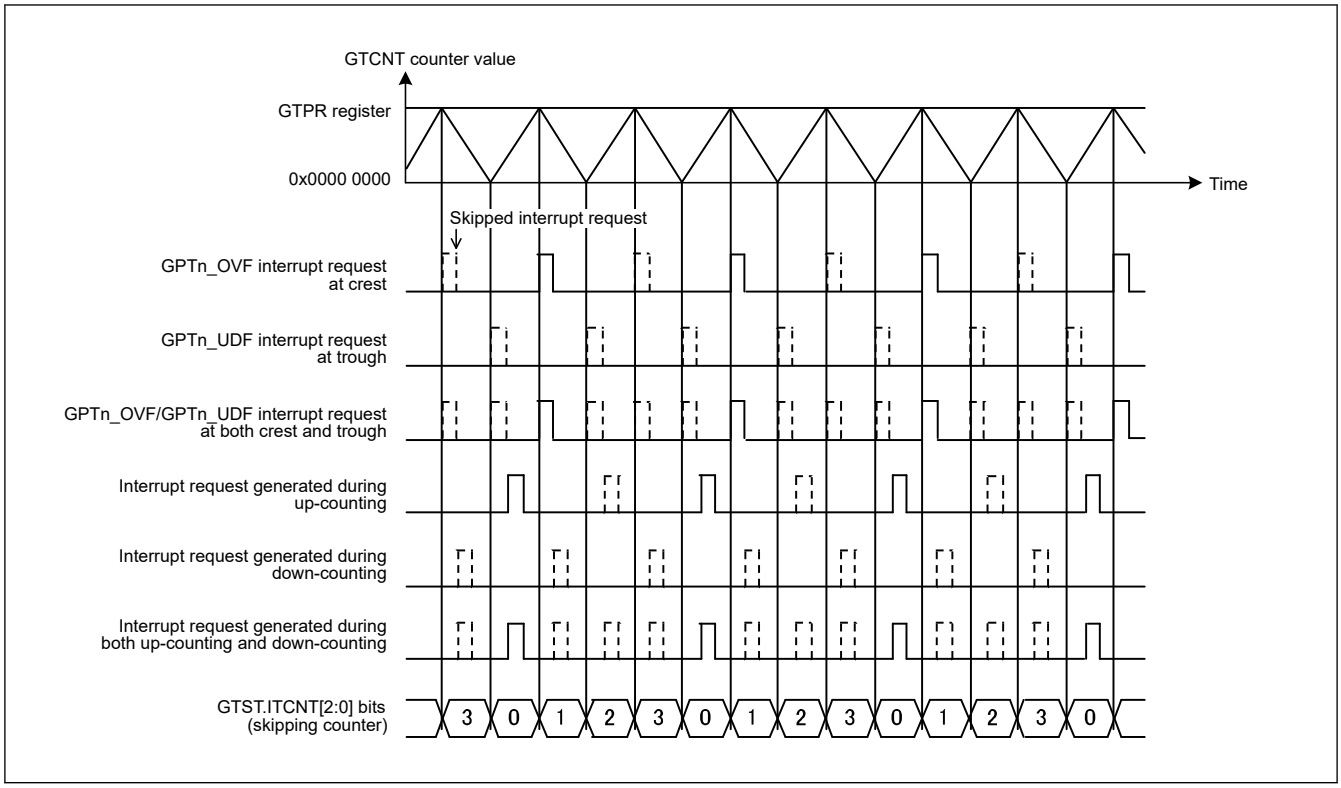


Figure 20.143 Example of Interrupt Skipping Function Operation (Triangle Waves, Counting and Skipping Both Troughs and Crests, Skipping Count: 3, Skipping Started at Down-Counting)

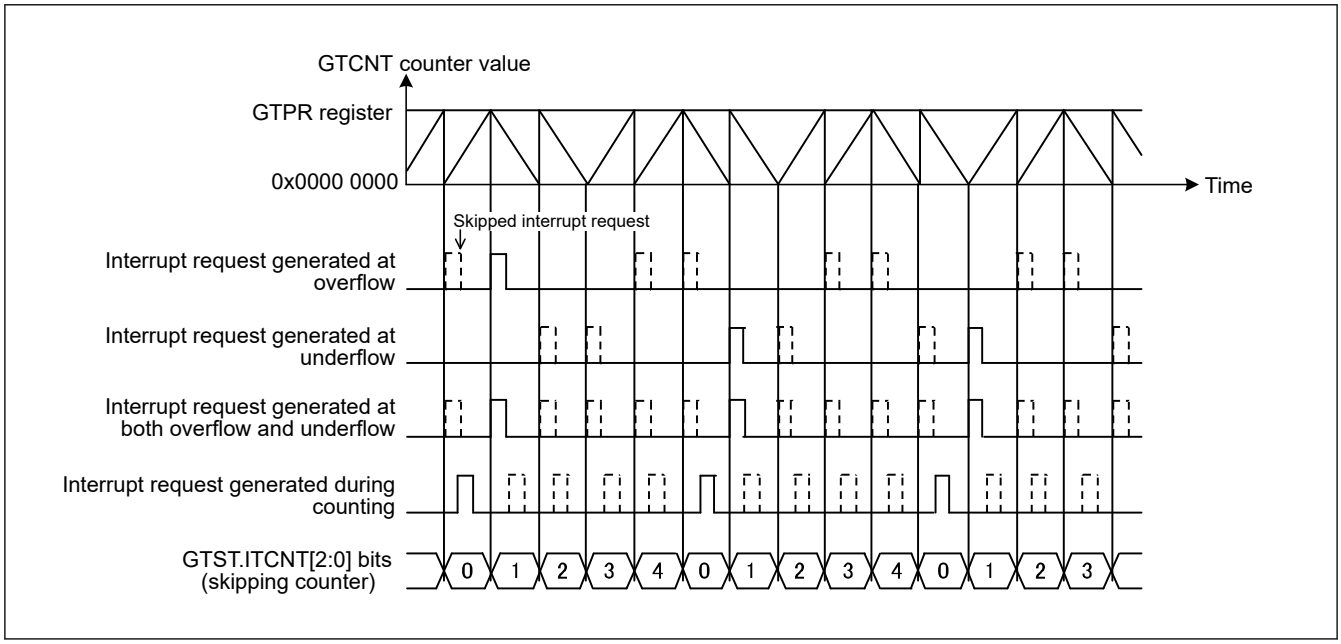


Figure 20.144 Example of Interrupt Skipping Function Operation (Saw Waves, Operation with Count Direction Changed, Counting and Skipping Both Overflows and Underflows, Skipping Count: 4)

20.4.3.2 Extended Interrupt Skipping Function

Overflow/underflow interrupt, compare match/input capture interrupt, A/D conversion start request, and buffer transfer can be skipped by counting the GTCNT counter overflow or underflow based on settings of the GTEITC, GTEITLI1, GTEITLI2 and GTEITLB registers.

Overflow/underflow interrupt, compare match/input capture interrupt, A/D conversion start request, and buffer transfer can individually be set for skipping and the skipping period using the GTEITLI1, GTEITLI2, and GTEITLB registers.

The skipping period is set, related to the operation of the two independent extended interrupt skipping counters (EITCNT1[3:0] and EITCNT2[3:0] bits in the GTEITC registers), as a period that either of the counter value for such skipping counters is other than 0 or other than the skipping count. The skipping period can also be set as a period that the both skipping counters are set as other than 0 or other than the skipping count for the counter value.

Figure 20.145 shows the counter operation for interrupt skipping by the GTITC register and extended interrupt skipping.

The counter operation for extended interrupt skipping is set by the GTEITC register.

The EITCNT1[3:0] bits are set for 0 as initial value and then they count the count source (in Figure 20.145, a crest is selected) selected by the extended interrupt counter 1 count source select bit (GTEITC.EIVTC1[1:0] bits), and repeat the count operation which returns to 0 when the skipping count (in Figure 20.145, count is 2) set by the extended interrupt skipping 1 skipping count setting bit (EIVTT1[3:0] bits) is achieved.

The EITCNT2[3:0] bits can be set for the initial value and they count the count source (in Figure 20.145, a trough is selected) selected by the extended interrupt counter 2 count source select bit (EIVTC2[1:0] bits), and repeat the count operation which return to 0 when the skipping count (in Figure 20.145 count is 2) set by the EIVTT2[3:0] bits is achieved. The initial value is set when the GTEITC register is written by the access of upper 16 bits or 32 bits while the setting for the extended interrupt skipping counter 2 is not counted (EIVTC2[1:0] bits are 00b) and when the write value to the EIVTC2[1:0] bits is other than 00b. When the initial value is set, the write value to the extended interrupt skipping counter 2 initial value bits (EITCNT2IV[3:0] bits) are set as the initial value for the EITCNT2[3:0] bits.

The extended interrupt skipping counter starts up-counting at the first count clock after the setting is modified from not counting to counting.

The interrupt skipping count counter (GTST.ITCNT[2:0] bits) for the interrupt skipping function set by the GTITC register is reset to 000b when the GTCNT counter operation is stopped: however, the EITCNT1[3:0] and EITCNT2[3:0] bits for the extended interrupt skipping function retain the value even after the GTCNT counter operation is stopped, and the counting can be resumed from the value before the counter is stopped. When values for the EITCNT1[3:0] and EITCNT2[3:0] bits are to be reset (0000b), set the EIVTC1[1:0] and EIVTC2[1:0] bits to the setting for not counting (not skipping) (00b).

When the skipping count is to be changed, change the count after stopping the skipping counter operation (set either of the EIVTC1[1:0] bits or the EIVTC2[1:0] bits to 00b).

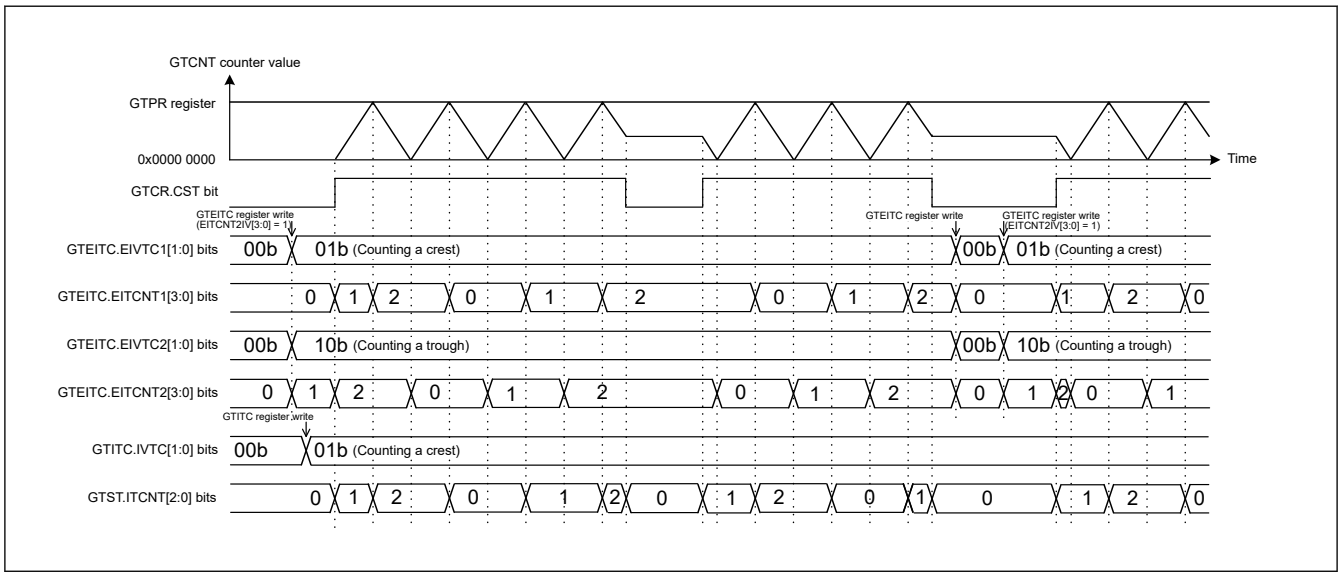


Figure 20.145 Example of counter operation for interrupt skipping

Interrupt skipping by the GTEITL1 register and A/D conversion start request skipping by the GTEITLI2 register can be performed simultaneously with skipping by the GTITC register or GTADCMSC register. A skipping period in this case is represented by OR-ed skipping periods of respective registers.

Figure 20.146 shows corresponding interrupt skipping operations by different registers are performed simultaneously.

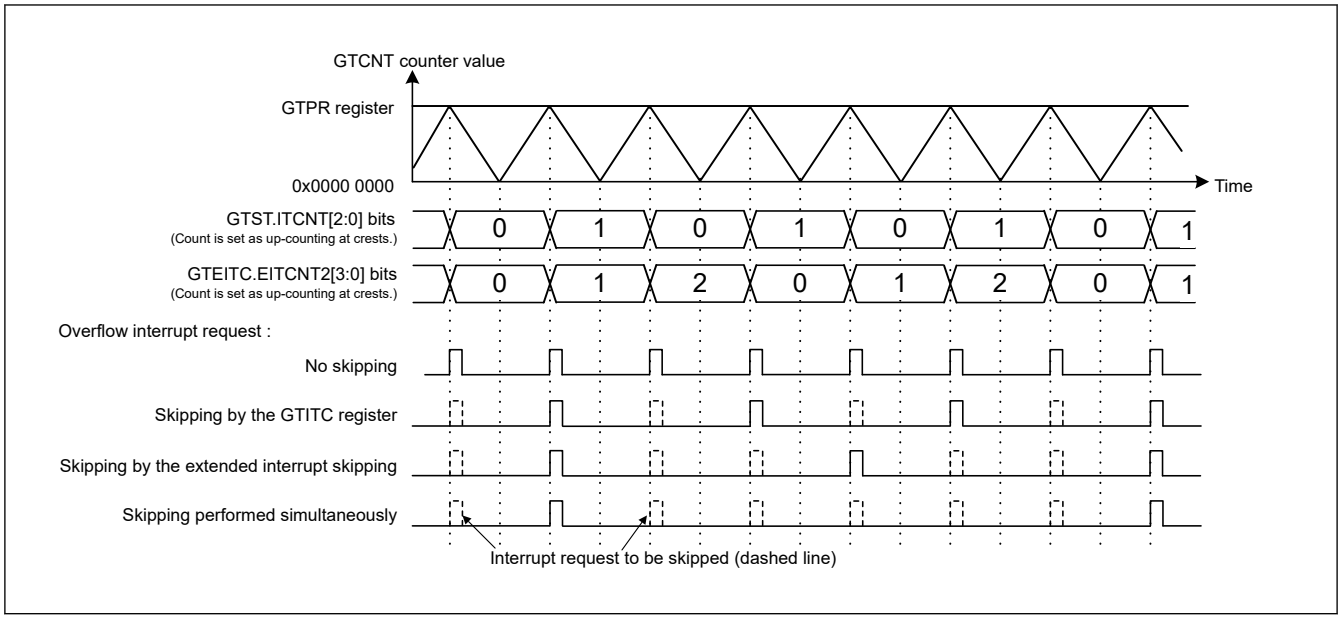


Figure 20.146 Example of interrupt skipping operation (skipping by the GTITC register: counting crests, extended interrupt skipping: EIVTC1[1:0] bits = 00b, EIVTC2[1:0] bits = 01b, EITLV[2:0] bits = 010b)

When the extended interrupt skipping selected by GTEITLI1 is performed, the relevant status flag is also skipped. The skipping function is continued even if the status flag is set to 1.

When A/D conversion start request skipping selected by GTEITLI2 is performed, the relevant status flag is also skipped. The skipping function is continued even if the status flag is set to 1.

Skipping of updating of status flag and ELC event source outputs corresponding to A/D conversion start request for which skipping can be set up in the GTEITLI2 register is only based on the settings of the GTITC register and the extended interrupt skipping register, and has no connection with the setting of A/D conversion start request enable bit in the GTINTAD register. The A/D conversion start request enable bit is used only for A/D conversion start request output(ELC event source output) after skipping.

A buffer transfer skipping by the GTEITLB register is performed in all of buffer operation which is enabled in the GTBER and GTDTCR registers and GTBER2 register, or all buffer operations performed by saw-wave one-shot pulse mode or triangle-wave PWM mode 3 or complementary PWM mode (excludes buffer transfer from GTCCRC, GTCCRE to GTCCRA).

An interrupt skipping and a buffer transfer skipping are operated individually. An interrupt output without performing a buffer transfer and performing a buffer transfer without an interrupt output can also be operated.

Figure 20.147 to Figure 20.154 show examples of extended skipping function operation.

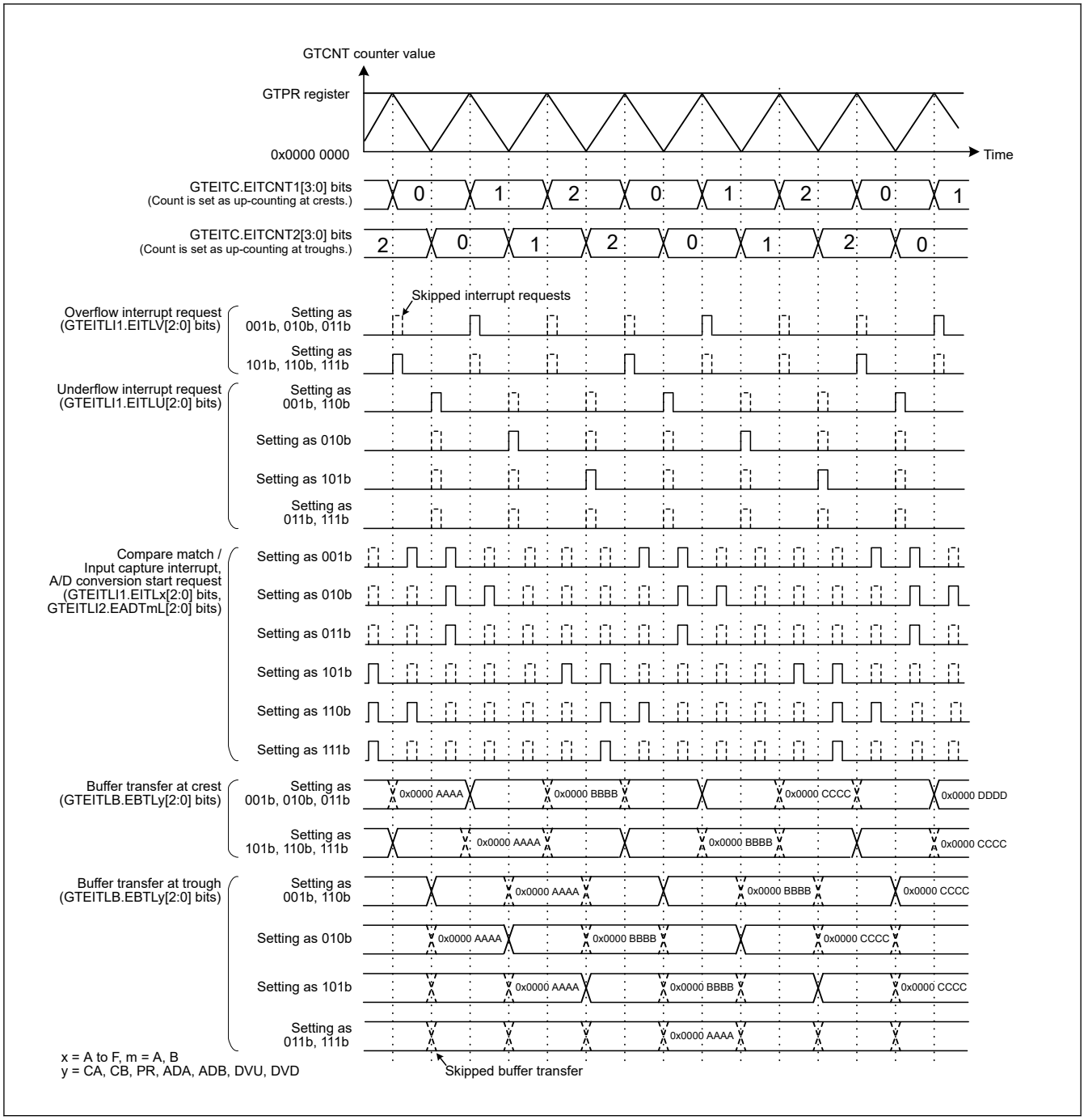


Figure 20.147 Example of extended interrupt skipping function (triangle-wave, counting crests, extended interrupt skipping 1 skipping count: 2, counting troughs, extended interrupt skipping 2 skipping count: 2, extended interrupt skipping counter 2 initial value: 0)

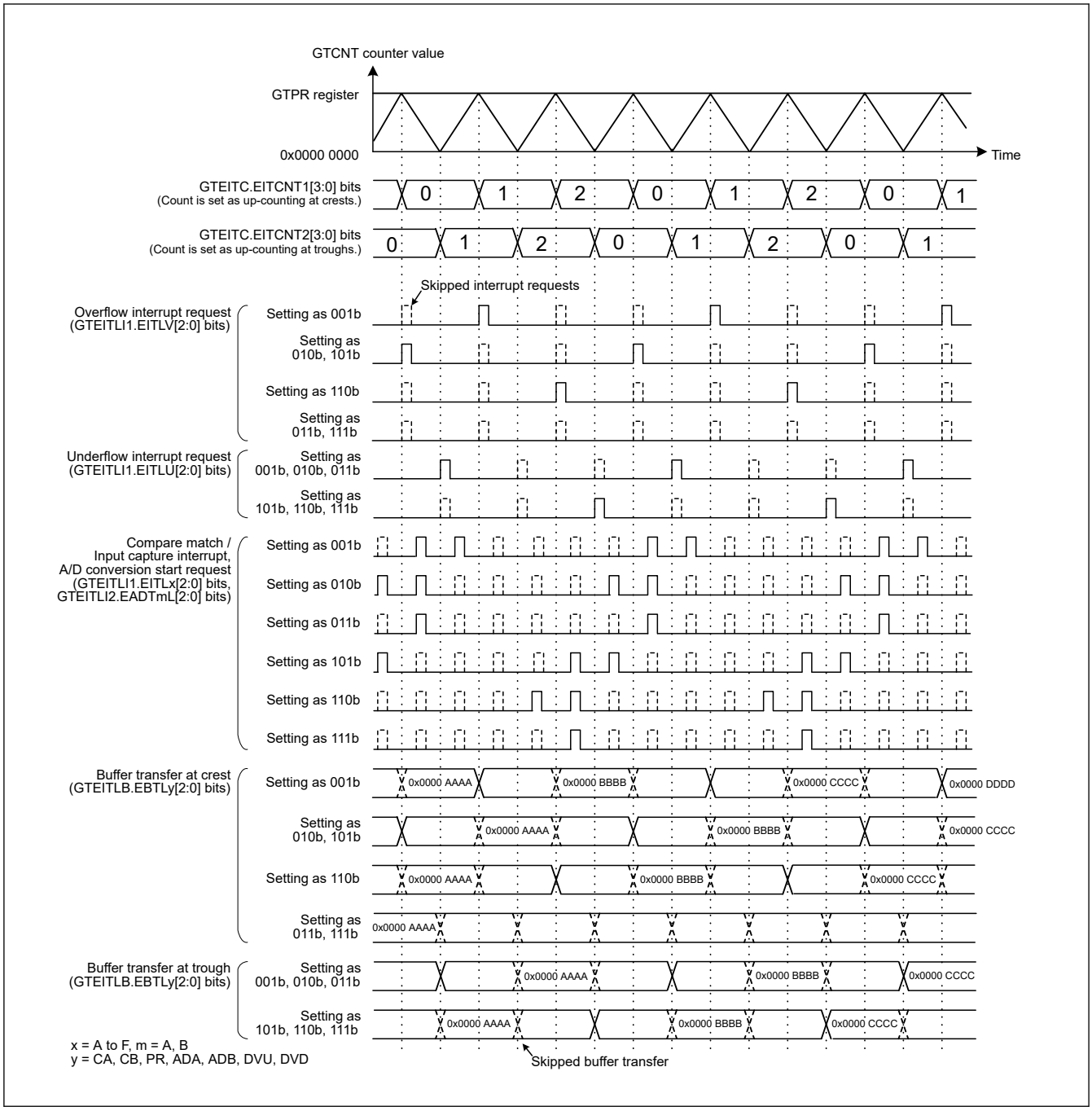


Figure 20.148 Example of extended interrupt skipping function (triangle-waves, count crests by the extended interrupt skipping 1 with skipping count: 2, count troughs by the extended interrupt skipping 2 with skipping count: 2, extended interrupt skipping counter 2 initial value: 1)

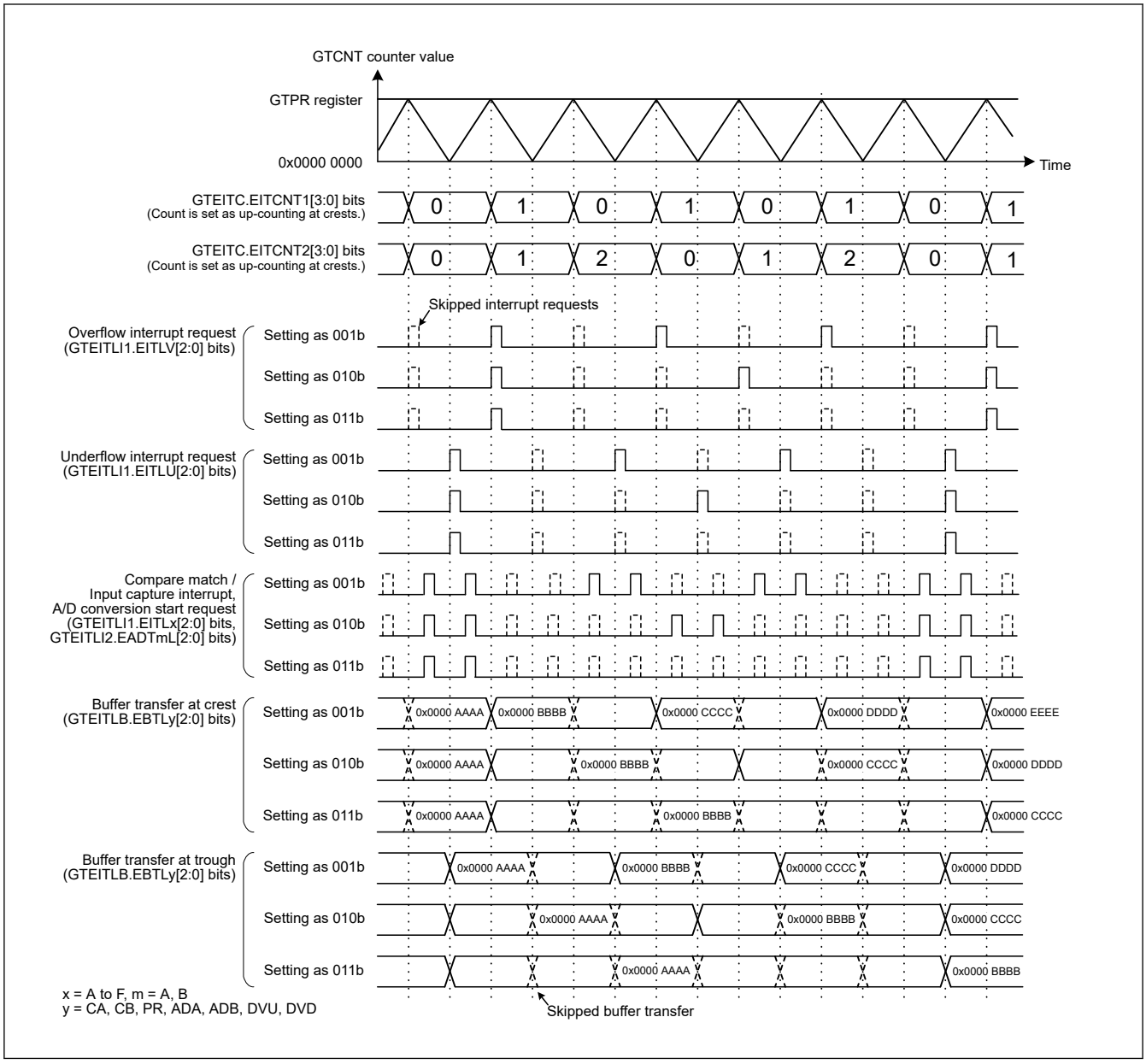


Figure 20.149 Example of extended interrupt skipping function operation (triangle-waves, counting crests, extended interrupt skipping 1 skipping count: 1, counting crests, extended interrupt skipping 2 skipping count: 2, extended interrupt skipping counter 2 initial value: 0, skipped in the period other than GTEITC.EITCNTk bits (k = 1, 2) as 0)

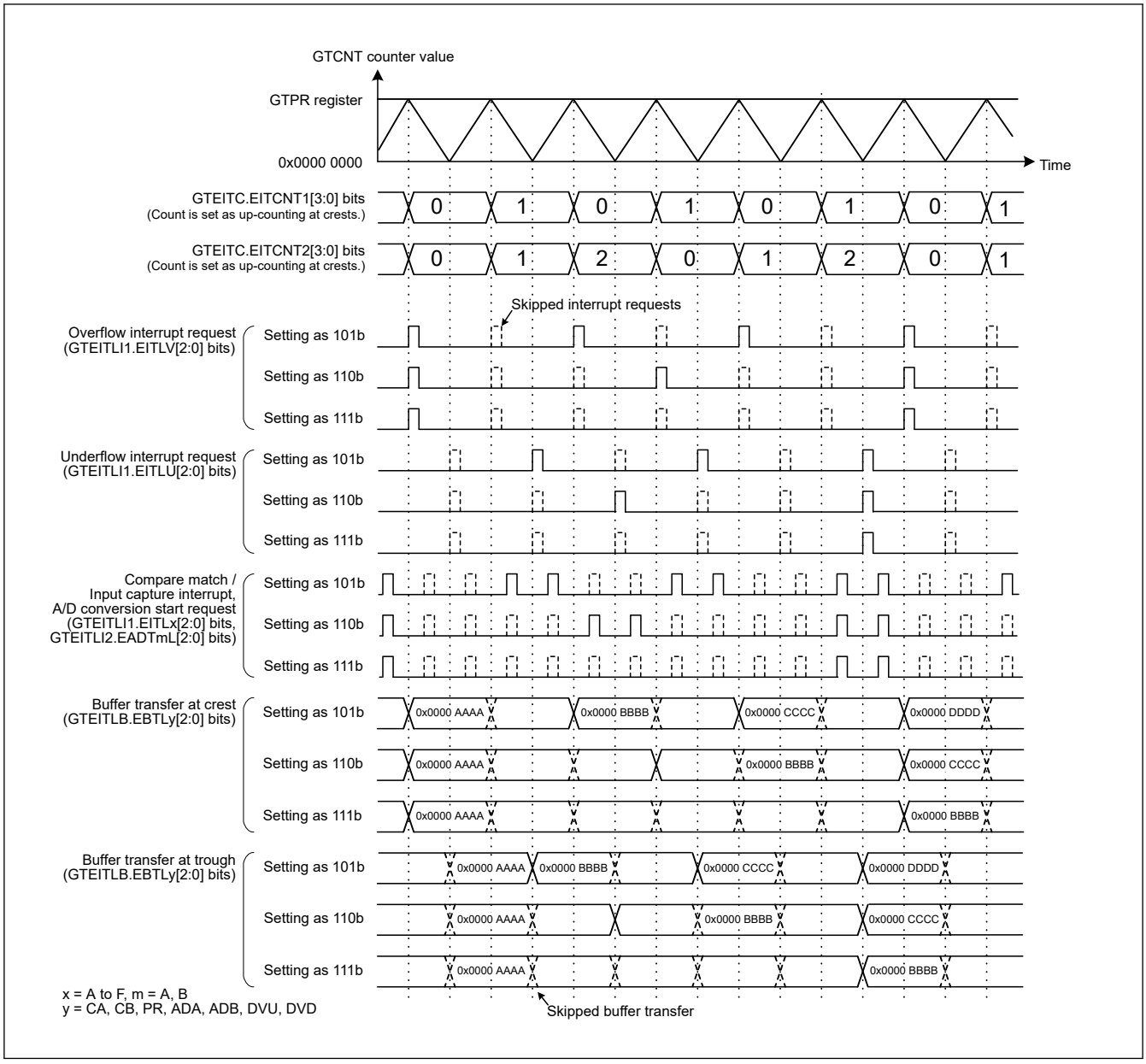


Figure 20.150 Example of extended interrupt skipping function operation (triangle-waves, counting crests, extended interrupt skipping 1 skipping count: 1, counting crests, extended interrupt skipping 2 skipping count: 2, extended interrupt skipping counter 2 initial value: 0, skipped in the period other than GTEITC.EITCNTk bits (k = 1, 2) as GTEITC.EIVTTk bits)

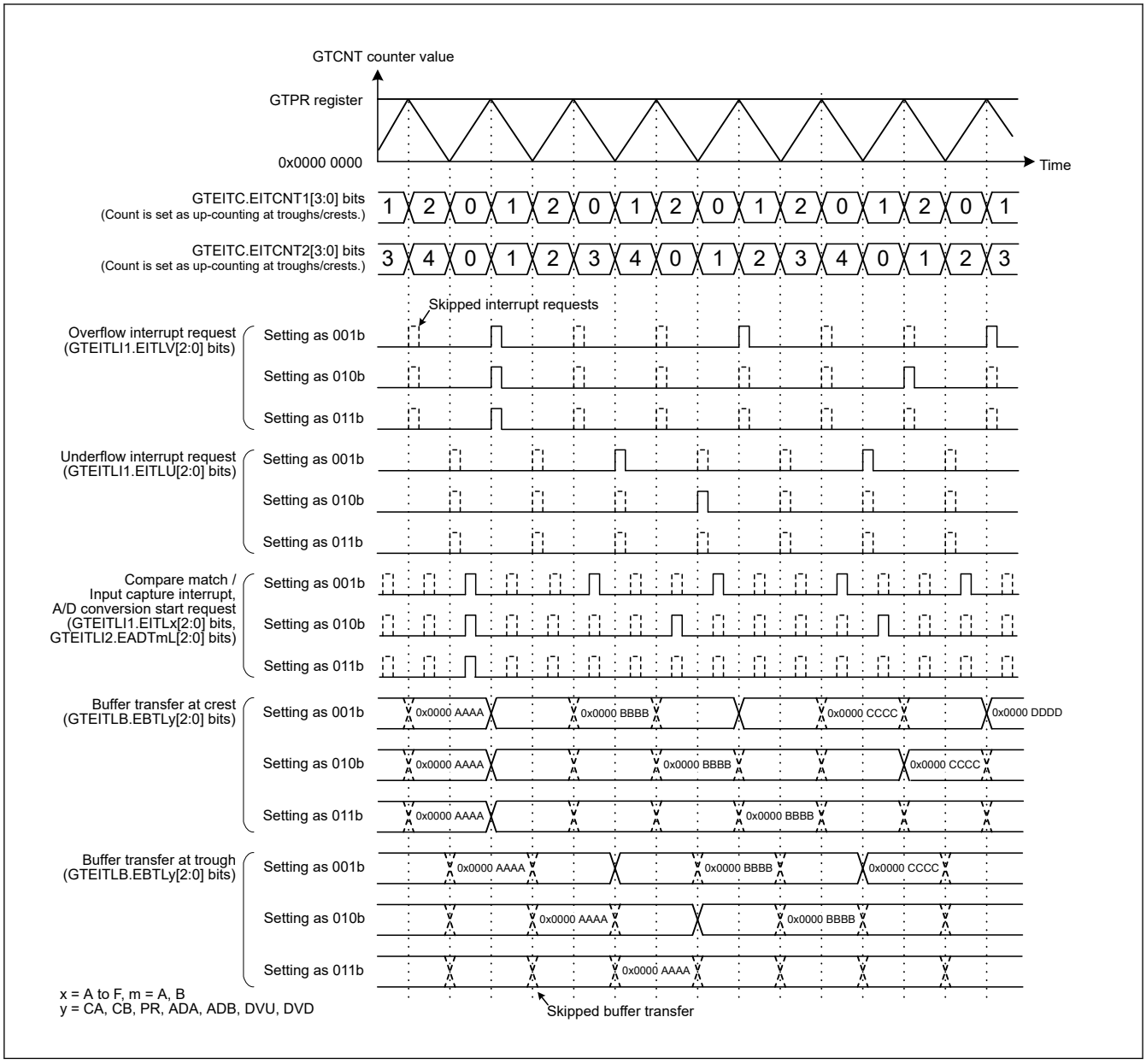


Figure 20.151 Example of extended interrupt skipping function operation (triangle-waves, counting both crests and troughs, extended interrupt skipping 1 skipping count: 2, counting both crests and troughs, extended interrupt skipping 2 skipping count: 4, extended interrupt skipping counter 2 initial value: 0, skipped in the period other than GTEITC.EITCNTk bits (k = 1, 2) as 0)

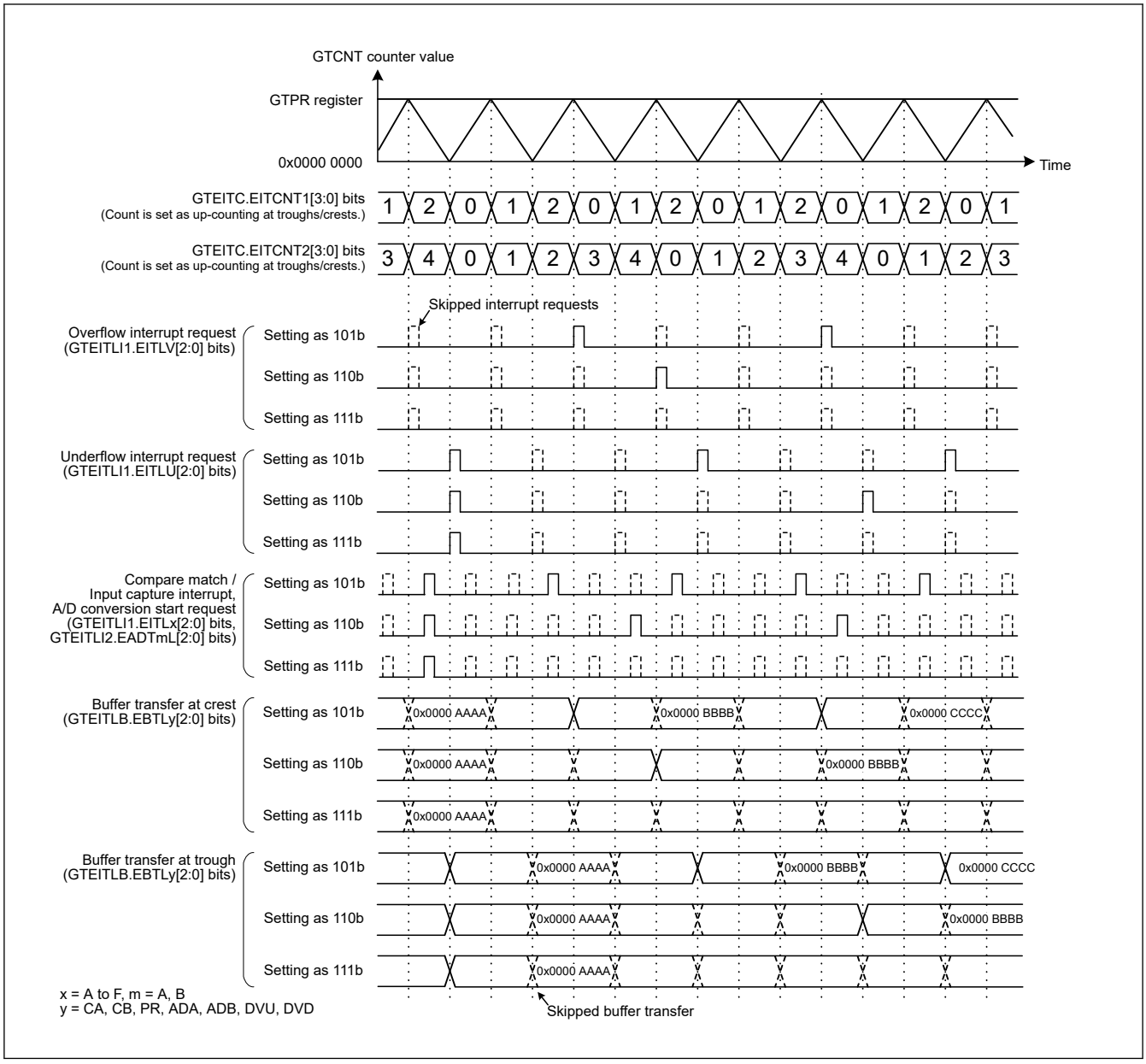


Figure 20.152 Example of extended interrupt skipping function operation (triangle-waves, counting both crests and troughs, extended interrupt skipping 1 skipping count: 2, counting both crests and troughs, extended interrupt skipping 2 skipping count: 4, extended interrupt skipping counter 2 initial value: 0, skipped in the period other than GTEITC.EITCNTk bits (k = 1, 2) as GTEITC.EIVTTk bit)

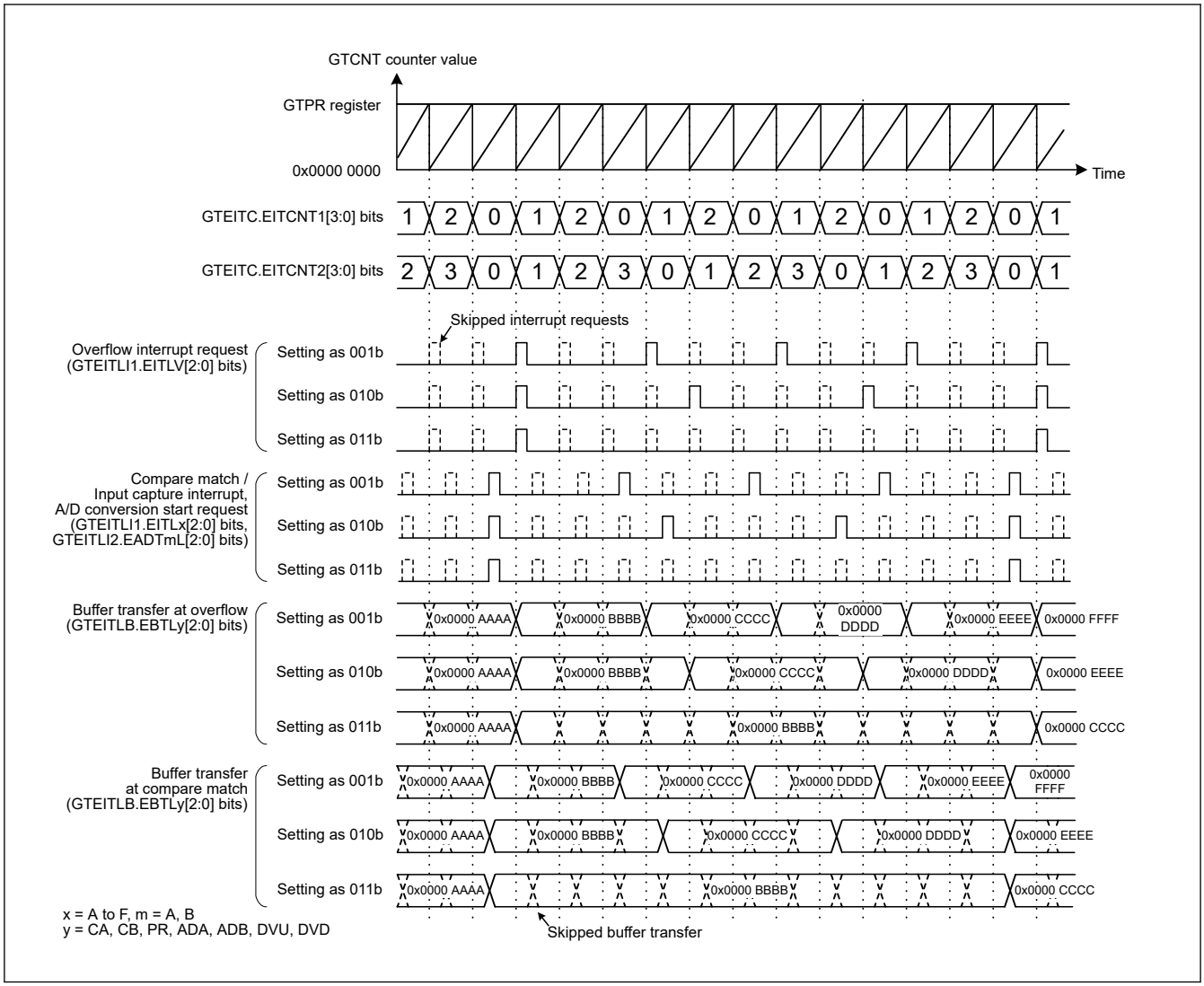


Figure 20.153 Example of extended interrupt skipping function operation (up-counting in saw-waves, counting overflow, extended interrupt skipping 1 skipping count: 2, counting overflow, extended interrupt skipping 2 skipping count: 3, extended interrupt skipping counter 2 initial value: 0, skipped in the period other than GTEITC.EITCNTk bits (k = 1, 2) as 0)

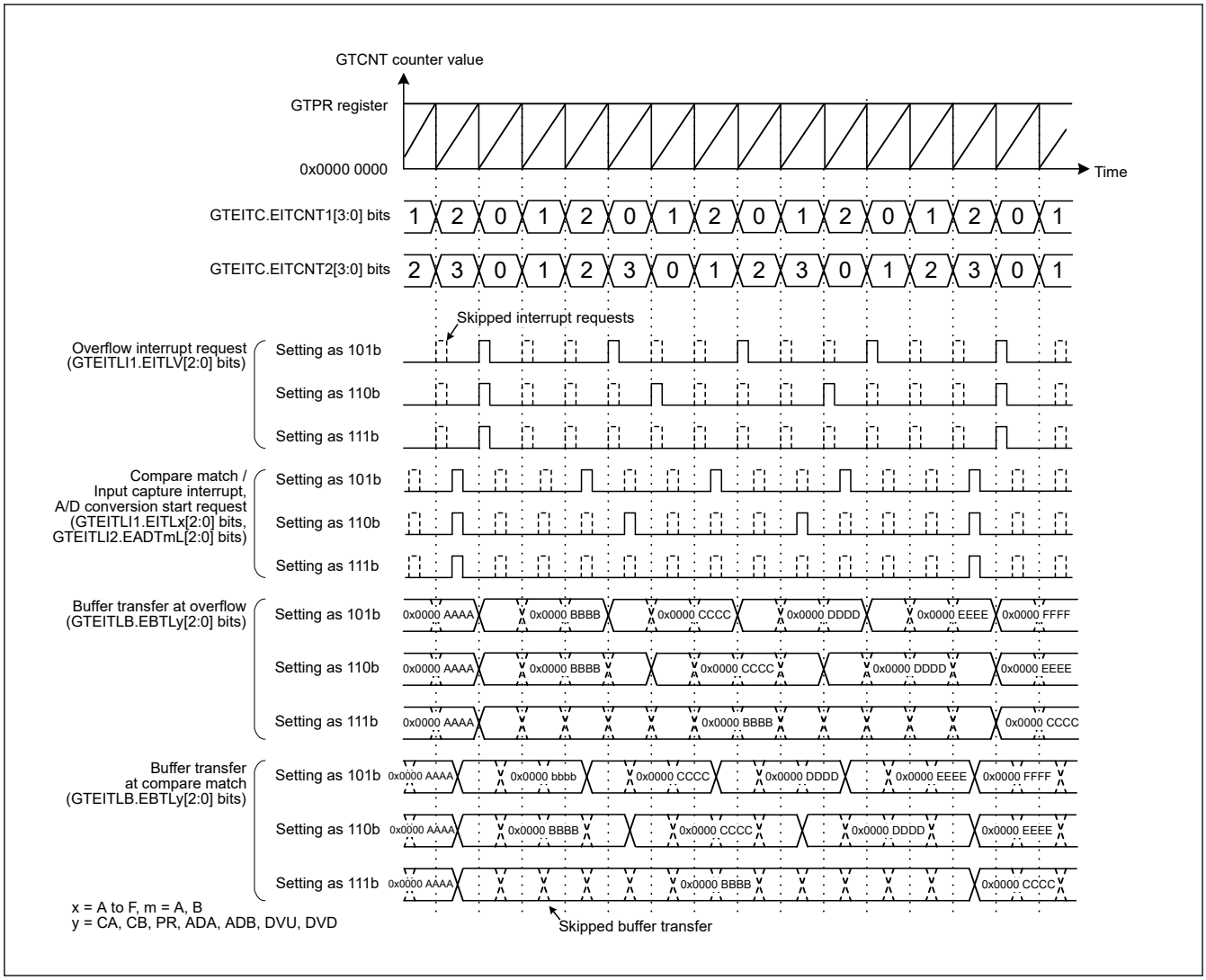


Figure 20.154 Example of extended interrupt skipping function operation (up-counting saw-waves, counting overflow, extended interrupt skipping 1 skipping count: 2, counting overflow, extended interrupt skipping 2 skipping count: 3, extended interrupt skipping counter 2 initial value: 0, skipped in the period other than GTEITC.EITCNTk bits (k = 1, 2) as GTEITC.EIVTTk bits.)

Figure 20.155 shows an example of the extended interrupt skipping operation on the input capture. When the setting is for the input capture operation setting (GTCR.ICDS bit = 0) at count stop of the GTCNT counter, the interrupt by the input capture and the extended skipping on buffer transfer is enabled even at count stop of the GTCNT counter.

When the input capture is generated at count stop of the GTCNT counter by setting the ICDS bit as 0, an interrupt and a buffer transfer are skipped if the skipping counter value is the same with the skipping period set in a corresponding interrupt skipping function select bit.

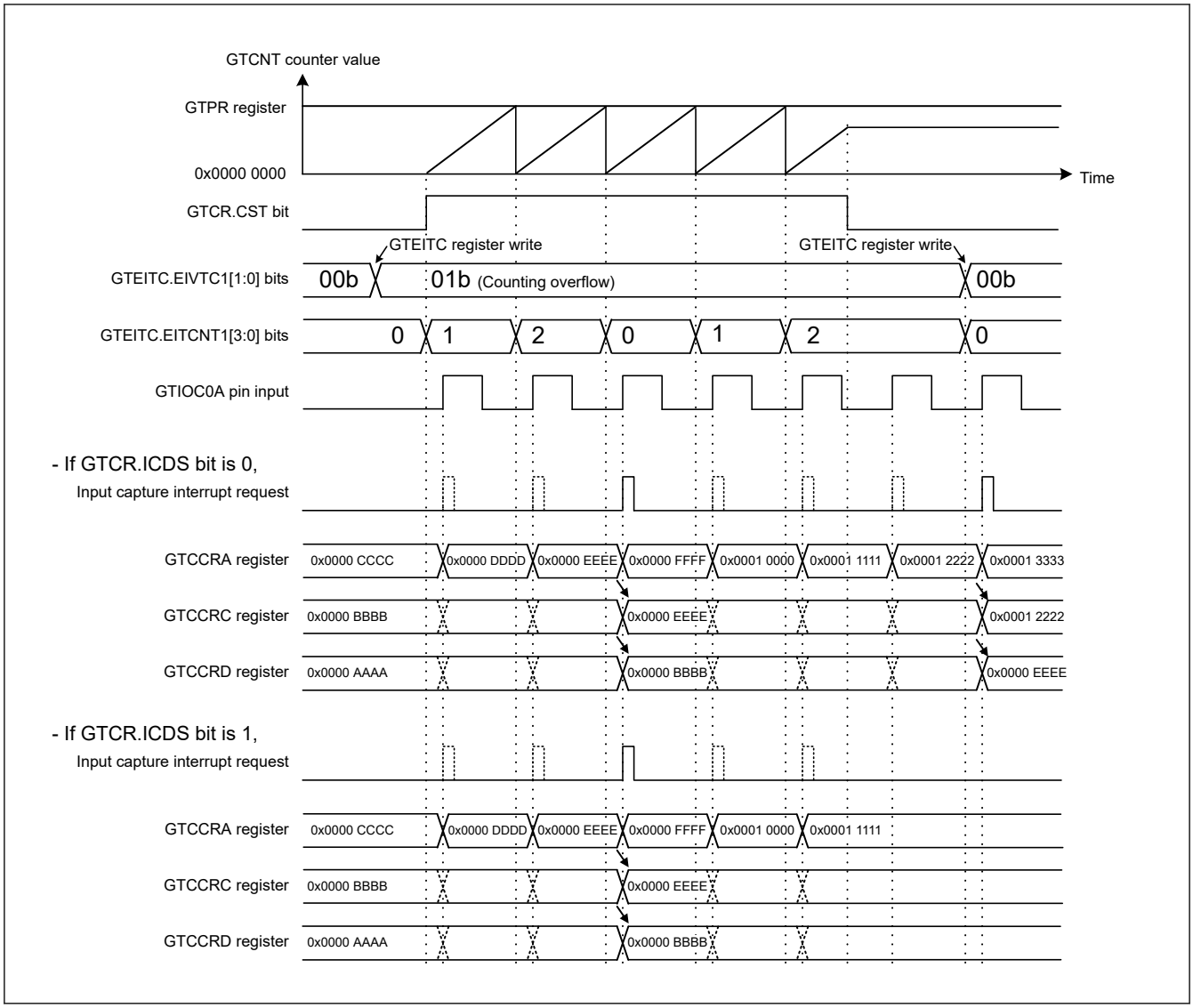


Figure 20.155 Example of input capture operation in the extended interrupt skipping function operation (up-counting saw-waves, counting overflow, extended interrupt skipping 1 skipping count: 2, skipped in the period other than the EITCNT1 as 0, input capture at the input rising)

Table 20.74 shows an example of setting the extended interrupt skipping.

The extended interrupt skipping counter 2 initial value is set by the written value to the ETICNT2IV[3:0] bits which is applied to change the setting from not counting the extended interrupt skipping counter 2 count source (GTEITC.EIVTC2[1:0] bits = 00b) to counting (EIVTC2[1:0] bits = other than 00b). Writing to the extended interrupt skipping counter 2 initial value bit (ETICNT2IV[3:0] bits) is performed only when the setting of the above mentioned extended interrupt skipping counter 2 initial value is written.

Table 20.74 Example for setting the extended interrupt skipping (1 of 2)

No.	Step Name	Description
1	Set GTCNT counter operation, Set buffer operation, Set compare match value	See section 20.3.2. Buffer Operation, section 20.3.3. PWM Output Operating Mode, and so on.
2	Set the extended interrupt skipping function	Select the skipping counter used for skipping and the skipping period with the skipping function select bit for the interrupt, for the A/D conversion start request, and for a buffer transfer, all which are to be skipped, in the GTEITLI1, GTEITLI2, and GTEITLB registers respectively.

Table 20.74 Example for setting the extended interrupt skipping (2 of 2)

No.	Step Name	Description
3	Set the extended skipping counter	With the GTEITC register, set a count source of the skipping counter used for skipping, skipping count, and the skipping counter 2 initial value as the following order. Set EIVTCk[1:0] bits (k = 1,2) as value other than 00b, and set EIVTTk[3:0] bits as value other than 0000b. When the skipping counter 2 is used, change EIVTC2[1:0] bits from 00b to value other than 00b as well as set EITCNT2IV[3:0] bits to the skipping counter 2 initial value.
4	Start count operation	Set GTCR.CST to 1 to start count operation.
5	Set buffer value for each cycle	See section 20.3.2. Buffer Operation, section 20.3.3. PWM Output Operating Mode, and so on.

20.4.3.3 A/D Conversion Start Request Compare Match Skipping Function

The A/D conversion start request and GTADTR register buffer transfer can be skipped by counting the compare matches of GTADTRA register and GTADTRB register based on settings of the GTADCMSC register and GTADCMSS register.

The skipping period is set, related to the operation of the two independent A/D conversion start request compare match skipping counters (ADCMSCNT1[3:0] and ADCMSCNT2[3:0] bits in the GTADCMSC register), as a period that either of the counter value for such skipping counters is other than 0 or other than the skipping count. The skipping period can also be set as a period that the both skipping counters are set as other than 0 or other than the skipping count for the counter value.

Figure 20.156 shows the counter operation for A/D conversion start request compare match skipping.

The counter operation for A/D conversion start request compare match skipping is set by the GTADCMSC register.

The ADCMSCNT1[3:0] bits count the count source (in Figure 20.156, a crest is selected) selected by the A/D conversion start request compare match skipping counter 1 count source select bit (GTADCMSC.ADCMSC1[1:0] bits), and repeat the count operation which returns to 0 when the skipping count (in Figure 20.156, count is 2) set by the A/D conversion start request compare match skipping 1 skipping count setting bit (ADCMST1[3:0] bits) is achieved.

The ADCMSCNT2[3:0] bits count the count source (in Figure 20.156, a trough is selected) selected by the A/D conversion start request compare match skipping counter 2 count source select bit (GTADCMSC.ADCMSC2[1:0] bits), and repeat the count operation which returns to 0 when the skipping count (in Figure 20.156, count is 2) set by the A/D conversion start request compare match skipping 2 skipping count setting bit (ADCMST2[3:0] bits) is achieved.

The ADCMSCNTk[3:0] bits (k = 1, 2) can be set for the initial value. The initial value is set when the GTADCMSC register is written by the access of 16 bits or 32 bits while the setting for the A/D conversion start request compare match skipping counter k is not counted (ADCMSCk [1:0] bits are 00b) and the write value to the ADCMSCk[1:0] bits is other than 00b. When the initial value is set, the write value to the A/D conversion start request compare match skipping counter k initial value bits (ADCMSCNTkIV[3:0] bits) are set as the initial value for the ADCMSCNT2[3:0] bits.

The A/D conversion start request compare match skipping counter starts up-counting at the first count clock after the setting is modified from not counting to counting.

The ADCMSCNT1[3:0] and ADCMSCNT2[3:0] bits for the A/D conversion start request compare match skipping function retain the value even after the GTCNT counter operation is stopped, and the counting can be resumed from the value before the counter is stopped. When values for the ADCMSCNT1[3:0] and ADCMSCNT2[3:0] bits are to be reset (0000b), set the ADCMSC1[1:0] and ADCMSC2[1:0] bits to the setting for not counting (not skipping) (00b).

When the skipping count is to be changed, change the count after stopping the skipping counter operation (set either of the ADCMSC1[1:0] bits or the ADCMSC2[1:0] bits to 00b).

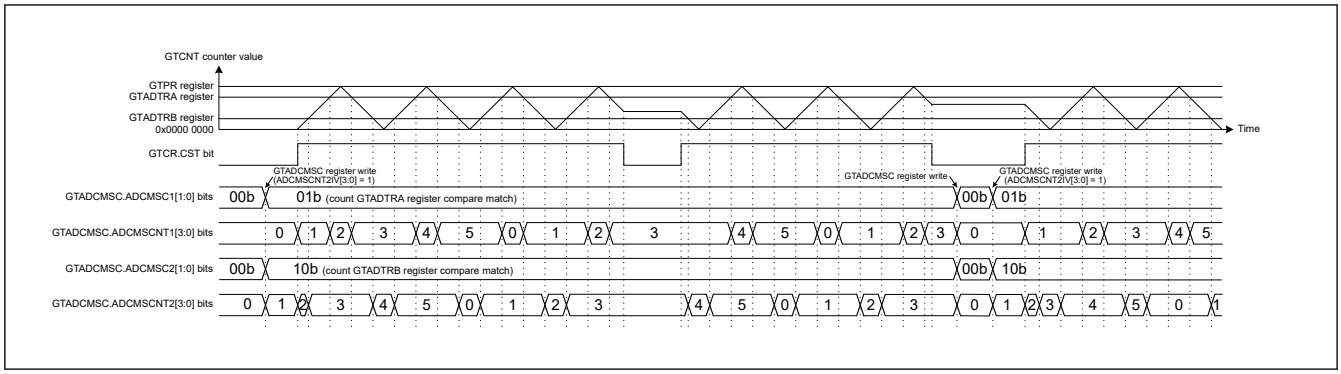


Figure 20.156 Example of A/D conversion start request compare match skipping function operation

A/D conversion start request skipping by the ADCMSC register can be performed simultaneously with skipping by the GTITC register or GTEITC register. A skipping period in this case is represented by OR-ed skipping periods of respective registers.

Figure 20.157 shows corresponding interrupt skipping operations and A/D conversion start request skipping operation are performed simultaneously.

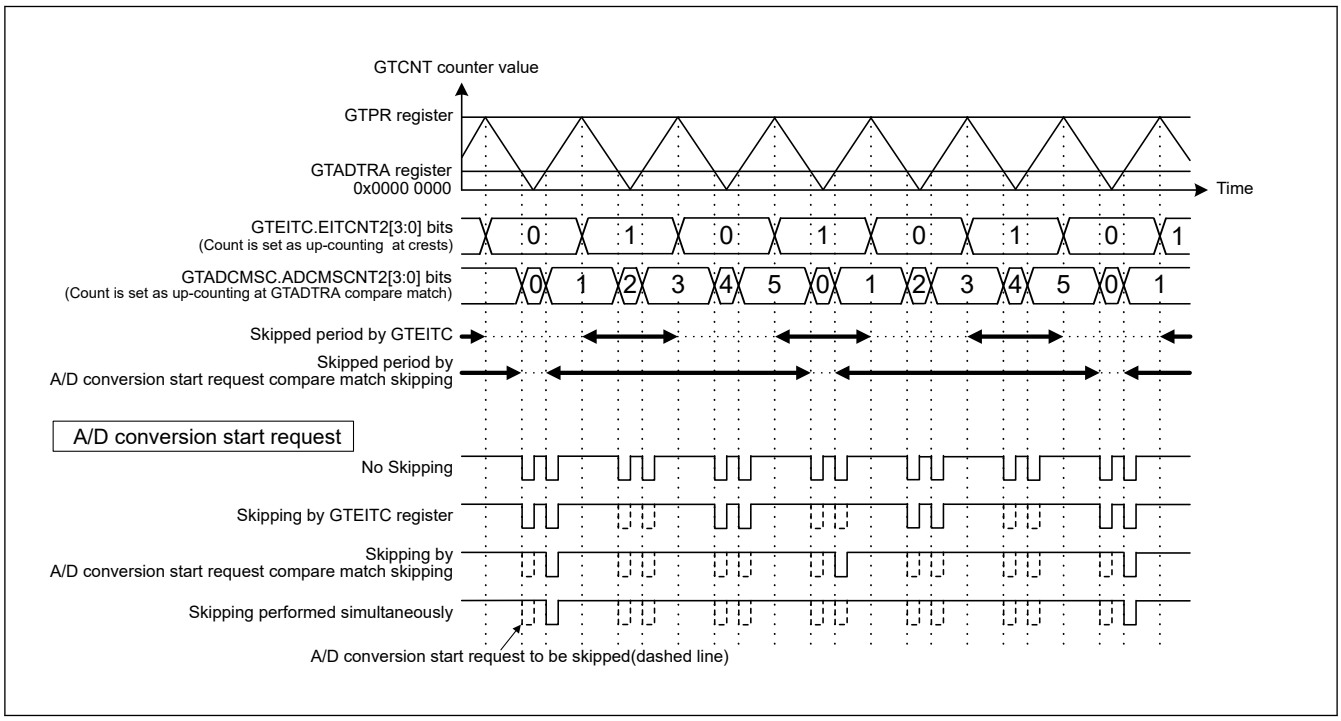


Figure 20.157 Example of A/D conversion start request compare match skipping function operation (extended interrupt skipping: EIVTC2[1:0] = 01b, EADTAL[2:0] = 010b, A/D conversion start request compare match skipping: ADCMSC2[1:0] = 01b, ADCMSAL[2:0] = 010b)

When A/D conversion start request skipping which can be set by the GTADCMSS register is performed, updating of status flag and the ELC event output depend on the A/D conversion start request enable bit in the GTINTAD register. All the operations by A/D conversion start request which are set as disable by the GTINTAD register is not performed.

A buffer transfer skipping by the GTADCMSS register is performed in all of buffer operation which is enabled in the GTBER and GTBER2 register, or all buffer operations performed by saw-wave one-shot pulse mode or triangle-wave PWM mode 3 or complementary PWM mode (excludes buffer transfer from GTCCRC, GTCCRE to GTCCRA).

An A/D conversion start request skipping and a buffer transfer skipping are operated individually.

Figure 20.158, Figure 20.159 show examples of extended skipping function operation.

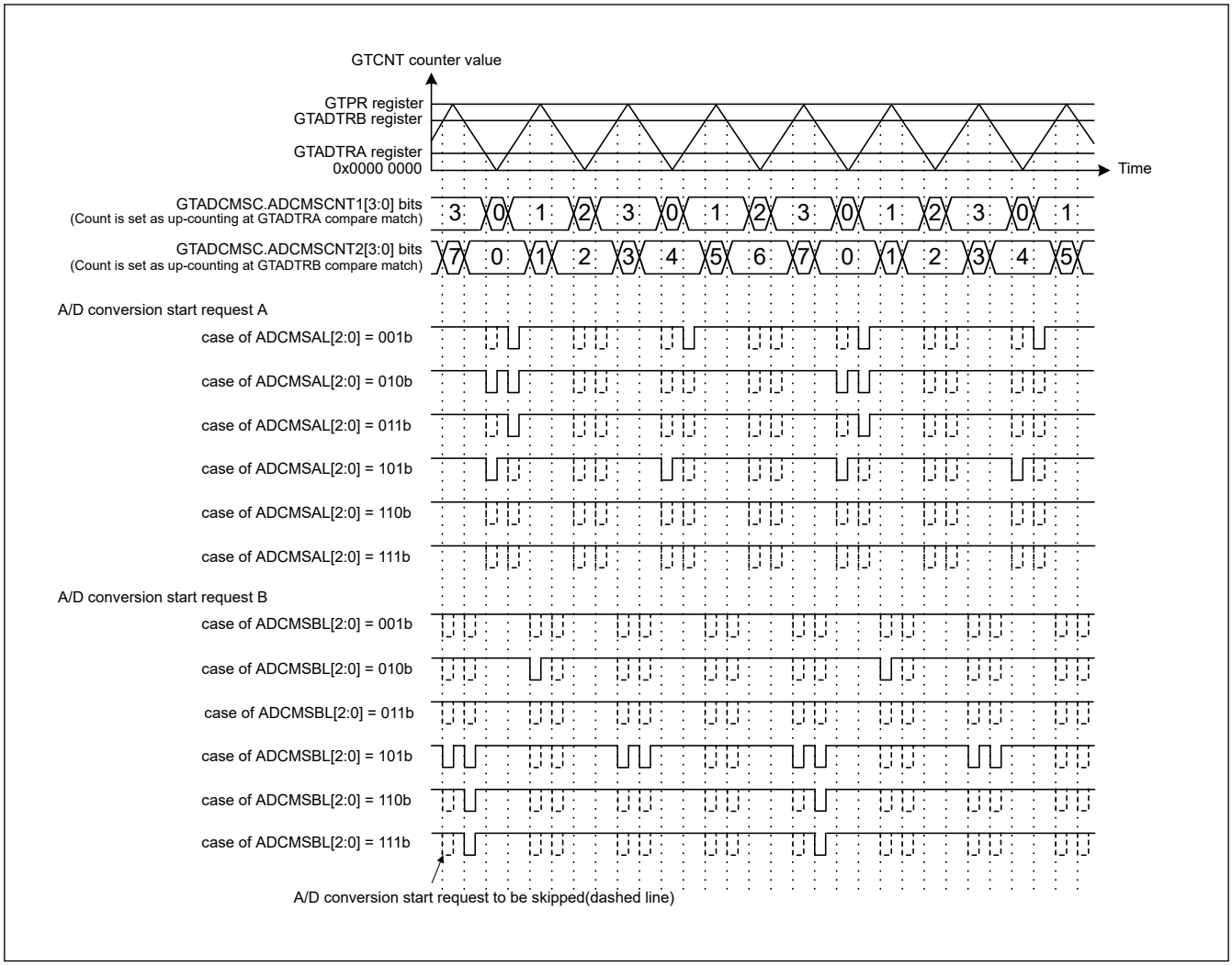


Figure 20.158 Example of A/D conversion start request compare match skipping function operation (triangle-wave, A/D conversion start request compare match skipping 1 skipping count: 3 counting GTADTRA compare match, A/D conversion start request compare match skipping 2 skipping count: 7 counting GTADTRB compare match)

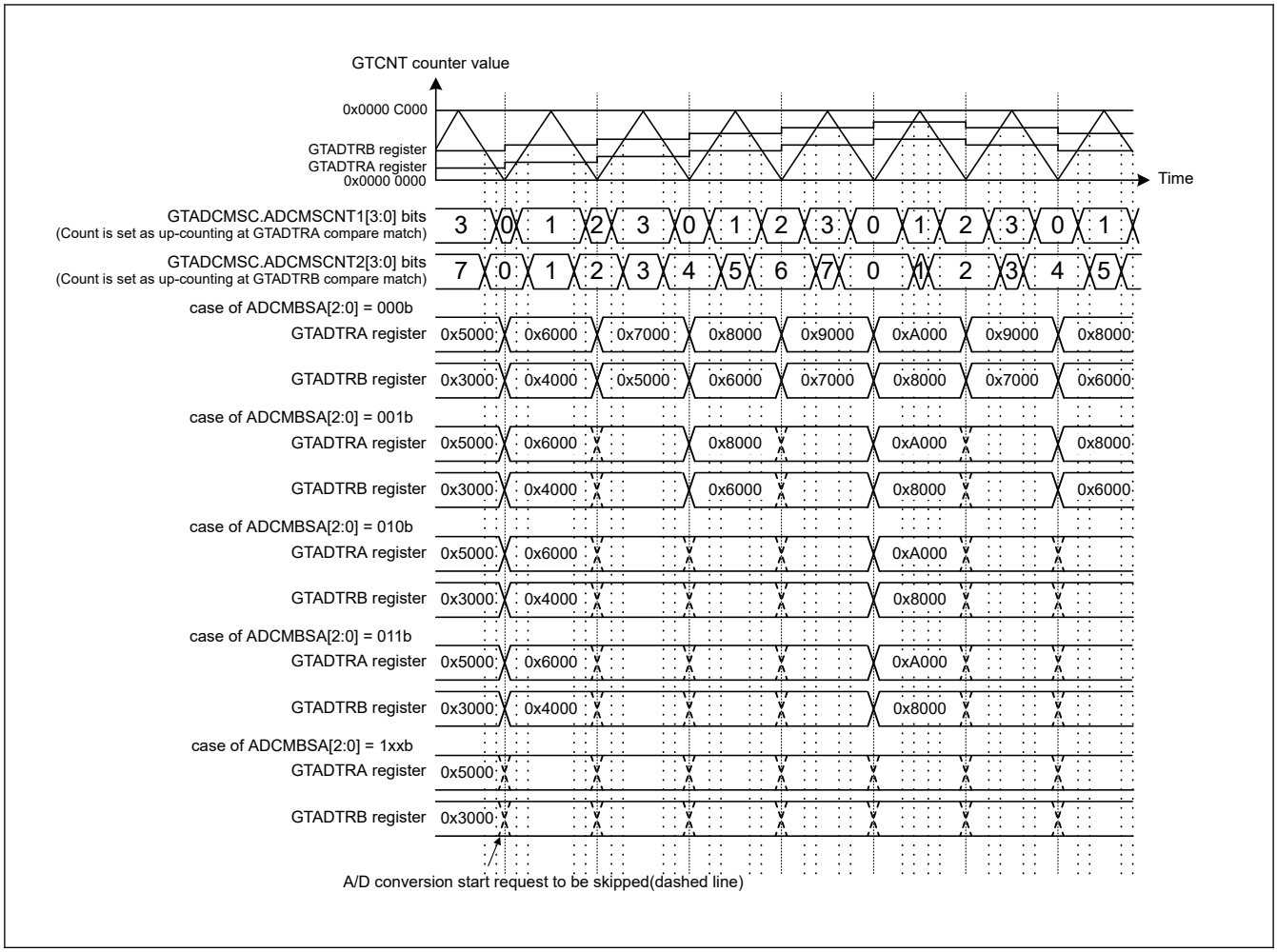


Figure 20.159 Example of A/D conversion start request compare match skipping function operation (triangle-wave, A/D conversion start request compare match skipping 1 skipping count: 3 counting GTADTRA compare match, A/D conversion start request compare match skipping 2 skipping count: 7 counting GTADTRB compare match, buffer transfer of GTADTRA and GTADTRB at trough)

Table 20.75 shows an example of setting the A/D conversion start request compare match skipping.

The A/D conversion start request compare match skipping counter 2 initial value is set by the written value to the ADCMSCNT2IV[3:0] bits which is applied to change the setting from not counting the extended interrupt skipping counter 2 count source (GTADCMSC.ADCMSC2[1:0] bits = 00b) to counting (ADCMSC2[1:0] bits = other than 00b). Writing to the A/D conversion start request compare match skipping counter 2 initial value bit (ADCMSCNT2IV[3:0] bits) is performed only when the setting of the above mentioned A/D conversion start request compare match skipping counter 2 initial value is written.

Table 20.75 Example for setting the A/D conversion start request compare match skipping (1 of 2)

No.	Step Name	Description
1	Set GTCNT counter operation, Set buffer operation, Set compare match value	See section 20.3.2. Buffer Operation, section 20.3.3. PWM Output Operating Mode, section 20.5. A/D Conversion Start Request, and so on.
2	Set the A/D conversion start request compare match skipping function	Select the skipping counter and the skipping period with the skipping function select bit for the A/D conversion start request or the bit for the buffer transfer to be skipped in the GTADCMSS register.

Table 20.75 Example for setting the A/D conversion start request compare match skipping (2 of 2)

No.	Step Name	Description
3	Set the A/D conversion start request compare match skipping counter k (k = 1, 2)	With the GTADCMSC register, set a count source of the skipping counter, skipping count, and the skipping counter k initial value as the following order. Set the GTADCMSC.ADCMSCk[1:0] bits as value other than 00b, and set the ADCMSTk[3:0] bits as value other than 0000b. Change the ADCMSCk[1:0] bits from 00b to value other than 00b as well as set the ADCMSCNTkIV[3:0] bits to the skipping counter k initial value.
4	Start count operation	Set GTCR.CST to 1 to start count operation.
5	Set buffer value for each cycle	See section 20.3.2. Buffer Operation, section 20.3.3. PWM Output Operating Mode, section 20.5. A/D Conversion Start Request, and so on.

20.5 A/D Conversion Start Request

The A/D conversion start request can be issued at a compare match between the GTCNT counter and the GTADTRA or GTADTRB register. Up-counting only, down-counting only, or both up-counting and down-counting can be specified by setting the GTINTAD register.

In complementary PWM mode, the A/D conversion start request can be issued at a compare match with the GTCNT counter of master channel.

During event count operation, the A/D conversion start request cannot be generated.

The A/D conversion start request is output as event signals to ELC.

The GTADTRA and GTADTRB registers each has two buffer registers. Buffer operation with the GTADTRA register used together with the GTADTBRA and GTADTDBRA registers, and buffer operation with the GTADTRB register used together with the GTADTBRB and GTADTDBRB registers can be performed.

The timing of the generation of requests to start A/D conversion can be monitored by an external pin. When the A/D conversion start request signal to be monitored is selected in the GTADSMR.ADSMSk bit (k = 0, 1) and when the output is enabled in the ADSMENk bit, a signal is output synchronized with a cycle frame of the timer used to generate the A/D conversion start request signal, of which the output is driven high at the generation of the A/D conversion start request signal by the GTADSMk pin, or at the end of the cycle of which the output is driven low. When a signal to request the start of A/D conversion is generated at the end of the cycle, the generation of this signal has priority in terms of monitoring output and the output remains at the high level till the end of the next cycle. The registers (GTADTRA and GTADTRB) that are sources of generating the A/D conversion start request signals and their counting directions can be checked by the A/D conversion start request flags (ADTRAUF, ADTRADF, ADTRBUF, and ADTRBDF) in the GTST register. When the output of the same A/D conversion start request signal monitoring output is enabled for multiple channels, ORed signals will be output from the GPT16.

Figure 20.160 shows an example of A/D conversion start request operation, Table 20.76 shows example for setting A/D conversion start request operation.

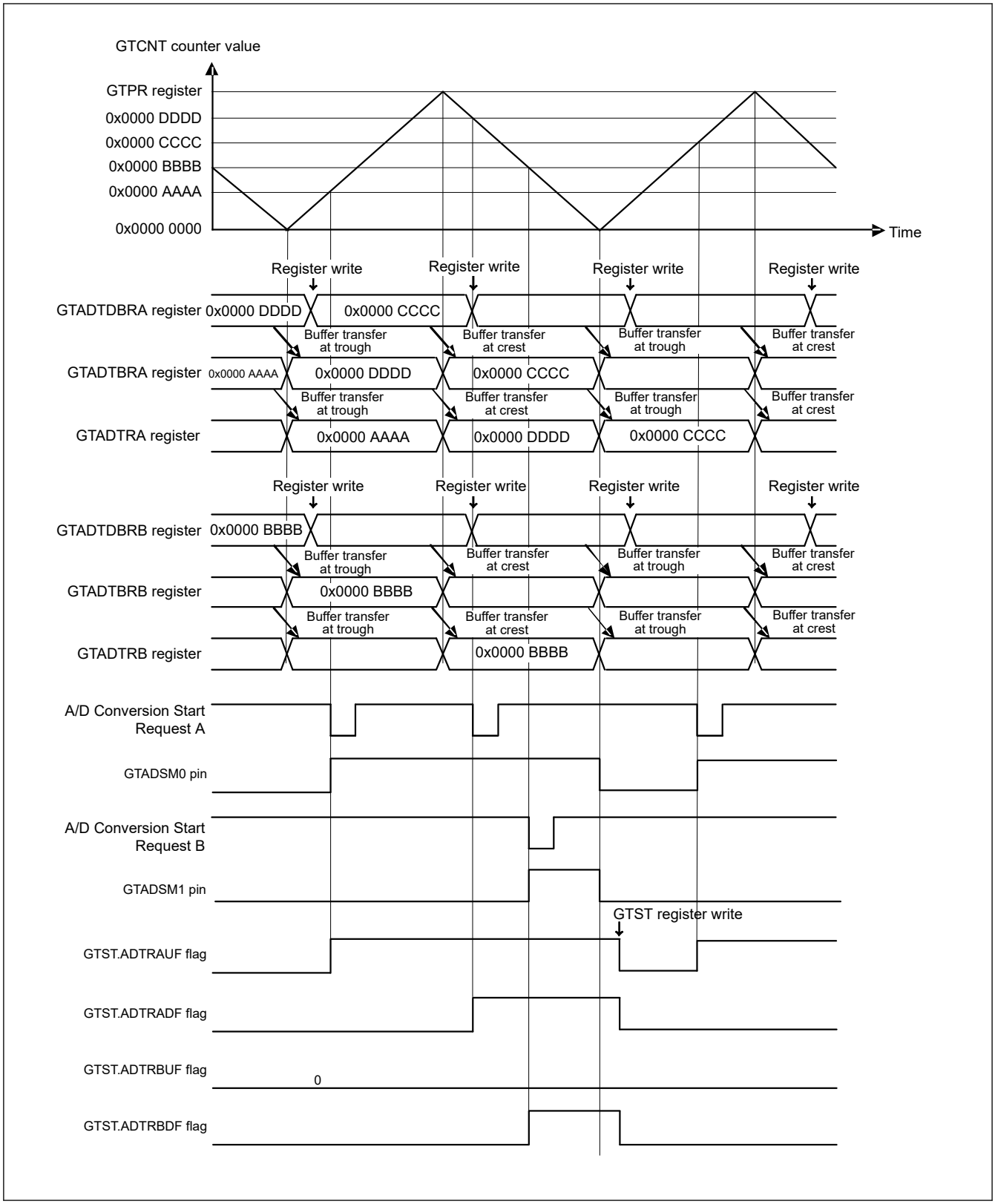


Figure 20.160 Example of A/D conversion start request timing operation (triangle-waves, double buffer operation, buffer transfer at both troughs and crests, A/D conversion start request by GTADTRA register at both up-counting and down-counting, A/D conversion start request by GTADTRB register at down-counting, monitoring of the GTADTRA register up-counting by the GTADSM0 pin, monitoring of the GTADTRB register down-counting by the GTADSM1 pin)

Table 20.76 Example for setting A/D conversion start request timing operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits or GTCR.MD[3:0] bits. In Figure 20.160, 100b, 101b, or 110b (0100b, 0101b, or 0110b) (triangle-wave PWM mode) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set buffer operation	Set buffer operation with the ADTTA[1:0], ADTTB[1:0], ADTDA, and ADTDB bits in the GTBER register. In Figure 20.160, ADTTA[1:0] = 11b, ADTTB[1:0] = 11b, ADTDA = 1, and ADTDB = 1.
6	Set compare match value	Set the A/D conversion start request point in the GTADTRA and GTADTRB registers.
7	Set buffer value	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.
8	Set A/D conversion start request for monitoring	Select the A/D conversion start request signal to be monitored with ADSMS0[1:0] and ADSMS1[1:0] bits in GTADSMR from GTADSM0 and GTADSM1 pins and enable output of the A/D conversion start request signal being monitored to ADSMEN0 and ADSMEN1 bits in GTADSMR. In Figure 20.160, ADSMS0[1:0] = 00b, ADSMS1[1:0] = 11b, ADSMEN0 = 1, and ADSMEN1 = 1.
9	Enable A/D conversion start request	Set to enable A/D conversion start request with the ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN bits in the GTINTAD register. In Figure 20.160, ADTRAUEN = 1, ADTRADEN = 1, ADTRBUEN = 0, and ADTRBDEN = 1.
10	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
11	Set buffer value for each cycle	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.

Figure 20.161 shows an example for A/D conversion start request timing operation.

This shows an example of the output of A/D conversion start request A by the ELC as start source 0 (ELC_AD00) for the A/D converter. The A/D conversion start request A signal is output by the ELC in response to a match in comparison with the GTADTRA register.

The A/D conversion start request A signal is output by the ELC in response to a match in comparison with the GTADTRA register. The A/D conversion start request A signal is synchronized with GTCLK. The ELC receives the signal in synchronization with PCLKB, and then outputs the A/D conversion start source 0 signal after 1 cycle of the PCLKB has elapsed.

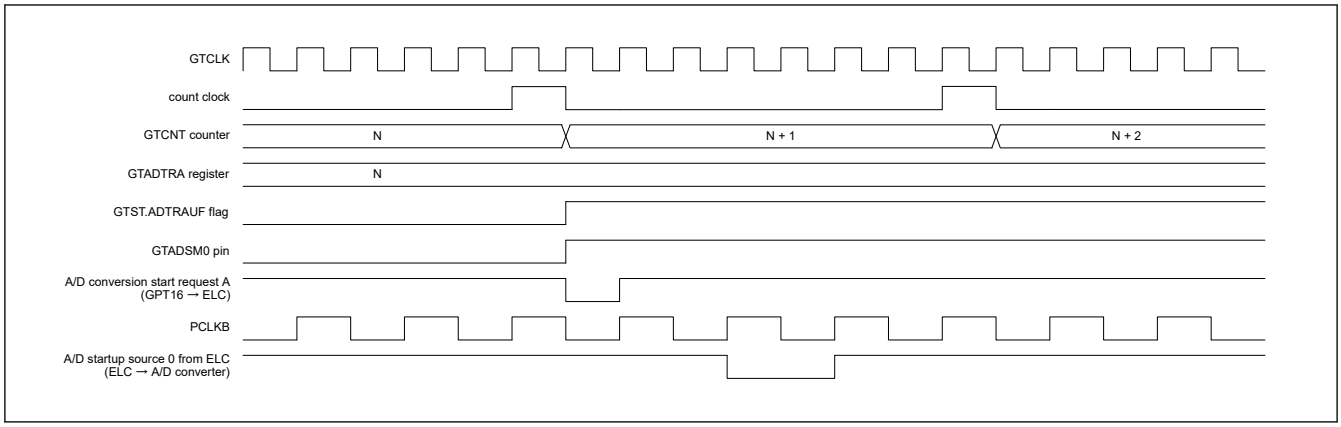


Figure 20.161 Example of A/D conversion start request timing operation

For the restriction of A/D Conversion Start Request, see section 16, Event Link Controller (ELC).

20.6 Operations Linked by ELC

20.6.1 Event Signal Output to ELC

The GPT can perform operation linked with another module set in advance when its interrupt request signal is used as an event signal by the Event Link Controller (ELC).

The event signal can be output regardless of the setting of the corresponding interrupt request enable bits except the A/D conversion start request. The A/D conversion start request during up-counting/down-counting can be enabled/disabled individually with the A/D conversion start request enable bit to output both interrupts and events output to ELC.

The GPT has the following ELC event signals:

- Generation of compare match and input capture A interrupt (GPTn_CCMPA)
- Generation of compare match and input capture B interrupt (GPTn_CCMPB)
- Generation of compare match C interrupt (GPTn_CMPC)
- Generation of compare match D interrupt (GPTn_CMPD)
- Generation of compare match E interrupt (GPTn_CMPE)
- Generation of compare match F interrupt (GPTn_CMPF)
- Generation of overflow interrupt (GPTn_OVF)
- Generation of underflow interrupt (GPTn_UDF)
- Generation of A/D conversion start request A (GPTn_ADTRGA)
- Generation of A/D conversion start request B (GPTn_ADTRGB)
- Finish of period count function (GPTn_PC)

Note: n = 0 to 3

20.6.2 Event Signal Inputs from ELC

The GPT can perform the following operations in response to a maximum of 8 events from the ELC:

- Start counting, stop counting, clear counting
- Up-counting, down-counting
- Input capture.

See section 16, Event Link Controller (ELC) for the connection between the ELC and the event signal input.

20.7 Noise Filter Function

Each pin for use in input capture and Hall sensor input to the GPT is equipped with a noise filter. The noise filter samples input signals at the sampling clock and removes the pulses whose length is less than 3 sampling cycles.

The noise filter functionality includes enabling and disabling the noise filter for each pin and setting of the sampling clock for each channel.

Figure 20.162 shows the timing of noise filtering.

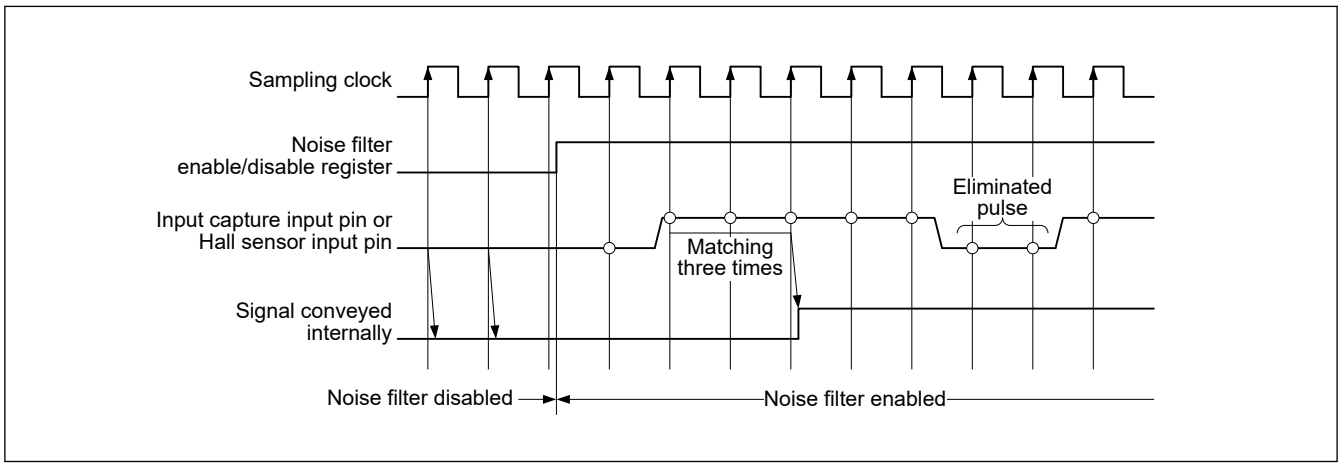


Figure 20.162 Timing of noise filtering

If noise filtering is enabled, the input capture operation or output phase switching operation is performed on the edges of the noise filtered signal after a delay of (sampling interval × 2 + GTCLK) at the shortest. This is due to the noise filtering for the input capture input or hall sensor input.

20.8 Protection Function

20.8.1 Write-Protection for Registers

To prevent registers from being accidentally modified, registers can be write-protected in channel units by setting GTWP.WP. Write-protection can be set for the following registers:

GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTITC, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTPDBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTDVD, GTDBU, GTDBD, GTSOS, GTSOTR, GTADSMR, GTEITC, GTEITLI1, GTEITLI2, GTEITLB, GTPC, GTADCMSC, GTADCMSS, GTBER2, GTOLBR, GTICCR.

Every bit in registers GTSTR, GTSTP and GTCLR which can update the corresponding registers in other channels and can be updated by any of the corresponding registers in other channels conversely, can be protected by setting the GTWP.STRWP, STPWP, and CLRWP bits, respectively, per channel.

Likewise, writing to the GTSECSR and GTSECR registers, which can control all channels by writing to the GTSECSR and GTSECR registers of a given channel, can by enabled or disabled by the setting of the GTWP.CMNWP bit.

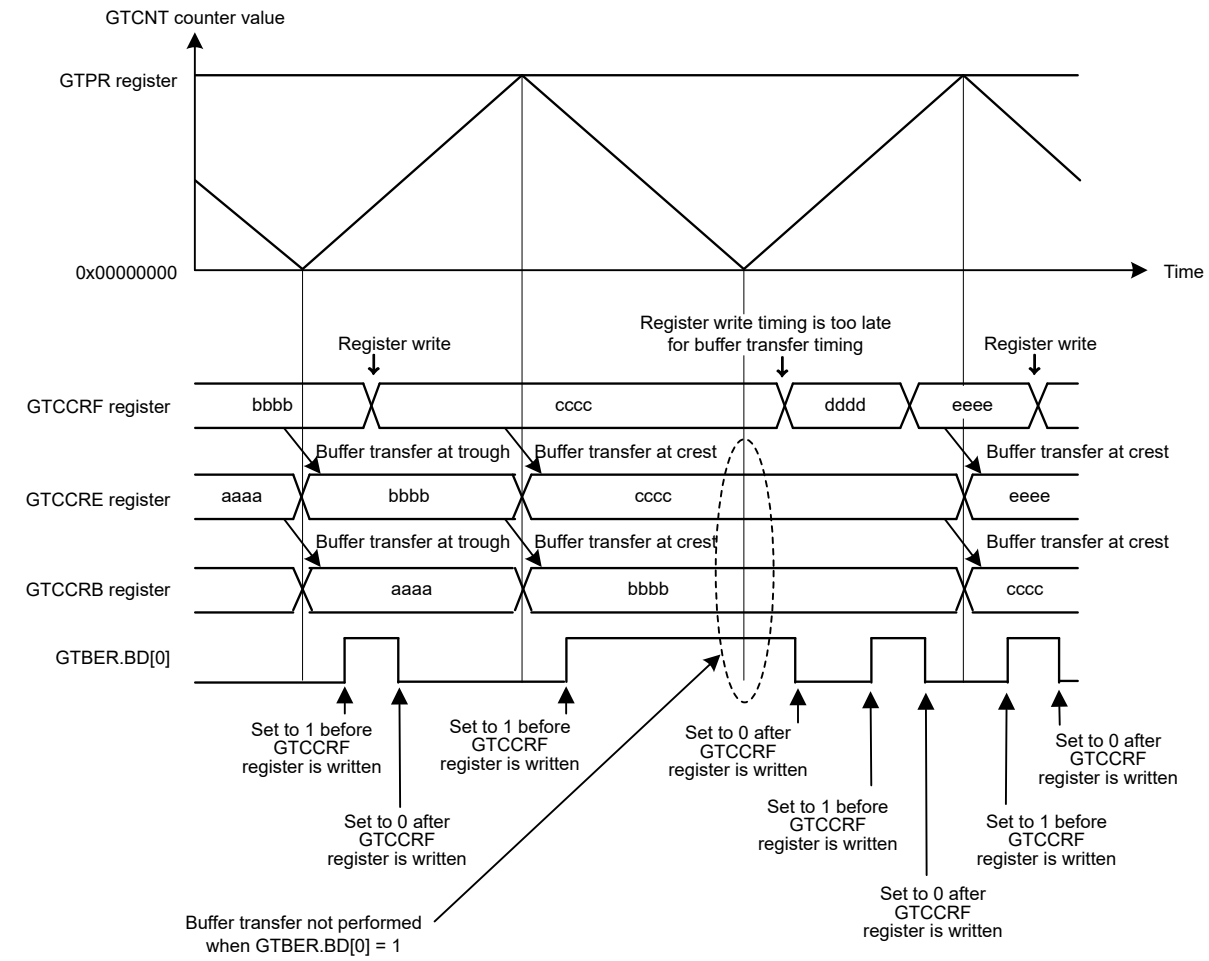
Protection using the GTWP register is only for write operations by the CPU. This protection does not cover updates to registers that occur in association with CPU writes.

20.8.2 Disabling of Buffer Operation

If the timing of the buffer register write is delayed relative to the timing for the buffer transfer, buffer operation can be suspended with the GTBER.BD[3], BD[2], BD[1] and BD[0] bits settings. Specifically, buffer transfer can be temporarily disabled even though a buffer transfer condition is generated during buffer register write, by setting the BD[3], BD[2], BD[1] and BD[0] bits to 1 (buffer operation disabled) before buffer register write, and setting the bits to 0 (buffer operation enabled) after completion of writing to all the buffer registers.

The BD[3], BD[2], BD[1] and BD[0] bits can be set on channel basis by writing directly to the GTBER register or it can be set to 0 simultaneously by setting the GTSECR register for multiple channels which were set by the GTSECSR register.

Figure 20.163 shows an example of operation for disabling buffer operation by writing to the GTBER register.



Note: The base address of registers are the same.

Figure 20.163 Example of operation for disabling buffer operation with triangle waves, double buffer operation, and buffer transfer at both troughs and crests

20.8.2.1 Simultaneous Control of Buffer Operations of Multiple Channels

The GTBER.BD bit can be set by writing directly to the GTBER register per channel or by making settings in the GTSECR register for multiple channels that have already set in the GTSECSR register.

Follow the procedure below to simultaneously set the GTBER.BD bits of multiple channels.

1. Select the channels for simultaneously setting by the GTSECSR register
   Set the GTSECSR register so that the values at the bit positions for the corresponding channels for simultaneously setting of the GTBER.BD bits become 1. All GTSECSR registers can be updated by writing to the GTSECSR register of any channel.
2. Simultaneously set the GTBER.BD bits by updating the GTSECR register
   In the GTSECR register, set the operation of the GTBER.BD bits (enabling or disabling of buffer operation) which are to be simultaneously set. Writing to a GTSECR register from any channel updates the GTBER.BD bits in all channels corresponding to the bits set as 1 in the GTSECSR register, in accordance with the value of the GTSECR register.

Figure 20.164 and Figure 20.165 show examples of simultaneously controlling the enabling or disabling of buffer operation for multiple channels.

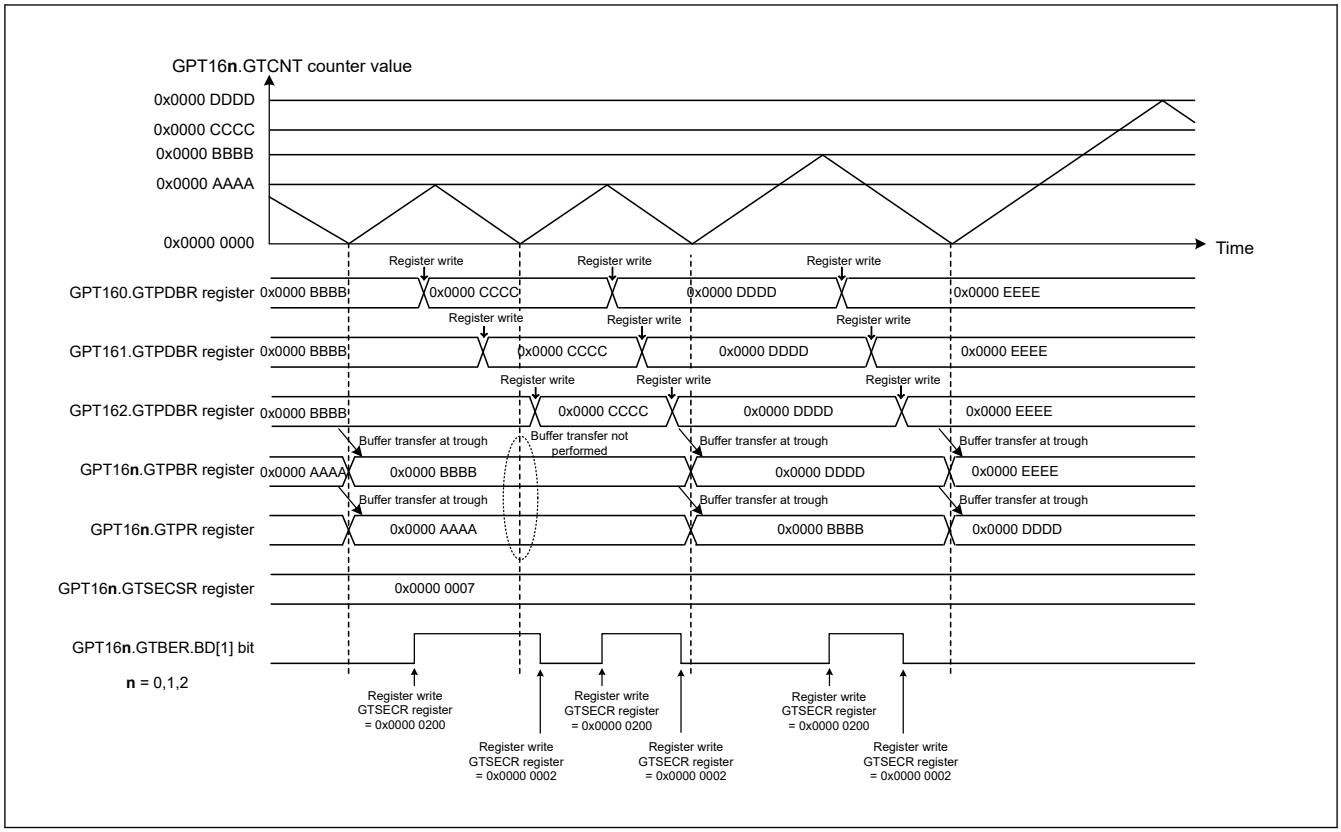


Figure 20.164 Example of multiple channel operation for disabling buffer operation (triangle-waves, double buffer operation)

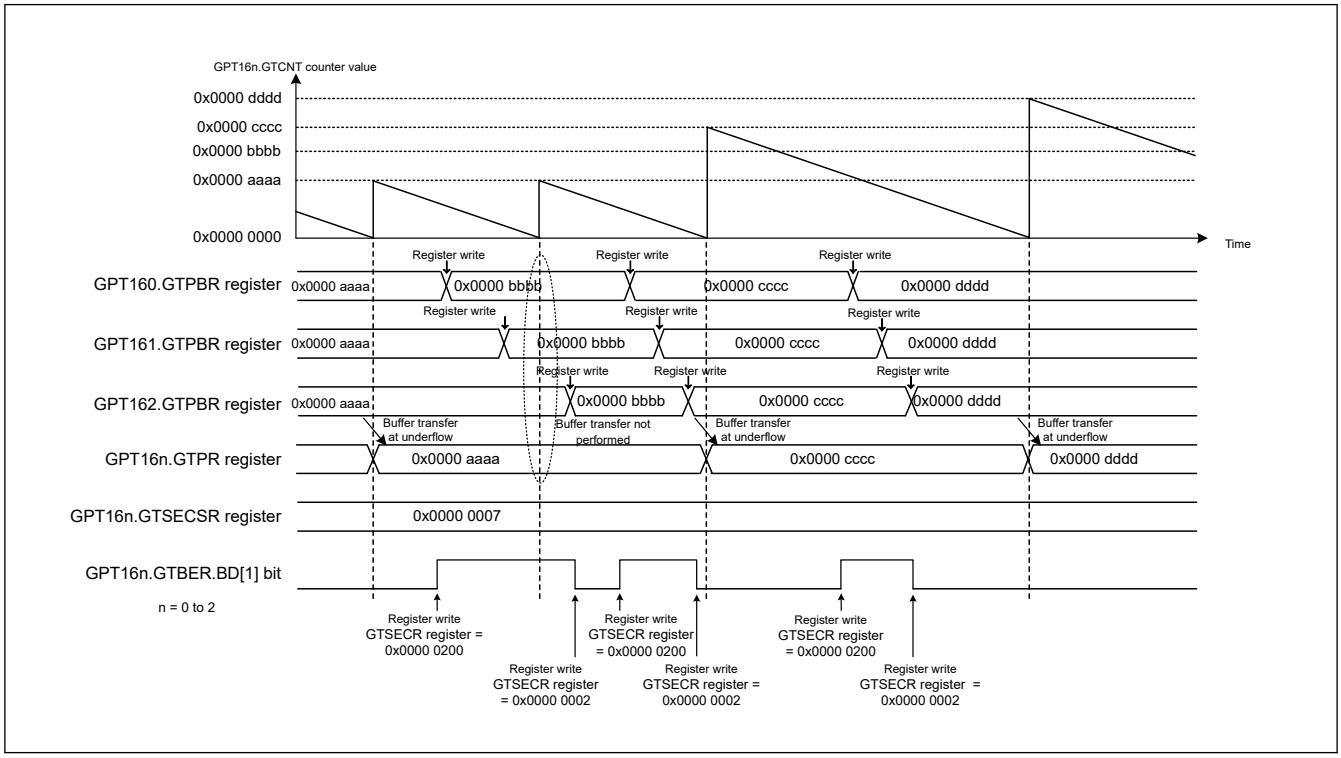


Figure 20.165 Example of multiple channel operation for disabling buffer operation (saw-waves, single buffer operation)

20.8.2.2 Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer

When a GTBER.DBRTECm (m = A, B) bit is set to 1 in saw-wave one-shot pulse mode or triangle-wave PWM mode 3, transfer from the intermediate buffer to the GTCCRm (m = A, B) register is repeated on a cyclic basis even while buffer transfer is disabled by the setting of the GTBER.BD[0] bit or by the buffer transfer extended skipping function.

(1) In saw-wave one-shot pulse mode

In saw-wave one-shot pulse mode, the compare match value for the first half of a waveform is stored in temporary register x (x = C, E) as the intermediate buffer for the GTCCRx (x = C, E) register and the compare match value for the second half of a waveform is stored in temporary register m (m = A, B) as the intermediate buffer for the GTCCRy (y = D, F) register, respectively, for compare match values during repeated buffer operation, and the given values are alternately transferred to the GTCCRm (m = A, B) registers.

Table 20.77 lists the types of buffer transfer of the GTCCR register during counting in saw-wave one-shot pulse mode.

While counting is stopped, the setting of the value in the temporary register is transferred through forcible buffer transfer. In forcible buffer transfer, the values of the GTCCRy (y = D, F) registers are transferred to temporary registers m (m = A, B), and the values of the GTCCRx (x = C, E) are transferred to temporary registers x (x = C, E) when the setting of the corresponding GTBER.DBRTECm (m = A, B) bit is 1, respectively.

When the setting of the GTBER.DBRTECm (m = A, B) bit is 1, values written by the CPU to the GTCCRm (m = A, B) registers are reflected as the values of temporary registers x (x = C, E).

Table 20.77 GTCCR buffer transfer operation in saw-wave one-shot pulse mode during GTCNT counting

GTBER.DBRTECm	Buffer transfer	Timing of transfer				
		GTCCRx ↓ GTCCRm	GTCCRx ↓ Temporary register x	Temporary register x ↓ GTCCRm	GTCCRy ↓ Temporary register m	Temporary register m ↓ GTCCRm
0	Transfer enabled period	Overflow or Underflow	No transfer	No transfer	Overflow or Underflow	GTCCRm compare match
	Transfer disabled period	No transfer	No transfer	No transfer	No transfer	No transfer
1	Transfer enabled period	Overflow or Underflow	Overflow or Underflow	No transfer	Overflow or Underflow	GTCCRm compare match
	Transfer disabled period	No transfer	No transfer	Overflow or Underflow	No transfer	GTCCRm compare match

Note: m = A, B
x = C, E
y = D, F

Figure 20.166 shows the operation of generating transfer-disabled period by extended buffer transfer skipping as an example of repeated double buffer operations when the GTCCR buffer transfer disabled in saw-wave one-shot pulse mode.

Figure 20.167 shows the operation of generating transfer-disabled period by updating the GTBER.BD[0] bit as an example of repeated double-buffer operations when the GTCCR buffer transfer is disabled in saw-wave one-shot pulse mode.

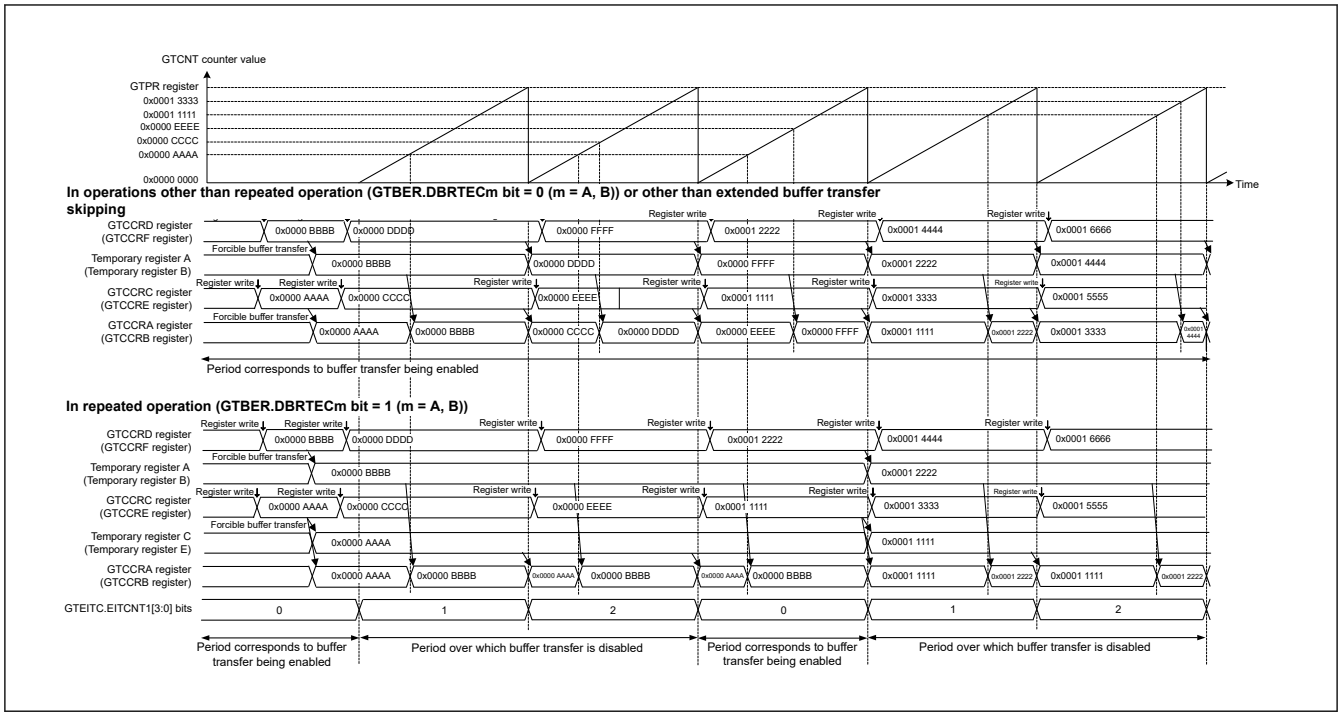


Figure 20.166 Example of repeated double-buffer operation when GTCCR buffer transfer is disabled (saw-wave one-shot pulse mode, using extended buffer transfer skipping, GTBER.BD[0] is constantly 0)

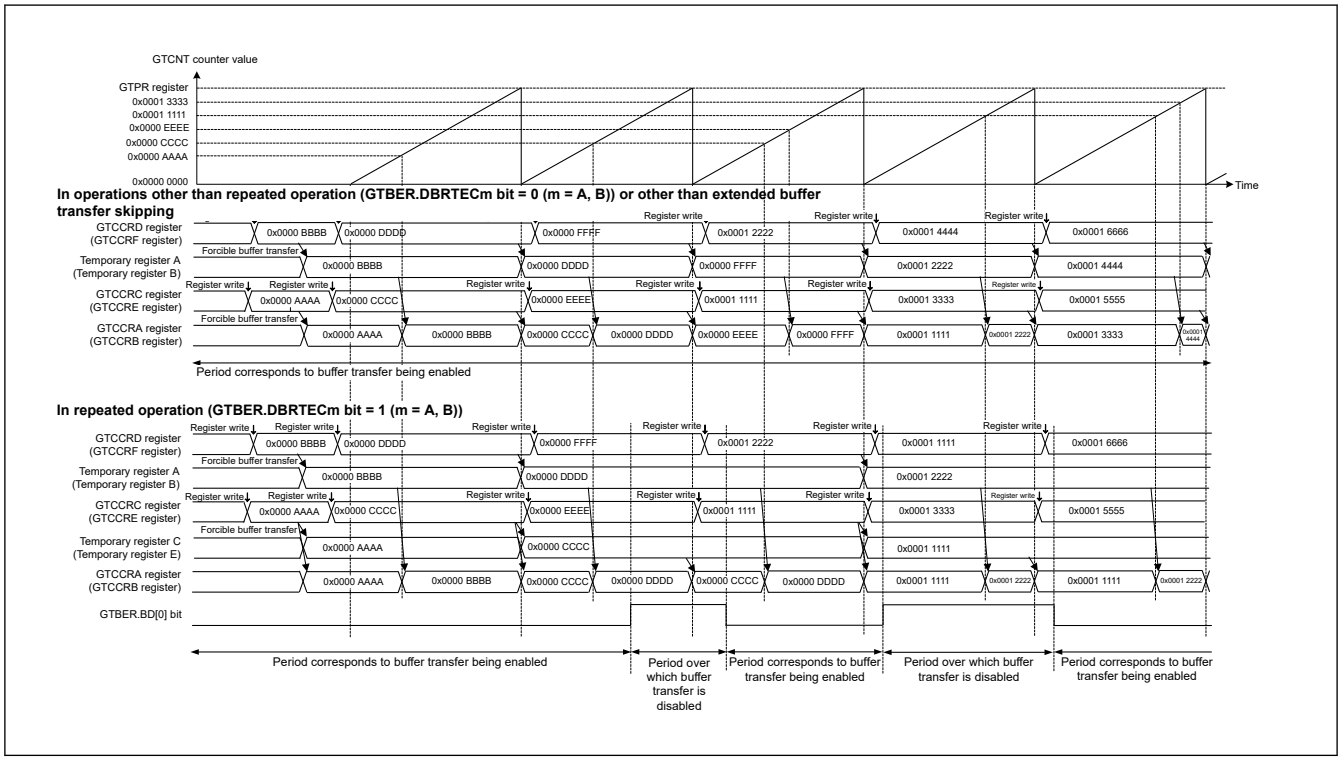


Figure 20.167 Example of repeated double-buffer operation when GTCCR buffer transfer is disabled (saw-wave one-shot pulse mode, updating the GTBER.BD[0] bit)

(2) In triangle-wave PWM mode 3

In triangle-wave PWM mode 3, the compare match value for the first half of a waveform is stored in the temporary register x (x = C, E) as the intermediate buffer for the GTCCRx (x = C, E) register and the compare match value for the second half of a waveform is stored in temporary register m (m = A, B) as the intermediate buffer for the GTCCRy (y = D, F) register,

respectively, for compare match values during repeated buffer operation, and the given values are alternately transferred to the GTCCRm (m = A, B) register.

Table 20.78 lists the types of buffer transfer of the GTCCR register during counting operation in triangle-wave PWM mode 3.

While counting is stopped, the setting of the value in the temporary register is transferred through forcible buffer transfer. In forcible buffer transfer, the values of the GTCCRy (y = D, F) registers are transferred to temporary registers m (m = A, B), and the values of the GTCCRx (x = C, E) are transferred to temporary registers x (x = C, E), when the setting of the corresponding GTBER.DBRTECm (m = A, B) bit is 1, respectively.

When the setting of the GTBER.DBRTECm (m = A, B) bit is set to 1, values written by the CPU to the GTCCRm (m = A, B) registers are reflected as the values of temporary registers x (x = C, E).

Table 20.78 GTCCR buffer transfer operation in triangle-wave PWM mode 3 during GTCNT counting

GTBER.DBRTECm	Buffer transfer	Timing of transfer				
		GTCCRx ↓ GTCCRm	GTCCRx ↓ Temporary register x	Temporary register x ↓ GTCCRm	GTCCRy ↓ Temporary register m	Temporary register m ↓ GTCCRm
0	Transfer enabled period	Trough	No transfer	No transfer	Trough	Crest
	Transfer disabled period	No transfer	No transfer	No transfer	No transfer	No transfer
1	Transfer enabled period	Trough	Trough	No transfer	Trough	Crest
	Transfer disabled period	No transfer	No transfer	Trough	No transfer	Crest

Note: m = A, B
x = C, E
y = D, F

Figure 20.168 shows the operation of generating transfer-disabled period by extended buffer transfer skipping as an example of repeated double buffer operations when the GTCCR buffer transfer disabled in triangle-wave PWM mode 3.

Figure 20.169 shows the operation of generating transfer-disabled period by updating the GTBER.BD[0] bit as an example of repeated double-buffer operations when the GTCCR buffer transfer is disabled in triangle-wave PWM mode 3.

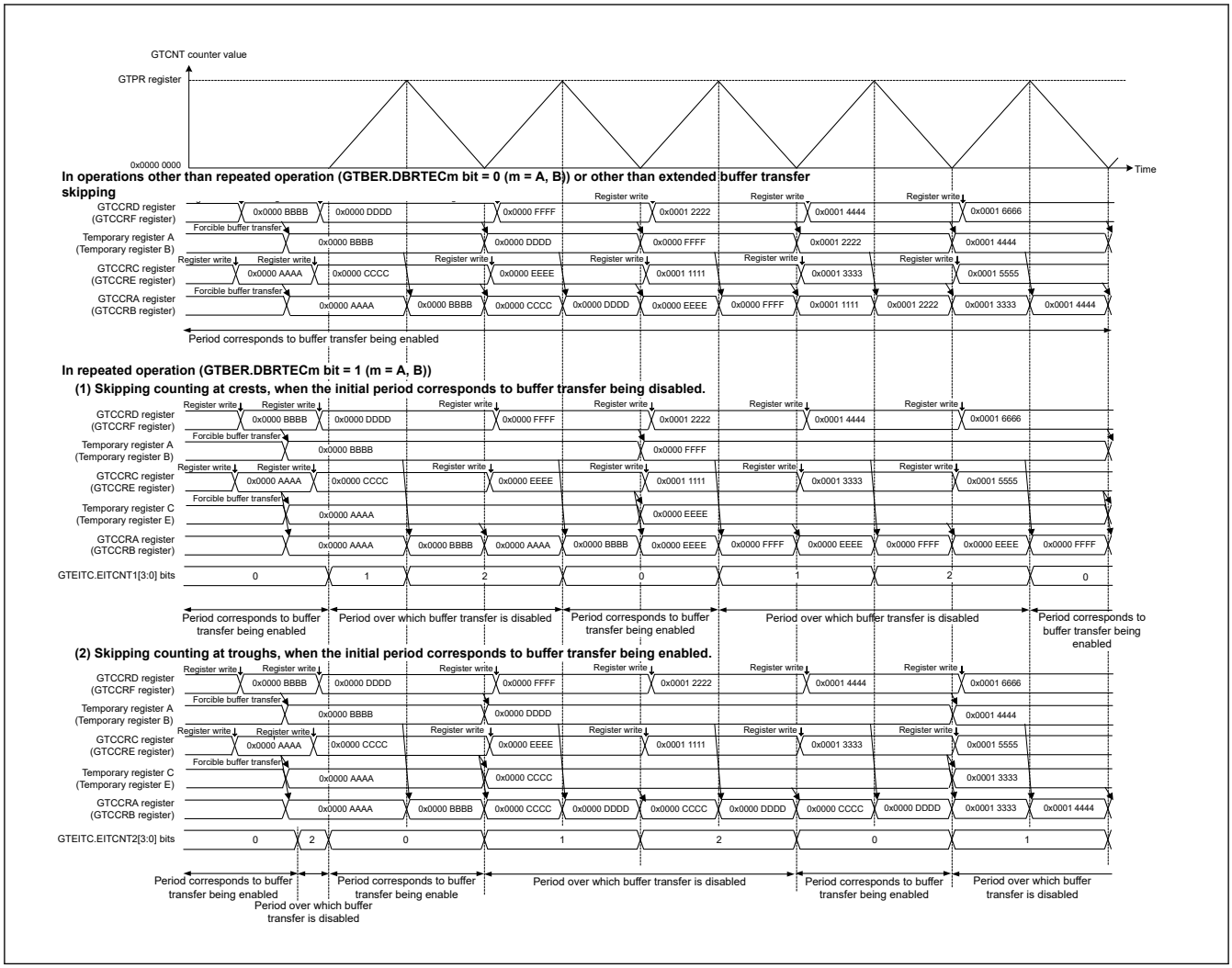


Figure 20.168 Example of repeated double-buffer operation when GTCCR buffer transfer is disabled (triangle-wave PWM mode 3, using extended buffer transfer skipping, GTBER.BD[0] is constantly 0)

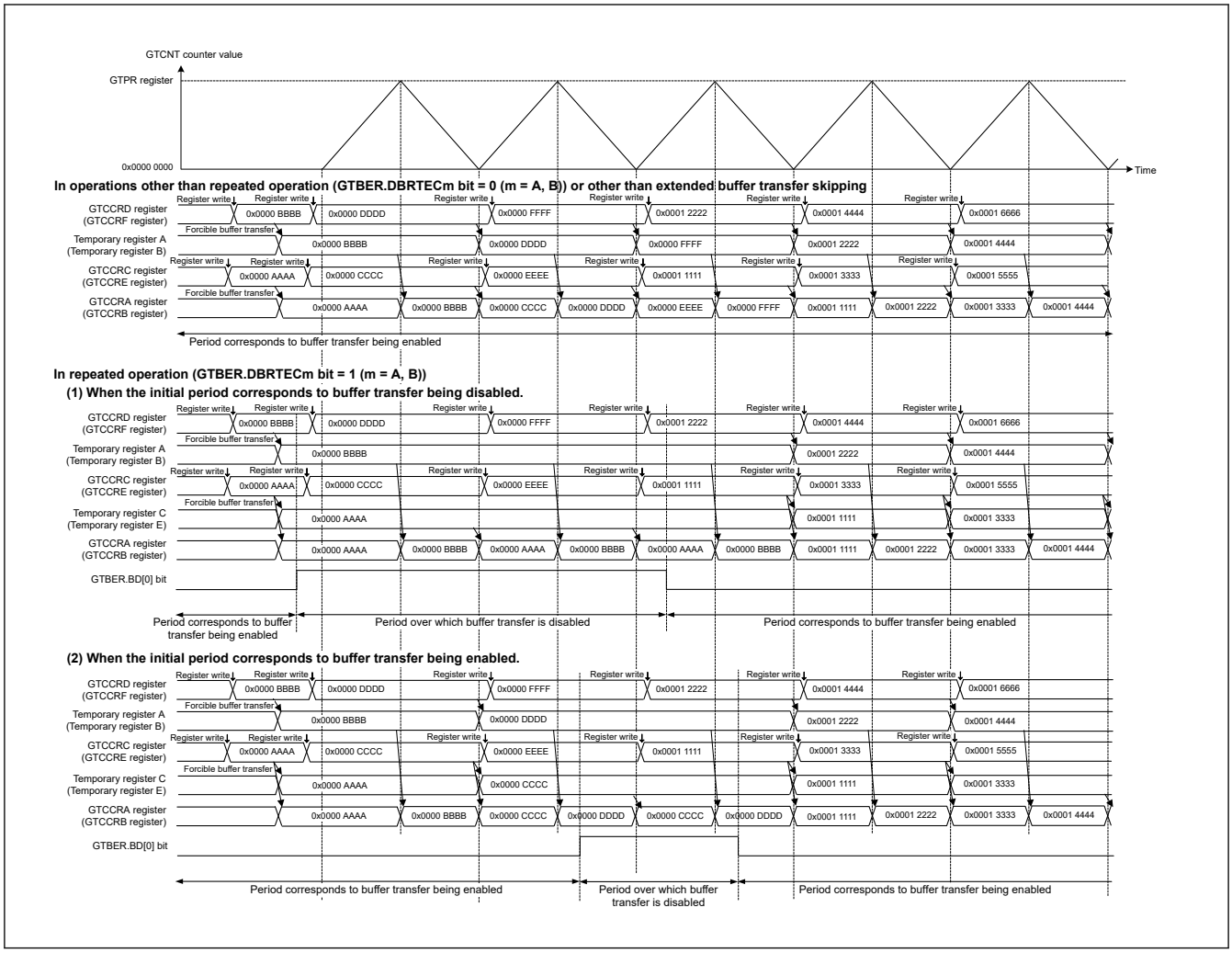


Figure 20.169 Example of repeated double-buffer operation when GTCCR buffer transfer is disabled (triangle-wave PWM mode 3, updating the GTBER.BD[0] bit)

20.8.3 GTIOCnm Pin Output Negate Control (n = 0 to 3, m = A, B)

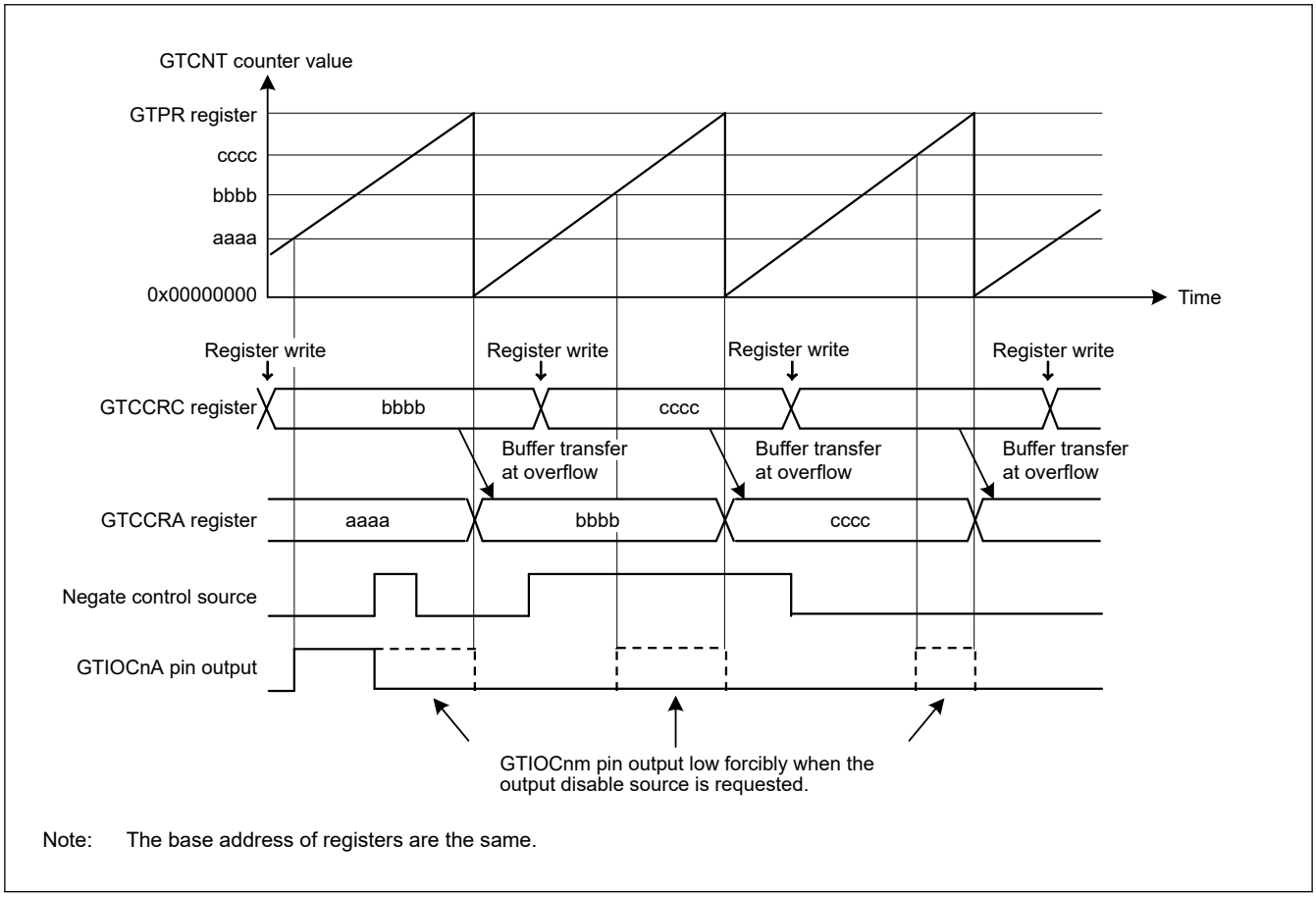
For protection from system failure, the output disable control that changes the GTIOCnm pin output value forcibly is provided for GTIOCnm pin output by the request of output disable from POEG. Output protection is required when a dead time error or the same output level being on the GTIOCnA and GTIOCnB pins is detected. GPT detects this condition and generates output disable requests to POEG according to the setting of the output disable request permission bits, such as GTINTAD.GRPDTE, GTINTAD.GRPABH, GTINTAD.GRPABL. After the POEG performs the logical OR of the output disable request from each channel and the output disable request from the external input, the POEG generates output disable requests to GPT.

One output disable signal (representing the shared output disable request signal of the GTIOCnA pin and the GTIOCnB pin) out of 4 output disable requests generated by the POEG is selected by setting GTINTAD.GRP[1:0]. The status of the selected disable output request is monitored by reading the GTST.ODF bit. The output level during output disable is set based on the GTIOR.OADF[1:0] bits for the GTIOCnA pin and the GTIOR.OBDF[1:0] setting for the GTIOCnB pin.

The change to the output disable state is performed asynchronously by generating the output disable request from the POEG. The release of the output disable state is performed at end of cycle by terminating the output disable request. It is after 3 GTCLK at shortest when the output disable condition is released after the output disable request becomes no longer satisfied. To reliably control output disabling, clear the flag of POEG for which the condition for the request to disable the output is no longer satisfied after 4 cycles of GTCLK.

When event count is performed or when the operating mode is saw-wave PWM mode 2 or when the output disable state should be released immediately without waiting for end of cycle, GTIOR.OADF[1:0] should be set to 00b (for GTIOCnA pin) or GTIOR.OBDF[1:0] should be set to 00b (for the GTIOCnB pin).

Figure 20.170 shows an example of the GTIOCnm pin output disable control operation. (n = 0 to 9, m = A, B)



Note: The base address of registers are the same.

Figure 20.170 Example of GTIOCnm pin output disable control operation in saw-wave up-counting, buffer operation, active level 1, high output at GTCCRA compare match, low output at cycle end, and low output at output disable (n = 0 to 3, m = A, B)

20.8.4 Output Protection Function for GTIOCnm Pin Output (n = 0 to 2, m = A, B)

To prepare for a case when an incorrect value (0x00000000 or a value greater than or equal to the GTPR register value) is set in the GTCCRA register, the output protection function for the GTIOCnm pin output (disabling function) is activated when the automatic dead time is set (GTDTCR.TDE bit = 1) in triangle-wave PWM mode.

The status of the output protection function can be read from the GTSOS.SOS[1:0] bits.

Figure 20.171 shows the output protection function state transition.

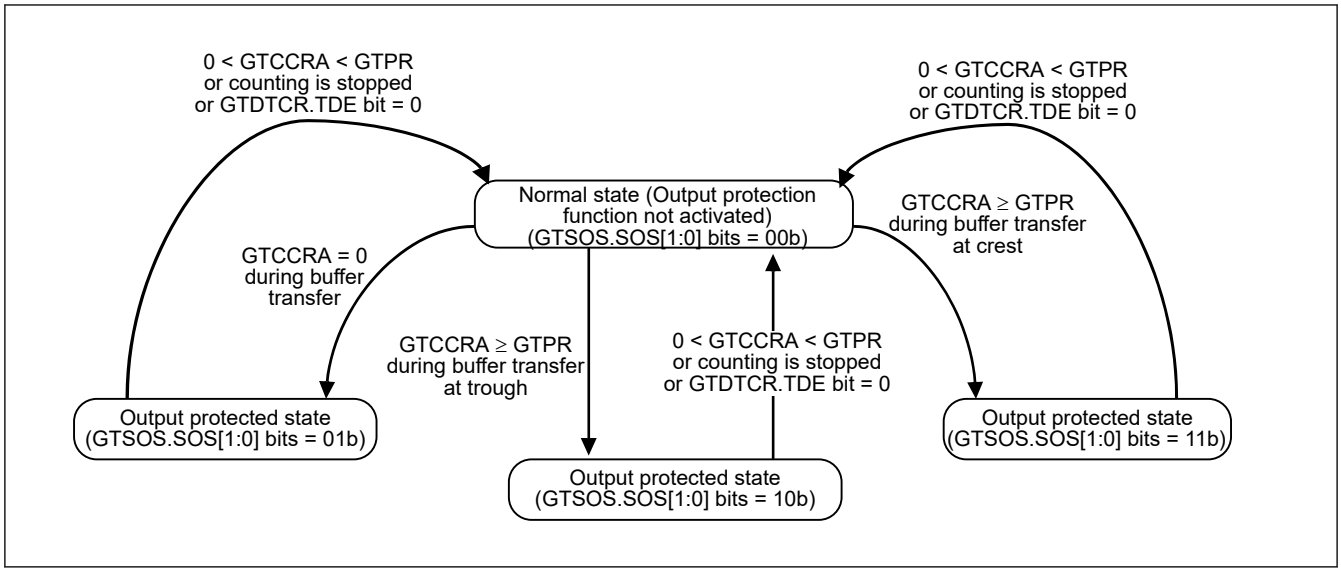
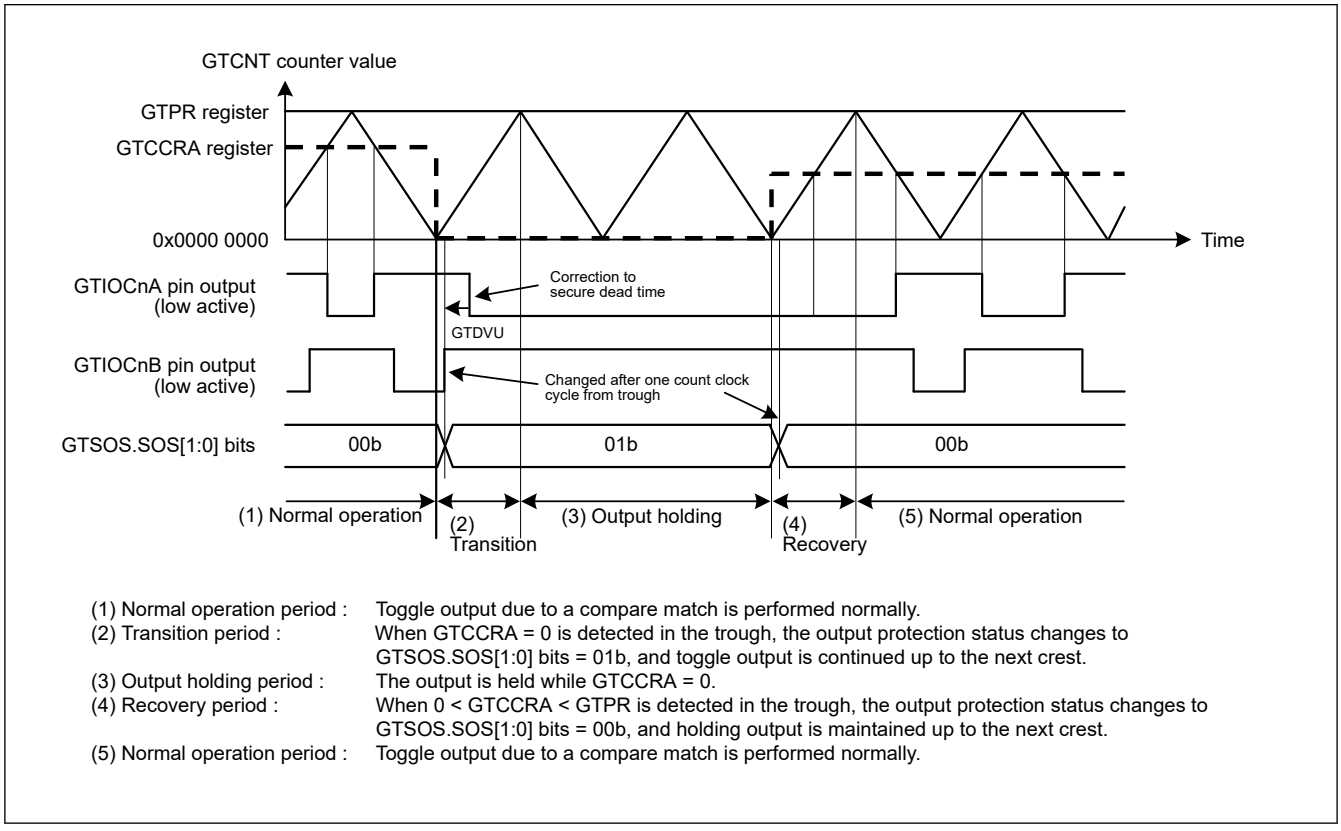


Figure 20.171 Output Protection Function

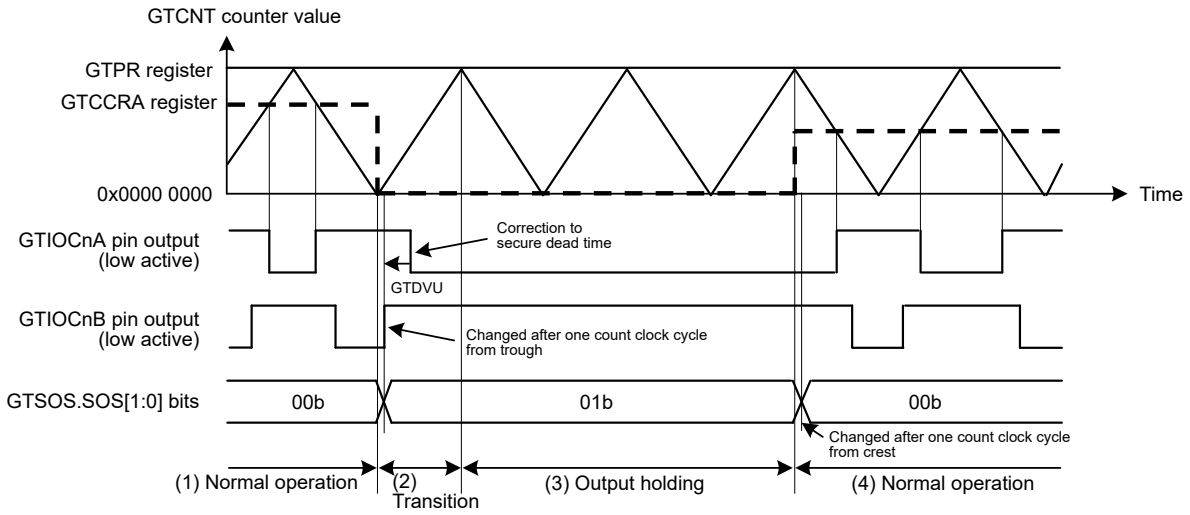
(1) Output Protection Function When the GTCCRA Register is Set to 0x0000 0000 during Buffer Transfer

Figure 20.172 and Figure 20.173 show examples of output protection function operation when the GTCCRA register is set to 0x0000 0000 during buffer transfer at troughs, and Figure 20.174 and Figure 20.175 show examples when the GTCCRA register is set to 0x0000 0000 during buffer transfer at crests.



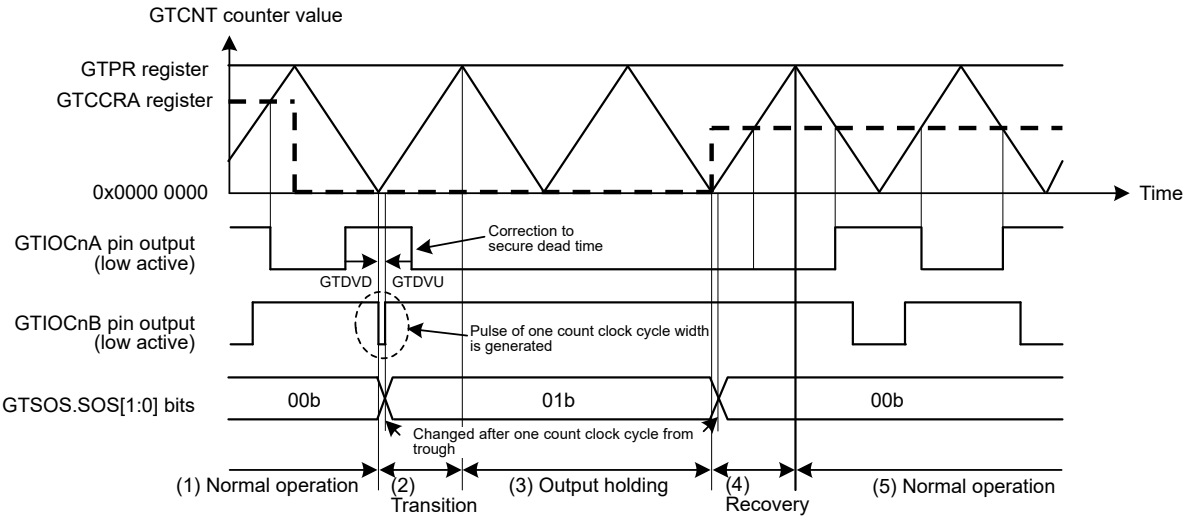
(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Transition period : When GTCCRA = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding period : The output is held while GTCCRA = 0.
(4) Recovery period : When 0 < GTCCRA < GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and holding output is maintained up to the next crest.
(5) Normal operation period : Toggle output due to a compare match is performed normally.

Figure 20.172 Example of output protection function operation when GTCCRA is set to 0x0000 0000 during buffer transfer at troughs (restored to 0 < GTCCRA < GTPR during buffer transfer at troughs, active level: low) (n = 0 to 2)



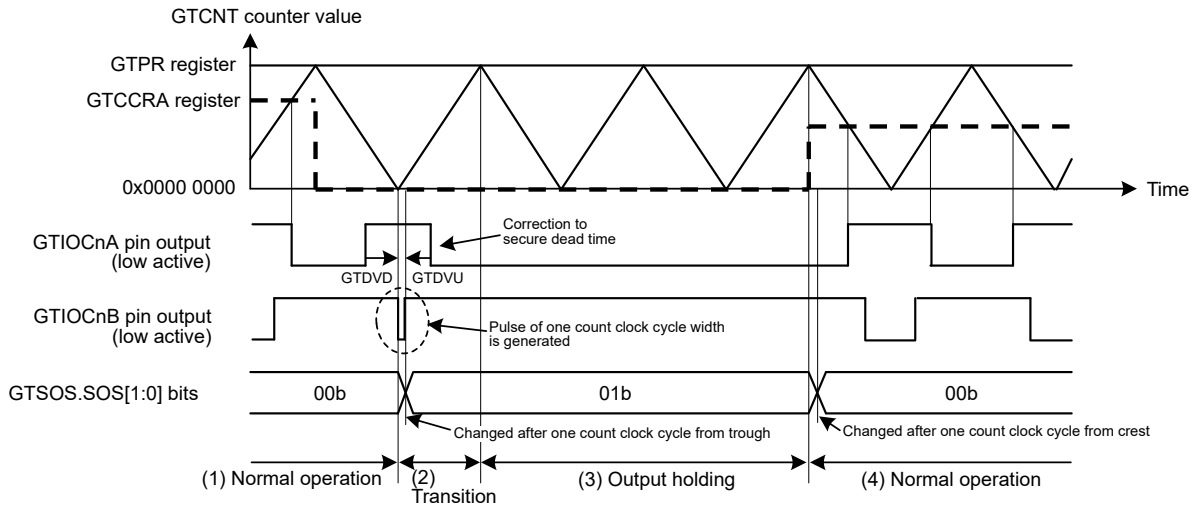
(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Transition period : When GTCCRA = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding period : The output is held while GTCCRA = 0.
(4) Normal operation period : When 0 < GTCCRA < GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 20.173 Example of output protection function operation when GTCCRA is set to 0x0000 0000 during buffer transfer at troughs (restored to 0 < GTCCRA < GTPR during buffer transfer at crests, active level: low) (n = 0 to 2)



(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Transition period : When GTCCRA = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding : The output is held while GTCCRA = 0.
(4) Recovery period : When 0 < GTCCRA < GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and holding output is maintained up to the next crest.
(5) Normal operation period : Toggle output due to a compare match is performed normally.

Figure 20.174 Example of output protection function operation when GTCCRA is set to 0x0000 0000 during buffer transfer at crests (restored to 0 < GTCCRA < GTPR during buffer transfer at troughs, active level: low) (n = 0 to 2)

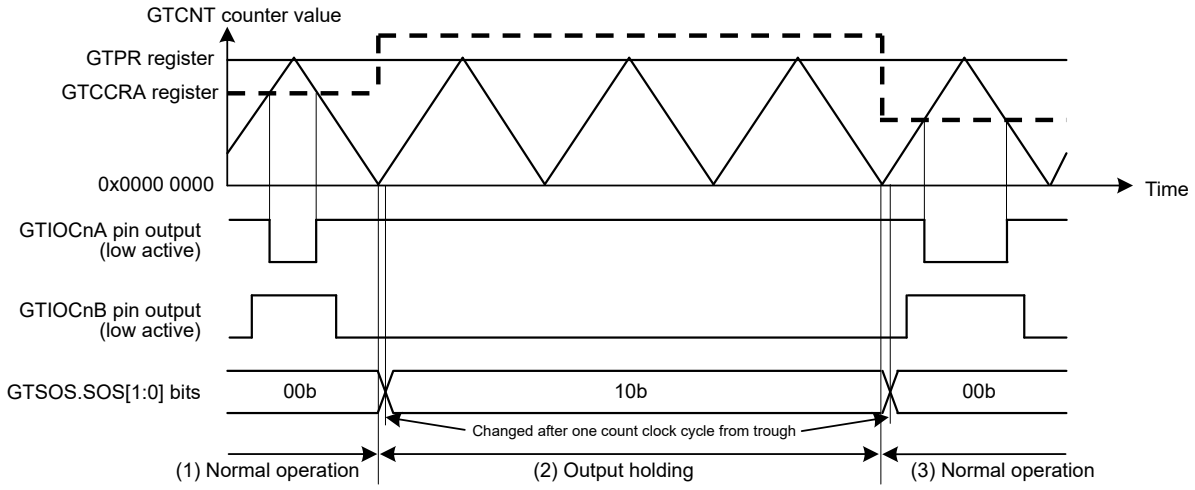


(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Transition period : When GTCCRA = 0 is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 01b, and toggle output is continued up to the next crest.
(3) Output holding period : The output is held while GTCCRA = 0.
(4) Normal operation period : When 0 < GTCCRA < GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 20.175 Example of output protection function operation when the GTCCRA is set to 0x0000 0000 during buffer transfer at crests (restored to 0 < GTCCRA < GTPR during buffer transfer at crests, active level: low) (n = 0 to 2)

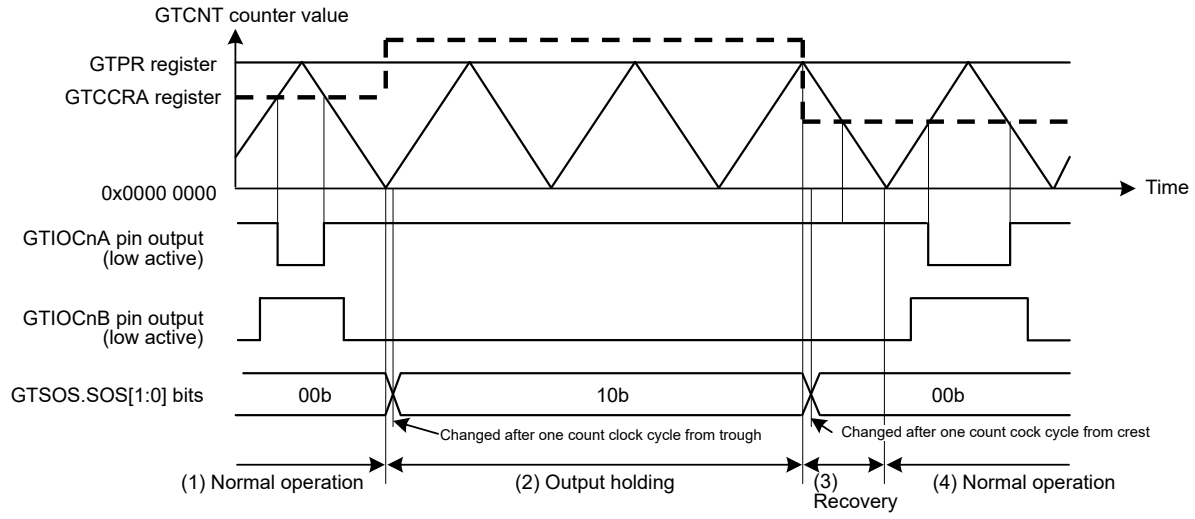
(2) Output Protection Function When GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Troughs

Figure 20.176 and Figure 20.177 show examples of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at troughs.



(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Output holding period : When GTCCRA ≥ GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 10b, and the output is held while GTCCRA ≥ GTPR.
(3) Normal operation period : When 0 < GTCCRA < GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 20.176 Example of output protection function operation when GTCCRA ≥ GTPR is set during buffer transfer at troughs (restored to 0 < GTCCRA < GTPR during buffer transfer at troughs, active level: low) (n = 0 to 2)

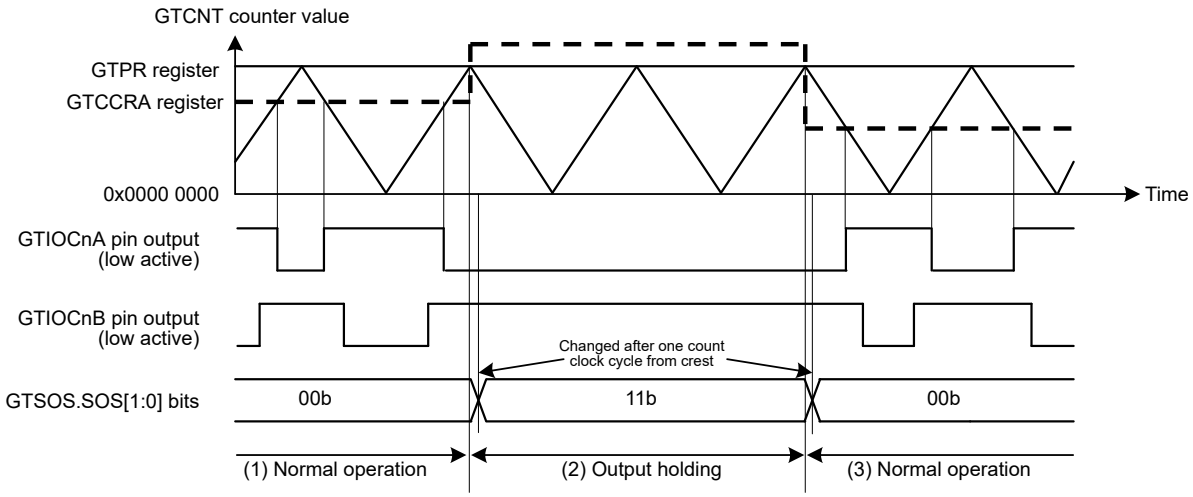


(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Output holding period : When GTCCRA ≥ GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 10b, and the output is held while GTCCRA ≥ GTPR.
(3) Recovery period : When 0 < GTCCRA < GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and holding output is maintained up to the next trough.
(4) Normal operation period : Toggle output due to a compare match is performed normally.

Figure 20.177 Example of output protection function operation when GTCCRA ≥ GTPR is set during buffer transfer at troughs (restored to 0 < GTCCRA < GTPR during buffer transfer at crests, active level: low) (n = 0 to 2)

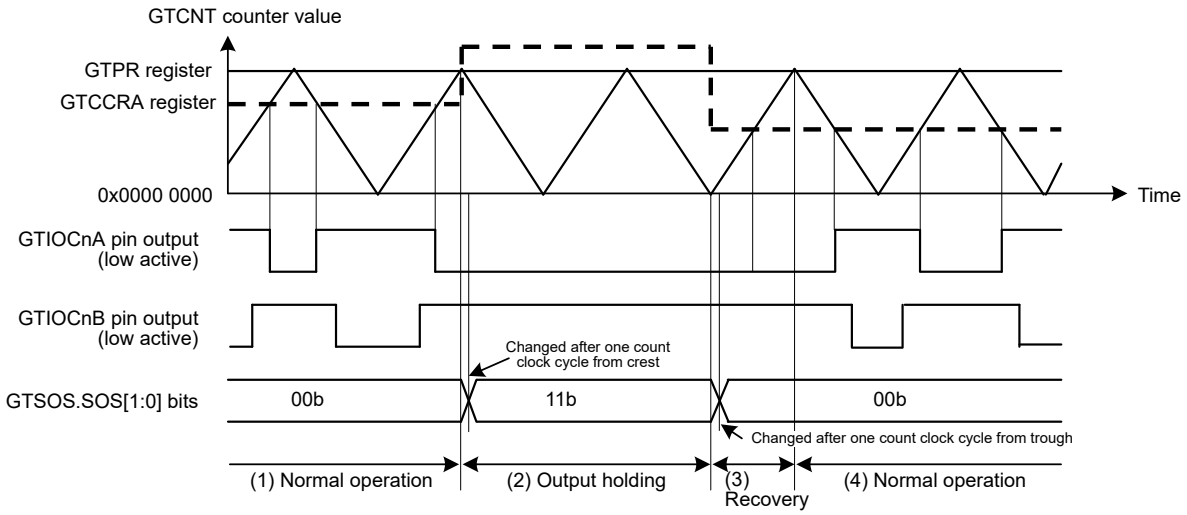
(3) Output Protection Function When GTCCRA ≥ GTPR is Set during Buffer Transfer at Crests

Figure 20.178 and Figure 20.179 show examples of output protection function operation when GTCCRA ≥ GTPR is set during buffer transfer at crests.



(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Output holding period : When GTCCRA ≥ GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 11b, and the output is held while GTCCRA ≥ GTPR.
(3) Normal operation period : When 0 < GTCCRA < GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 20.178 Example of output protection function operation when GTCCRA ≥ GTPR is set during buffer transfer at crests (restored to 0 < GTCCRA < GTPR during buffer transfer at crests, active level: low) (n = 0 to 2)



(1) Normal operation period : Toggle output due to a compare match is performed normally.
(2) Output holding period : When GTCCRA ≥ GTPR is detected in the crest, the output protection status changes to GTSOS.SOS[1:0] = 11b, and the output is held while GTCCRA ≥ GTPR.
(3) Recovery period : When 0 < GTCCRA < GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] = 00b, and holding output is maintained up to the next crest.
(4) Normal operation period : Toggle output due to a compare match is performed normally.

Figure 20.179 Example of output protection function operation when GTCCRA ≥ GTPR is set during buffer transfer at crests (restored to 0 < GTCCRA < GTPR during buffer transfer at troughs, active level: low) (n = 0 to 2)

(4) Restricted Specification of Output Protection Function

Even if an incorrect value (0x0000 0000 or a value greater than or equal to the GTPR register value) is set in the GTCCRA register during count operation, the output protection functions in a specific way such that one of the positive- and negative-phase outputs becomes non-active. However, if the following condition is not satisfied, the output protection does not operate normally.

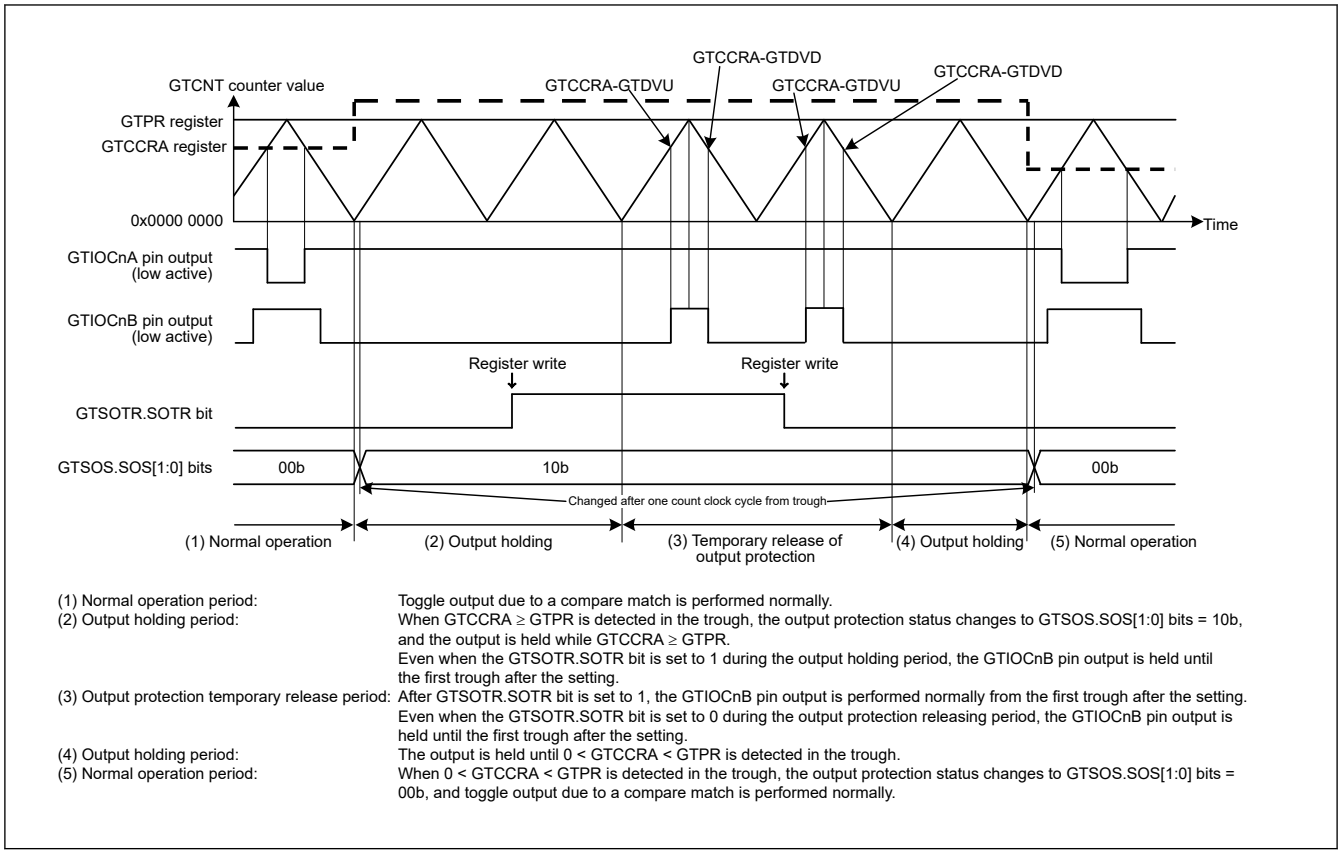
- When the GTCCRA register value at the start of count operation is greater than 0x0000 0000, and less than the setting value of the GTPR register

(5) Temporary Release of Output Protection Function

When the GTSOS.SOS[1:0] bits = 10b (protected state in which GTCCRA register ≥ GTPR register has occurred during transfer at trough), the protected state of the GTIOCnB pin output can be temporarily released by setting the GTSOTR.SOTR bit to 1. The SOS[1:0] bits retain 10b even if the output protection function is released.

When the SOTR bit is set to 0, the GTIOCnB pin output protection can be restarted.

Figure 20.180 shows examples of the operation of temporary release of output protection when the setting of the GTCCRA register ≥ GTPR register during buffer transfer at troughs.



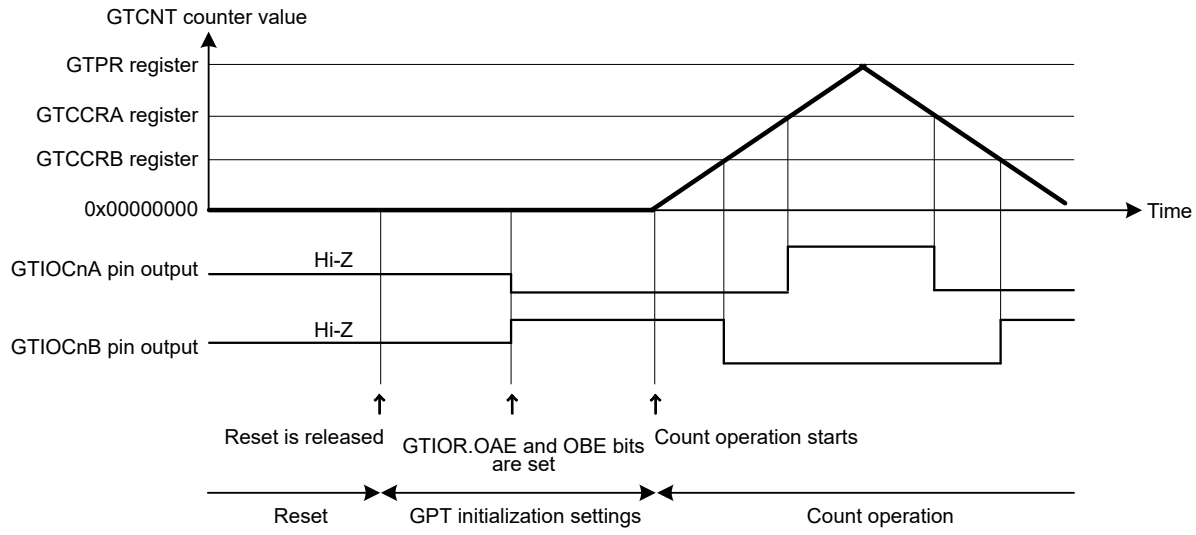
(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA ≥ GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 10b, and the output is held while GTCCRA ≥ GTPR.
Even when the GTSOTR.SOTR bit is set to 1 during the output holding period, the GTIOCnB pin output is held until the first trough after the setting.
(3) Output protection temporary release period: After GTSOTR.SOTR bit is set to 1, the GTIOCnB pin output is performed normally from the first trough after the setting.
Even when the GTSOTR.SOTR bit is set to 0 during the output protection releasing period, the GTIOCnB pin output is held until the first trough after the setting.
(4) Output holding period: The output is held until 0 < GTCCRA < GTPR is detected in the trough.
(5) Normal operation period: When 0 < GTCCRA < GTPR is detected in the trough, the output protection status changes to GTSOS.SOS[1:0] bits = 00b, and toggle output due to a compare match is performed normally.

Figure 20.180 Example of temporary release of output protection when the setting of the GTCCRA ≥ GTPR during buffer transfer at troughs (restored to 0 < GTCCRA < GTPR during buffer transfer at troughs, active level: low) (n = 0 to 2)

20.9 Initialization Method of Output Pins

20.9.1 Pin Settings after Reset

The GPT registers are initialized at a reset. Start counting after selecting the port pin function with the PmnPFS register, setting GTIOR.OAE and GTIOR.OBE bits, and outputting the GPT function to external pins.



[Setting examples]
GTIOR.GTIOA[4:0] bits: Initial low output, output retained at cycle end, output toggled at compare match
GTIOR.GTIOB[4:0] bits: Initial high output, output retained at cycle end, output toggled at compare match

Note: The base address of registers are the same.

Figure 20.181 Example of pin settings after reset

20.9.2 Pin Initialization Due to Error during Operation

If an error occurs during GPT operation, the following four types of pin control can be performed before pin initialization:

- Set the OAHLD and OBHLD bits in GTIOR to 1 and retain the outputs at count stop
- Set the OAHLD and OBHLD bits in GTIOR to 0, specify arbitrary output values at OADFLT and OBDFLT in GTIOR, and output the arbitrary values at count stop
- Set the pin to output an arbitrary value as a general output port by setting the PDR, PODR registers and PmnPFS.PMR bit of the I/O port in advance. Set the OAE and OBE bits in GTIOR to 0, and the control bit associated with the pin in the PMR to 0 to allow arbitrary values to be output from the pin set as a general output port when an error occurs.
- Drive the output to a high impedance state using the POEG function.

If the automatic dead time setting is made, clear the GTDTCR.TDE bit to 0 after counting stops. When counting stops, only the values of registers that are changed by a GPT external source change. If counting is resumed, operation continues from where it stopped. If counting is stopped, the registers must be initialized before counting starts.

20.10 Usage Notes

20.10.1 Module-Stop Function Setting

The Module Stop Control Register can enable or disable GPT operation. The GPT is initially stopped after a reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

20.10.2 GTCCRn Settings During Compare Match Operation (n = A to F)

(1) When automatic dead time setting is made in triangle-wave PWM mode

The GTCCRA register must satisfy all of the following conditions:

- GTDVU < GTCCRA
- GTCCRA > GTDVD
- 0 < GTCCRA < GTPR

When the setting of GTCCRA = 0 or GTCCRA ≥ GTPR is made for the GTCCRA register during count operation, the output protection function is activated. However, if the following condition is not satisfied, the output protection function does not work normally:

- The value of the GTCCRA register at the start of counting is larger than 0 and less than GTPR.

For details, see section 20.8.4. Output Protection Function for GTIOCnm Pin Output (n = 0 to 2, m = A, B).

(2) When automatic dead time setting is not made in triangle-wave PWM mode

The GTCCRA register must be set within the range of 0 < GTCCRA < GTPR. If GTCCRA = 0 or GTCCRA = GTPR is set, a compare match occurs within the cycle only when GTCCRA = 0 or GTCCRA = GTPR is satisfied. When GTCCRA > GTPR, no compare match occurs.

Similarly, GTCCRB must be set within the range of 0 < GTCCRB < GTPR. If GTCCRB = 0 or GTCCRB = GTPR is set, a compare match occurs within the cycle only when GTCCRB is 0 or GTCCRB = GTPR is satisfied. When GTCCRB > GTPR, no compare match occurs.

(3) When automatic dead time setting is made in saw-wave one-shot pulse mode

The GTCCRC and GTCCRD registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, the correct output waveforms with secured dead time may not be obtained.

- In up-counting: GTCCRC < GTCCRD, GTCCRC > GTDVU, GTCCRD < GTPR - GTDVD (in GPT160 to GPT162), GTCCRD < GTPR – GTDVU (in GPT163)
- In down-counting: GTCCRC > GTCCRD, GTCCRC < GTPR - GTDVU, GTCCRD > GTDVD (in GPT160 to GPT162), GTCCRD > GTDVU (in GPT163)

(4) When automatic dead time setting is not made in saw-wave one-shot pulse mode

The GTCCRC and GTCCRD registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRC < GTCCRD < GTPR
- In down-counting: GTPR > GTCCRC > GTCCRD > 0

Similarly, GTCCRE and GTCCRF must be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRE < GTCCRF < GTPR
- In down-counting: GTPR > GTCCRE > GTCCRF > 0.

(5) In saw-wave PWM mode 1

The GTCCRA register must be set with the range of 0 < GTCCRA < GTPR. If GTCCRA = 0 or GTCCRA = GTPR is set, a compare match occurs within the cycle only when GTCCRA = 0 or GTCCRA = GTPR is satisfied. If GTCCRA > GTPR is set, no compare match occurs.

Similarly, GTCCRB must be set with the range of 0 < GTCCRB < GTPR. If GTCCRB = 0 or GTCCRB = GTPR is set, a compare match occurs within the cycle only when GTCCRB = 0 or GTCCRB = GTPR is satisfied. If GTCCRB > GTPR is set, no compare match occurs.

(6) In complementary PWM mode 1, 2, 3

The GTCCRn register must be set with the range of 0 ≤ GTCCRn ≤ GTPR + GTDVU.

(7) In complementary PWM mode 4

In single buffer operation, the GTCCRn register must be set with the range of 0 ≤ GTCCRn ≤ GTPR + GTDVU.

In double buffer operation, the GTCCRn register must be set with the range of GTDVU < GTCCRn < GTPR.

20.10.3 Setting Range for GTCNT Counter

Other than the Saw-wave PWM mode 2 and Complementary PWM mode, the GTCNT counter register must be set with the range of 0 ≤ GTCNT ≤ GTPR.

If GTCNT counter > GTPR register is set and the counter starts, the count operation performs the following three cases.

1. In triangle-waves
   After the start of counting, GTCNT counter = GTPR register and GTST.TCUF flag = 0 are set, the counter performs down-counting.
2. In saw-wave up-counting
   After the start of counting, GTCNT counter = 0 is set, the counter performs up-counting.
3. In saw-wave down-counting
   After the start of counting, once GTCNT counter = 0 is set and GTCNT counter = GTPR register is set, the counter performs down-counting.

20.10.4 Setting Range of GTPBR and GTPDBR in Complementary PWM Mode

In complementary PWM modes 1, 3, and 4, if GTPR buffer transfer occurs at the end of the crest section, set GTPBR and GTPDBR value to the restricted range (GTPBR ≥ GTPR - GTDVU, GTPDBR ≥ GTPR - GTDVU) so that the GTPR value after the buffer transfer is not less than the GTCNT counter value of the master channel at the end of the crest section. When GTPR buffer transfer occurs at the end of the trough section or clearing counter, there is no restriction on the setting range of GTPBR and GTPDBR.

20.10.5 Starting and Stopping the GTCNT Counter

The control timing of starting and stopping the GTCNT counter by the GTCR.CST bit synchronizes the count clock that is selected in GTCR.TPCS[3:0]. When GTCR.CST is updated, the GTCNT counter starts/stops after a count clock that is selected in GTCR.TPCS[3:0]. Therefore, an event generated before the GTCNT counter actually starts is ignored, resulting in situations in which an event is accepted or an interrupt occurs after GTCR.CST is set to 0.

20.10.6 Priority Order of Each Event

(1) GTCNT register

Table 20.79 shows a priority order of events updating the GTCNT register.

Table 20.79 Priority order of sources updating GTCNT

Source updating GTCNT	Priority order
Writing by CPU (writing to GTCNT/GTCLR)	High
Clear by hardware sources set in GTCSR	↑
Count up or down by hardware sources set in GTUPSR/GTDNSR	↑
Count operation	Low

If up-counting and down-counting by hardware sources occur at the same time, the GTCNT counter value does not change. When there is a conflict between updating the GTCNT register and reading by the CPU, pre-update data is read.

(2) GTCR.CST bit

When there is a conflict between starting/stopping by hardware sources set in the GTSSR/GTPSR registers and writing by the CPU (writing to GTCR/GTSTR/GTSTP registers), the writing by CPU has priority over the starting/stopping by hardware sources.

In case that stop by the period count function conflicts with start by the CPU writing (GTCR register writing/GTSTR register writing), the period count function is finished with setting the GTST.PCF flag. The CST bit is not changed and the GTCNT continues to count.

When there is a conflict between starting by hardware sources set in the GTSSR register and stopping by hardware sources set in GTPSR register, the GTCR.CST bit value does not change. When there is a conflict between updating the GTCR.CST bit and reading by the CPU (reading from GTCR/GTSTR/GTSTP registers), pre-update data is read.

(3) GTCCRm registers (m = A to F)

When there is a conflict between input capture/buffer transfer operation and writing to the GTCCRm registers, the writing to GTCCRm registers has priority over input capture/buffer transfer operation. When there is a conflict between input capture and writing to the counter register by the CPU or updating the counter register by hardware sources, the pre-update counter value is captured. When there is a conflict between updating the GTCCRm registers and reading by the CPU, pre-update data is read.

(4) GTPR register

When there is a conflict between buffer transfer operation and writing to the GTPR register, writing to GTPR register has priority over buffer transfer operation. When there is a conflict between updating GTPR register and reading by the CPU, pre-update data is read.

(5) GTADTRm registers (m = A, B)

When there is a conflict between buffer transfer operation and writing to GTADTRm register, writing to GTADTRm register has priority over buffer transfer operation.

When there is a conflict between updating the GTADTRm register and reading by the CPU, pre-update data is read.

(6) GTDVm registers (m = U, D)

When there is a conflict between buffer transfer operation and writing to GTDVm register, writing to GTDVm register has priority over buffer transfer operation.

When there is a conflict between updating the GTDVm register and reading by the CPU, pre-update data is read.

(7) GTIOR.GTIOm registers (m = A, B)

When there is a conflict between buffer transfer operation and writing to GTIOR.GTIOm register, writing to GTIOR.GTIOm register has priority over buffer transfer operation.

When there is a conflict between updating the GTIOR.GTIOm and reading by the CPU, pre-update data is read.

20.10.7 Counter Clear Operation in Complementary PWM Mode

During complementary PWM mode, the counter clear at the end of the trough section including the initial output section (GTCNT count value of the master channel matches the value of GTDVU register) is prohibited. When the counter is cleared in complementary PWM mode, adjust the timing so that counter clearing does not occur at the end of the trough section. By using synchronous clear operation by inter channel cooperation as described in section 20.3.8.3. Synchronous Clear Operation by Inter Channel Cooperation and selecting compare match as the counter clearing source, it is possible to avoid the counter clear at the end of the trough section.

20.10.8 Setting Range of the Compare Match Registers when GTIOR.CPSCIR = 1

In complementary PWM mode, when GTIOR.CPSCIR = 1 and the initial output of GTIOCnA/GTIOCnB pins after synchronization clearing is disabled, set the compare match registers (GTCCRA, GTCCRC, GTCCRD, GTCCRE, GTCCRF) to greater than twice the value of GTDVU.

20.10.9 Prohibit Invalid Register Setting

The register settings that are determined to be invalid, such as "The setting is invalid during event counting operation", are not guaranteed. These settings are prohibited.

21. Low Power Asynchronous General Purpose Timer (AGTW)

21.1 Overview

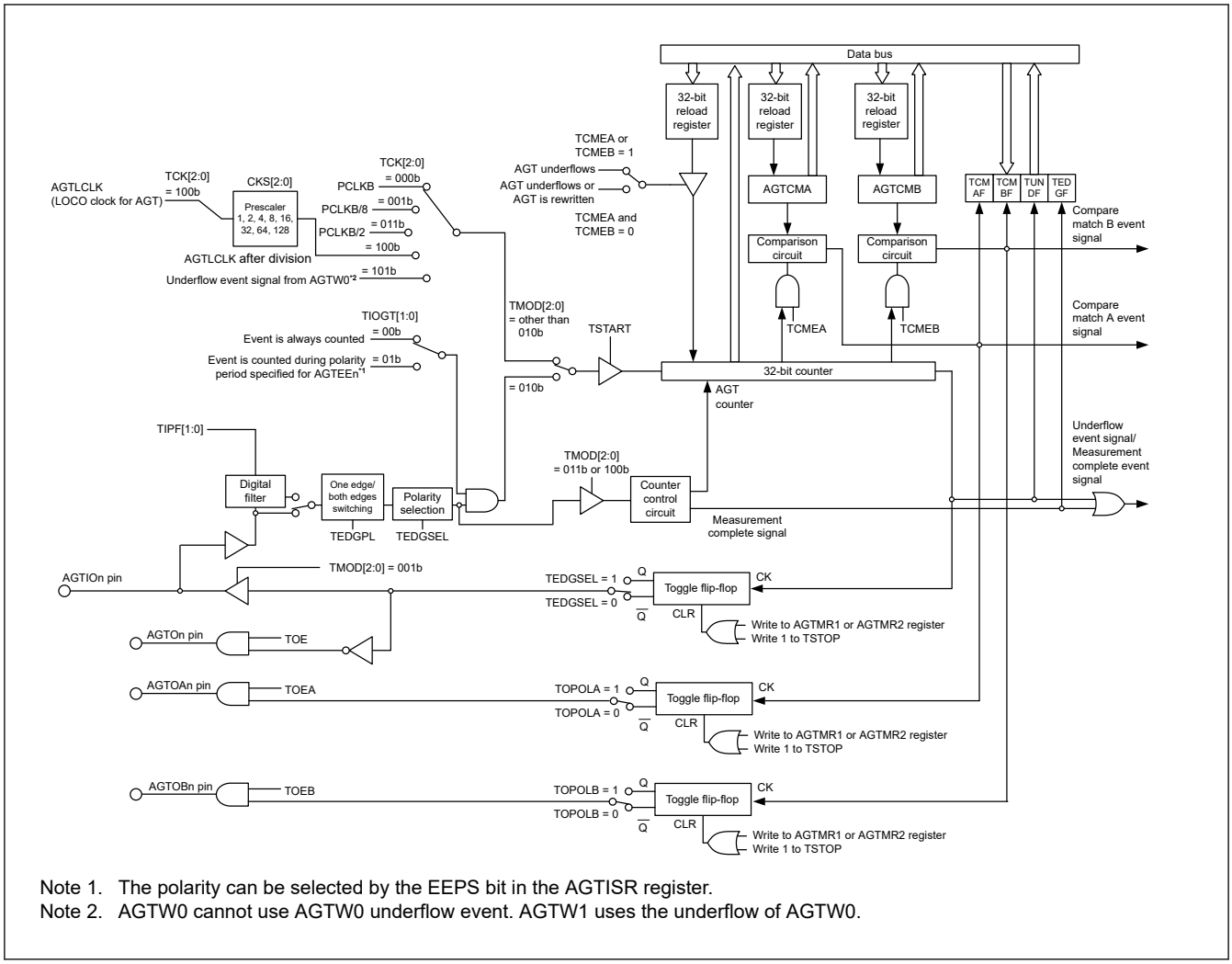
The Low Power Asynchronous General Purpose Timer (AGTW) is a 32-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events. This timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and can be accessed with the AGT register.

Table 21.1 lists the AGTW specifications, Figure 21.1 shows a block diagram, and Table 21.2 lists the I/O pins.

Table 21.1 AGTW specifications

Parameter		Description
Operating modes	Timer mode	The count source is counted
	Pulse output mode	The count source is counted and the output is inverted at each timer underflow
	Event counter mode	An external event is counted
	Pulse width measurement mode	An external pulse width is measured
	Pulse period measurement mode	An external pulse period is measured
Number of Channels		32 bits × 2 channels (AGTWn (n = 0, 1))
Count source (operating clock)*2	Timer mode	PCLKB, PCLKB/2, PCLKB/8, AGTLCLK/d (d = 1, 2, 4, 8, 16, 32, 64, or 128), or underflow signal of AGTW0 selectable.*1
	Pulse output mode	
	Pulse width measurement mode	
	Pulse period measurement mode	
	Event counting mode	External event input
Interrupt and Event Link function		• Underflow event signal or measurement complete event signal – When the counter underflows – When the measurement of the active width of the external input pin (AGTIOn) completes in pulse width measurement mode – When the set edge of the external input pin (AGTIOn) is input in pulse period measurement mode. • Compare match A event signal – When the values of AGT register and AGTCMA register matched (compare match A function enabled). • Compare match B event signal – When the values of AGT and AGTCMB registers matched (compare match B function enabled). • Return from Software Standby mode can be performed with AGT1_AGTCMAI, AGT0_AGTCMAI *3
Selectable functions		• Compare match function One or two of the AGT Compare Match A register and AGT Compare Match B register is selectable.

Note 1. AGTW0 cannot use underflow signal. AGTW1 connects directly with the underflow event signal from the AGTW0 timer.
Note 2. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source clock.
Note 3. For details, see section 10, Low Power Modes.



Note 1. The polarity can be selected by the EEPS bit in the AGTISR register.
Note 2. AGTW0 cannot use AGTW0 underflow event. AGTW1 uses the underflow of AGTW0.

Figure 21.1 AGTW block diagram

Table 21.2 AGTW I/O pins

Pin name	I/O	Function
AGTEEn	Input	External event input enable for AGTW
AGTIOn	Input/output	External event input and pulse output for AGTW
AGTOn	Output	Pulse output for AGTW
AGTOAn	Output	Compare match A output for AGTW
AGTOBn	Output	Compare match B output for AGTW

Note: Channel number: n = 0, 1

21.2 Register Descriptions

21.2.1 AGT : AGT Counter Register

Base address: AGTWn = 0x4008_4000 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	32-bit counter and reload register Setting range : 0x00000000 to 0xFFFFFFFF	R/W

AGTWn.AGT is a 32-bit register. The write value is written to the reload register and the read value is read from the counter.

The states of the reload register and the counter change according to the TSTART bit in the AGTCR register and TCMEA/TCMEB bit in the AGTCMSR register. For details, see section 21.3.1. Reload Register and Counter Rewrite Operation.

When 1 is written to the TSTOP bit in the AGTCR register, AGT counter is forcibly stopped and set to 0xFFFFFFFF.

When the TCK[2:0] bits setting in the AGTMR1 register are a value other than 001b (PCLKB/8) or 011b (PCLKB/2), if the AGT register is set to 0x00000000, a request signal to the ICU, the DTC, and the ELC is generated once immediately after the count starts. The and AGTOn, AGTIOn pin output are toggled.

When the AGT register is set to 0x00000000 in event counter mode, regardless of the value of TCK[2:0] bits, a request signal to the ICU, the DTC, and the ELC is generated once immediately after the count starts.

In addition, the AGTOn pin output is toggled even during a period other than the specified count period. When the AGT register is set to 0x00000001 or more, a request signal is generated each time AGT underflows.

21.2.2 AGTCMA : AGT Compare Match A Register

Base address: AGTWn = 0x4008_4000 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	32-bit compare match A data is stored.*1 Setting range : 0x00000000 to 0xFFFFFFFF	R/W

Note 1. Set the AGTCMA register to 0xFFFFFFFF when compare match A is not used.

The AGTCMA register is a read/write register to set a value for compare match with the AGT counter. The states of the reload register and compare register A change according to the TSTART bit in the AGTCR register. For details, see section 21.3.2. Reload Register and AGT Compare Match A/B Register Rewrite Operation.

21.2.3 AGTCMB : AGT Compare Match B Register

Base address: AGTWn = 0x4008_4000 + 0x0100 × n (n = 0, 1)

Offset address: 0x08

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	32-bit compare match B data is stored.*1 Setting range : 0x00000000 to 0xFFFFFFFF	R/W

Note 1. Set the AGTCMB register to 0xFFFFFFFF when compare match B is not used.

The AGTCMB register is a read/write register to set a value for compare match with the AGT counter. The states of the reload register and compare register B change according to the TSTART bit in the AGTCR register. For details, see section 21.3.2. Reload Register and AGT Compare Match A/B Register Rewrite Operation.

21.2.4 AGTCR : AGT Control Register

Base address: AGTWn = 0x4008_4000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TCMBF	TCMAF	TUNDF	TEDGF	—	TSTOP	TCSTF	TSTART
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSTART	AGT Count Start*2 0: Count stops 1: Count starts	R/W
1	TCSTF	AGT Count Status Flag*2 0: Count stopped 1: Count in progress	R
2	TSTOP	AGT Count Forced Stop*1 0: Writing is invalid 1: The count is forcibly stopped	W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TEDGF	Active Edge Judgment Flag 0: No active edge received 1: Active edge received	R/(W)*3
5	TUNDF	Underflow Flag 0: No underflow 1: Underflow	R/(W)*3
6	TCMAF	Compare Match A Flag 0: No match 1: Match	R/(W)*3
7	TCMBF	Compare Match B Flag 0: No match 1: Match	R/(W)*3

Note 1. When 1 (count is forcibly stopped) is written to the TSTOP bit, the TSTART bit and TCSTF flag are initialized at the same time. The pulse output level is also initialized. The read value is 0.
Note 2. For information on using the TSTART bit and TCSTF flag, see section 21.4.1. Count Operation Start and Stop Control.
Note 3. Only 0 can be written to clear the flag.

TSTART bit (AGT Count Start)

The count operation is started by writing 1 to the TSTART bit and stopped by writing 0. When the TSTART bit is set to 1 (count starts), the TCSTF flag is set to 1 (count in progress) in synchronization with the count source. Also, after 0 is written to the TSTART bit, the TCSTF flag is set to 0 (count stops) in synchronization with the count source. For details, see section 21.4.1. Count Operation Start and Stop Control.

TCSTF flag (AGT Count Status Flag)

The TCSTF flag indicates the AGT count status.

[Setting condition]

- When 1 is written to the TSTART bit (the TCSTF flag is set to 1 in synchronization with the count source).

[Clearing conditions]

- When 0 is written to the TSTART bit (the TCSTF flag is set to 0 in synchronization with the count source)
- When 1 is written to the TSTOP bit.

TSTOP bit (AGT Count Forced Stop)

When 1 is written to the TSTOP bit, the count is forcibly stopped. The read value is 0.

TEDGF flag (Active Edge Judgment Flag)

The TEDGF flag indicates that an active edge was detected.

[Setting condition]

- When the measurement of the active width of the external input pin (AGTIOn) is complete in pulse width measurement mode
- When the set edge of the external input pin (AGTIOn) is input in pulse period measurement mode.

[Clearing condition]

- When 0 is written to this flag by software.

TUNDF flag (Underflow Flag)

The TUNDF flag indicates that the counter underflowed.

[Setting condition]

- When the counter underflows.

[Clearing condition]

- When 0 is written to this flag by software.

TCMAF flag (Compare Match A Flag)

The TCMAF flag indicates that compare match A was detected.

[Setting condition]

- When the value in the AGT register matches the value in the AGTCMA register.

[Clearing condition]

- When 0 is written to this flag by software.

TCMBF flag (Compare Match B Flag)

The TCMBF flag indicates that compare match B was detected.

[Setting condition]

- When the value in the AGT register matches the value in the AGTCMB register.

[Clearing condition]

- When 0 is written to this flag by software.

21.2.5 AGTMR1 : AGT Mode Register 1

Base address: AGTWn = 0x4008_4000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TCK[2:0]			TEDGPL	TMOD[2:0]		
Value after reset:	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TMOD[2:0]	Operating Mode*3 0 0 0: Timer mode 0 0 1: Pulse output mode 0 1 0: Event counter mode 0 1 1: Pulse width measurement mode 1 0 0: Pulse period measurement mode Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
3	TEDGPL	Edge Polarity*4 0: Single-edge 1: Both-edge	R/W
6:4	TCK[2:0]	Count Source*1 *2 *5 *7 0 0 0: PCLKB 0 0 1: PCLKB/8 0 1 1: PCLKB/2 1 0 0: Divided clock AGTLCLK specified by CKS[2:0] bits in the AGTMR2 register 1 0 1: Underflow event signal from AGTW0*6 1 1 0: Setting prohibited Others: Setting prohibited	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: Write access to the AGTMR1 register initializes the output from the AGTOn, AGTIOn, AGTOAn, and AGTOBn pins. For details on the output level at initialization, see section 21.2.7. AGTIOC : AGT I/O Control Register.
Note 1. When event counter mode is selected, the external input pin (AGTIOn) is selected as the count source regardless of the setting of TCK[2:0] bits.
Note 2. Do not switch count sources during count operation. Only switch count sources when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).
Note 3. The operating mode can only be changed when the count is stopped while both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count is stopped). Do not change the operating mode during count operation.
Note 4. The TEDGPL bit is enabled only in event counter mode.
Note 5. To run AGT in Software Standby mode, Snooze mode, select AGTLCLK (TCK[2:0] = 100b).
Note 6. AGTW0 cannot use AGTW0 underflow (setting prohibited). AGTW1 uses the AGTW0 underflow.
Note 7. Do not change the TCK[2:0] bits when the CKS[2:0] bits in the AGTMR2 register is not 000b. First, change the CKS[2:0] bits in the AGTMR2 register to 000b. Then change the TCK[2:0] bits and wait for one cycle of the count source.

21.2.6 AGTMR2 : AGT Mode Register 2

Base address: AGTWn = 0x4008_4000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0E

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

Bit	Symbol	Function	R/W
2:0	CKS[2:0]	AGTLCLK Count Source Clock Frequency Division Ratio*1 *2 *3 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	LPM	Low Power Mode 0: Normal mode 1: Low power mode	R/W

Note 1. Do not rewrite the CKS[2:0] bits during count operation. Only rewrite the CKS[2:0] bits when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).
Note 2. When count source is AGTLCLK, the switch of CKS[2:0] bits is valid.
Note 3. Do not switch the TCK[2:0] bits in the AGTMR1 register when CKS[2:0] bits are not 000b. Switch the TCK[2:0] bits in the AGTMR1 register after CKS[2:0] bits are set to 000b, and wait for 1 cycle of the count source.

CKS[2:0] bit (AGTLCLK Count Source Clock Frequency Division Ratio)

CKS[2:0] bits select the Count Source Clock Frequency Division Ratio for AGTLCLK.

LPM bit (Low Power Mode)

The LPM bit sets the low power operation, which impacts access to certain AGT registers. Set this bit to 1 to operate in low power.

When this bit is 1, access to the following registers is prohibited:

- AGT/AGTCMA/AGTCMB/AGTCR.

After this bit is switched from 1 to 0, the first access to the register is constrained as follows:

- When reading from the AGT register, read AGT register twice. Only the second reading of data is valid.
- When writing to the AGT, AGTCMA, AGTCMB, and AGTCR register, allow at least 2 cycles of the count source clock when writing to the register.
- When confirm the value written to the AGT, AGTCMA, AGTCMB, and AGTCR registers.
  - When the count operation is stopped; after writing data, it can be read in the next cycle.
  - When the count operation is operating; after writing data, it can be read 4 cycles after the count source clock.

Figure 21.2 shows the flow of how to write LPM bit.

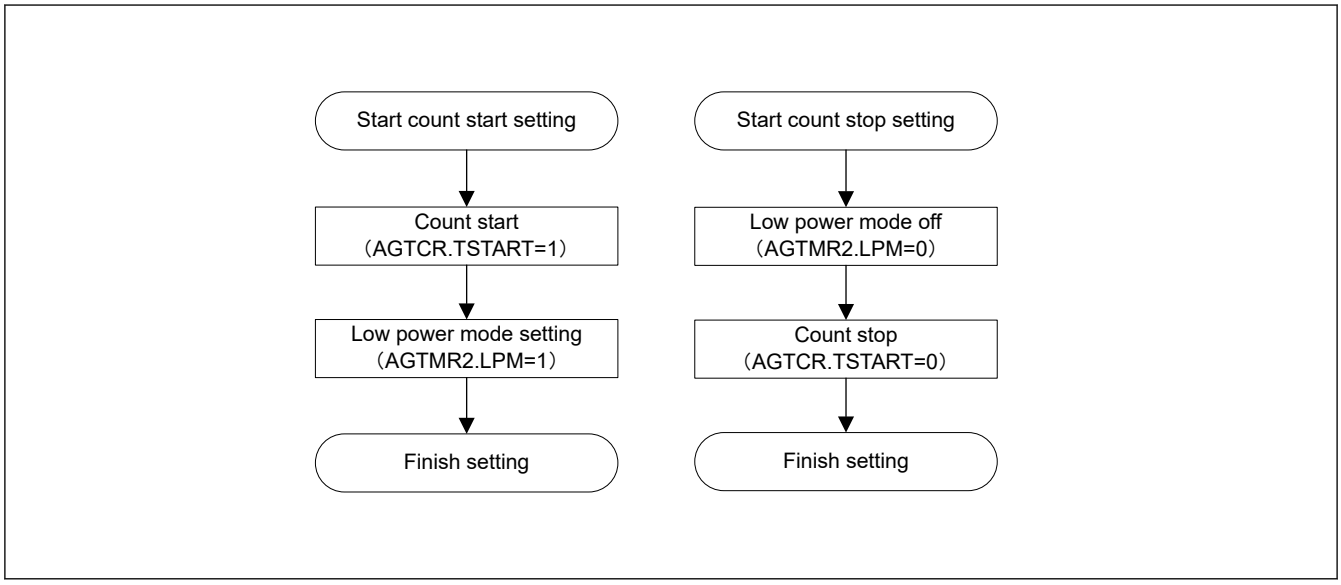


Figure 21.2 LPM how to write flow chart

21.2.7 AGTIOC : AGT I/O Control Register

Base address: AGTWn = 0x4008_4000 + 0x0100 × n (n = 0, 1)

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIOGT[1:0]		TIPF[1:0]		—	TOE	—	TEDG SEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TEDGSEL	I/O Polarity Switch Function varies depending on the operating mode (see Table 21.3 and Table 21.4).	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	TOE	AGTOn pin Output Enable 0: AGTOn pin output disabled 1: AGTOn pin output enabled	R/W

Bit	Symbol	Function	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	TIPF[1:0]	Input Filter*3 These bits specifies the sampling frequency of the filter for the AGTIOn input. If the input to the AGTIOn pin is sampled and the value matches three successive times, that value is taken as the input value. 0 0: No filter 0 1: Filter sampled at PCLKB 1 0: Filter sampled at PCLKB/8 1 1: Filter sampled at PCLKB/32	R/W
7:6	TIOGT[1:0]	Count Control*1 *2 0 0: Event is always counted 0 1: Event is counted during polarity period specified for AGTEEn pin Others: Setting prohibited	R/W

Note 1. When AGTEEn pin is used, the polarity to count an event can be selected with the EEPS bit in the AGTISR register.
Note 2. TIOGT[1:0] bits are enabled only in event counter mode.
Note 3. When event counter mode operation is performed during Software Standby mode, the digital filter function cannot be used.

TEDGSEL bit (I/O Polarity Switch)

The TEDGSEL bit switches the AGTOn pin output polarity and the AGTIOn pin input/output edge and polarity.

In pulse output mode, it only controls polarity of the AGTOn pin output and AGTIOn pin output. AGTOn pin output and AGTIOn pin output are initialized when the AGTMR1 register is written or the TSTOP bit in the AGTCR register is written with 1.

TOE bit (AGTOn pin Output Enable)

The TOE bit selects whether the AGTOn pin output is disabled or enabled.

TIPF[1:0] bits (Input Filter)

The TIPF[1:0] bits specify the sampling frequency of the AGTIOn pin input filter. When the input to the AGTIOn pin is sampled and the values match three times in succession, the value is regarded as the input value.

TIOGT[1:0] bits (Count Control)

The TIOGT[1:0] bits control the event count.

Table 21.3 AGTIOn pin I/O edge and polarity switching

Operating mode	Function
Timer mode	Not used
Pulse output mode	0: Output is started at high (initialization level: high) i.e. inverted output 1: Output is started at low (initialization level: low). i.e. normal output
Event counter mode	0: Count on rising edge 1: Count on falling edge.
Pulse width measurement mode	0: Low-level width is measured 1: High-level width is measured.
Pulse period measurement mode	0: Measure from one rising edge to the next rising edge 1: Measure from one falling edge to the next falling edge.

Table 21.4 AGTOn pin output polarity switching

Operating mode	Function
All modes	0: Output is started at low (initial level: low): Normal output 1: Output is started at high (initial level: high): Inverted output

21.2.8 AGTISR : AGT Event Pin Select Register

Base address: AGTWn = 0x4008_4000 + 0x0100 × n (n = 0, 1)

Offset address: 0x11

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

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	EEPS	AGTEEn Polarity Selection 0: An event is counted during the low-level period 1: An event is counted during the high-level period	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

EEPS bit (AGTEEn Polarity Selection)

The EEPS bit selects the polarity of events to be counted.

21.2.9 AGTCMSR : AGT Compare Match Function Select Register

Base address: AGTWn = 0x4008_4000 + 0x0100 × n (n = 0, 1)

Offset address: 0x12

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TOPOLB	TOEB	TCMEB	—	TOPOLA	TOEA	TCMEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCMEA	AGT Compare Match A Register Enable*1 *2 0: AGT Compare match A register disabled 1: AGT Compare match A register enabled	R/W
1	TOEA	AGTOAn Pin Output Enable*1 *2 0: AGTOAn pin output disabled 1: AGTOAn pin output enabled	R/W
2	TOPOLA	AGTOAn Pin Polarity Select*1 *2 0: AGTOAn pin output is started on low. i.e. normal output 1: AGTOAn pin output is started on high. i.e. inverted output	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TCMEB	AGT Compare Match B Register Enable*1 *2 0: Compare match B register disabled 1: Compare match B register enabled	R/W
5	TOEB	AGTOBn Pin Output Enable*1 *2 0: AGTOBn pin output disabled 1: AGTOBn pin output enabled	R/W
6	TOPOLB	AGTOBn Pin Polarity Select*1 *2 0: AGTOBn pin output is started on low. i.e. normal output 1: AGTOBn pin output is started on high. i.e. inverted output	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. Do not rewrite the AGTCMSR register during a count operation. Only rewrite the AGTCMSR register when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).

Note 2. Do not set 1 when in pulse width measurement mode or pulse period measurement mode.

21.2.10 AGTIOSEL : AGT Pin Select Register

Base address: AGTWn = 0x4008_4000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0F

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

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	TIES	AGTIOn Pin Input Enable 0: External event input is disabled during Software Standby mode 1: External event input is enabled during Software Standby mode	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The AGTIOSEL register sets the AGTIOn pin when using the AGTIOn pin in Software Standby mode.

TIES bit (AGTIOn Pin Input Enable)

The TIES bit enables or disables an external event input.

21.3 Operation

21.3.1 Reload Register and Counter Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and the counter differs depending on the value of the TSTART bit in the AGTCR register and of the TCMEA or TCMEB bit in the AGTCMSR register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and the counter. When the TSTART bit is 1 (count starts) and the TCMEA bit and TCMEB bit are 0 (AGT compare match A/B register are invalid), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the next count source. When the TSTART bit is 1 (count starts) and the TCMEA bit or the TCMEB bit is 1 (AGT compare match A register or compare match B register is valid), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the underflow of the counter.

Figure 21.3 and Figure 21.4 show the timing of rewrite operation with TSTART bit value and TCMEA/TCMEB bit value.

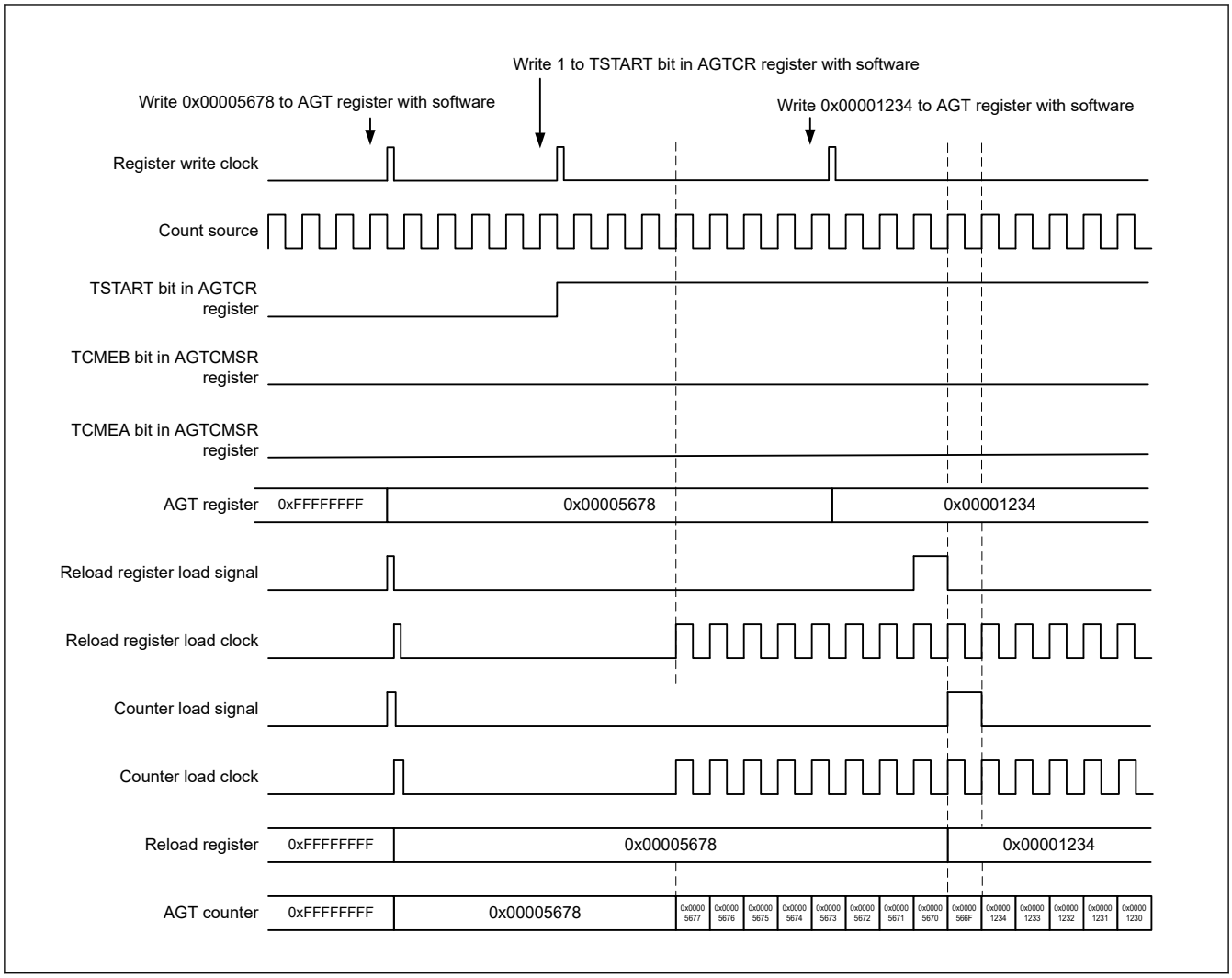


Figure 21.3 Timing of rewrite operation with TSTART, TCMEA, and TCMEB bit value when AGT compare match A register and AGT compare match B register is invalid

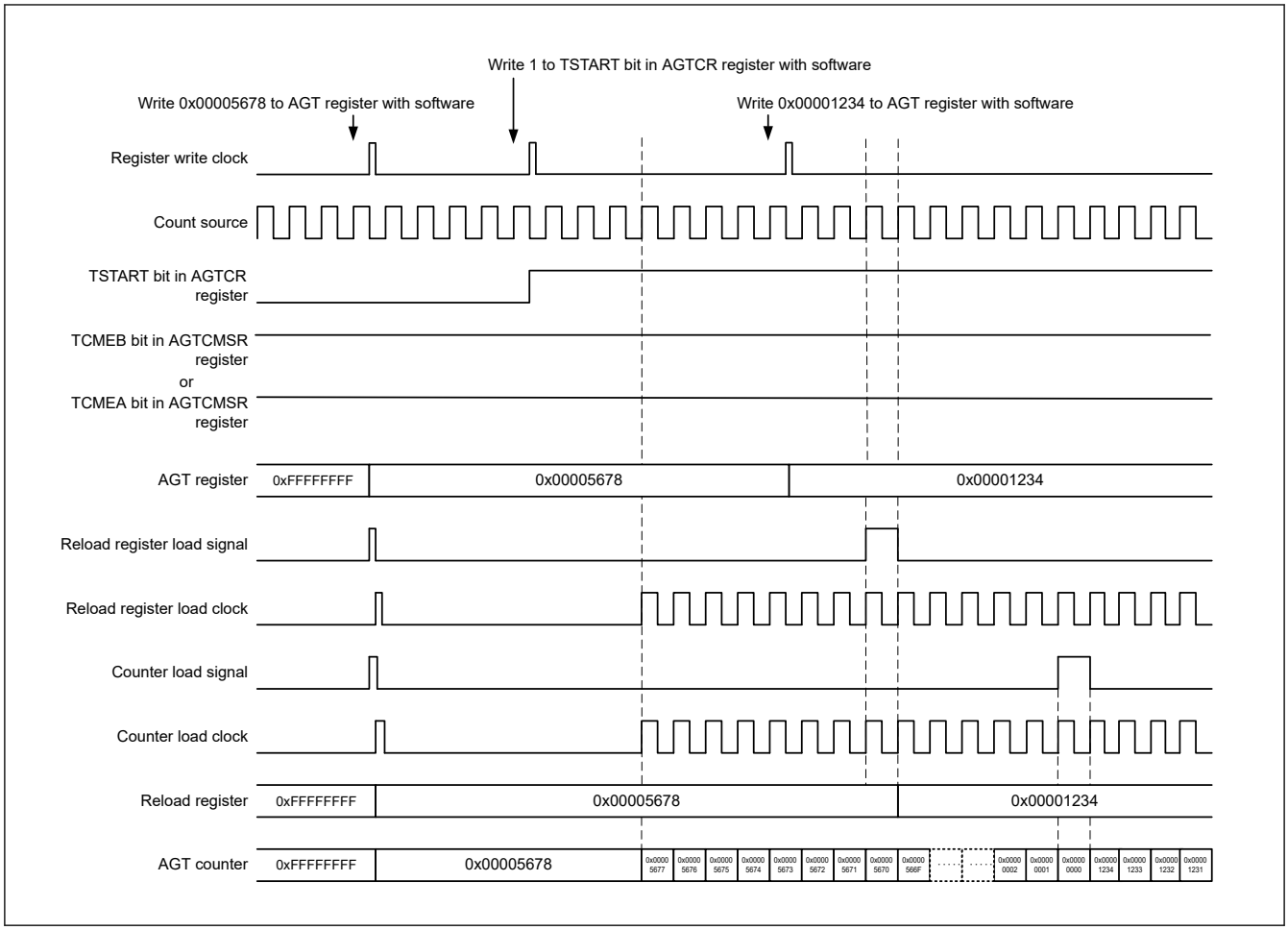


Figure 21.4 Timing of rewrite operation with TSTART bit value and TCMEA or TCMEB bit value when AGT compare match A register or AGT compare match B register is valid

21.3.2 Reload Register and AGT Compare Match A/B Register Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and AGT compare register A/B depends on the value of the TSTART bit in the AGTCR register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and AGT compare register A/B. When the TSTART bit is 1 (count starts), the value is written to the reload register in synchronization with the count source, and then to the compare register in synchronization with the underflow of the counter.

Figure 21.5 shows the timing of rewrite operation with TSTART bit value for compare register A. AGT Compare register B is of the same timing as AGT compare register A.

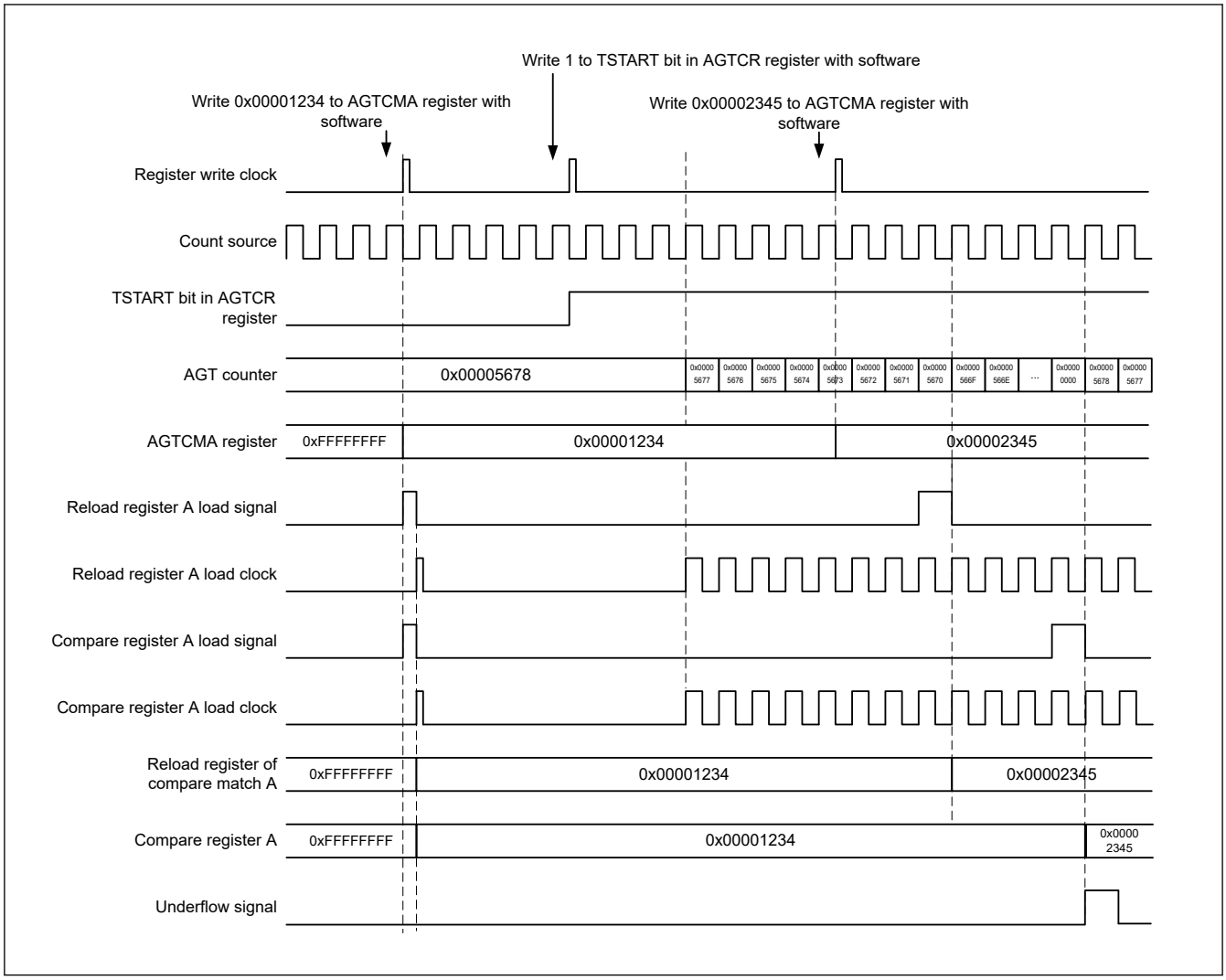


Figure 21.5 Timing of rewrite operation with the TSTART bit value for AGT compare register A

21.3.3 Timer Mode

In this mode, the AGT counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register. In timer mode, the count value is decremented by 1 on each rising edge of the count source. When the count value reaches 0x00000000 and the next count source is input, an underflow occurs and an interrupt request is generated.

Figure 21.6 shows the operation example in timer mode.

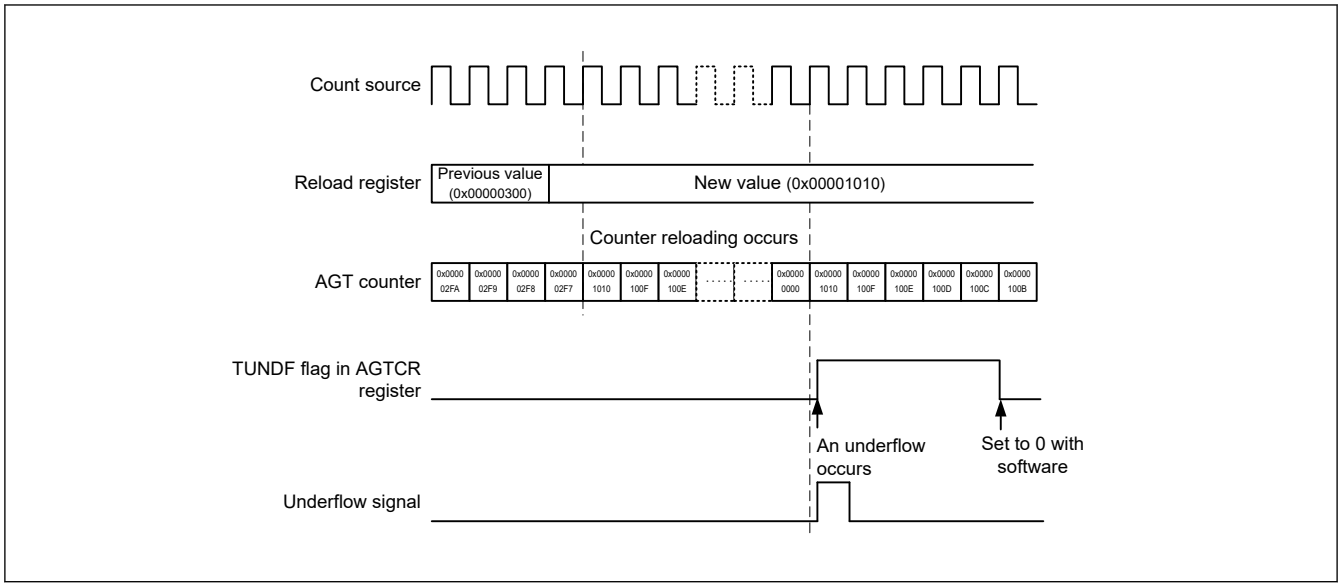


Figure 21.6 Operation example in timer mode

21.3.4 Pulse Output Mode

In pulse output mode, the counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register, and the output level of the AGTIOn and AGTOn pins inverted each time an underflow occurs.

In pulse output mode, the count value is decremented by 1 on each rising edge of the count source. When the count value reaches 0x00000000 and the next count source is input, an underflow occurs and an interrupt request is generated. In addition, a pulse can be output from the AGTIOn and AGTOn pins. The output level is inverted each time an underflow occurs. The pulse output from the AGTOn pin can be stopped with the TOE bit in the AGTIOC register. The output level can be selected with the TEDGSEL bit in the AGTIOC register.

Figure 21.7 shows the operation example in pulse output mode.

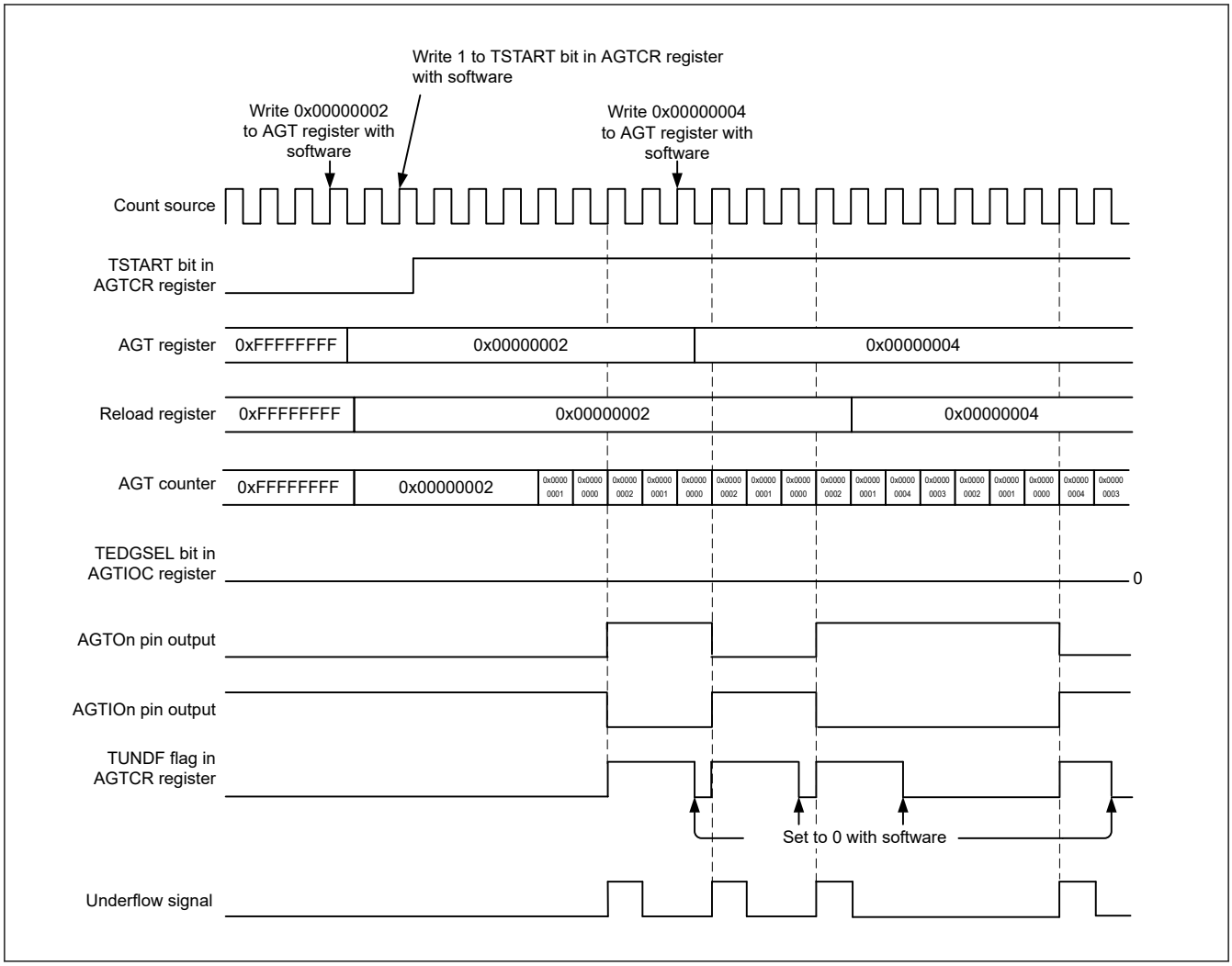


Figure 21.7 Operation example in pulse output mode

21.3.5 Event Counter Mode

In event counter mode, the counter is decremented by an external event signal (count source) input to the AGTIOn pin. Various periods for counting events can be set with the TIOGT[1:0] bits in the AGTIOC register and AGTISR registers. In addition, the filter function for the AGTIOn pin input can be specified with bits TIPF[1:0] in the AGTIOC register. The output from the AGTOn pin can be toggled even in event counter mode.

Figure 21.8 shows the operation example in event counter mode.

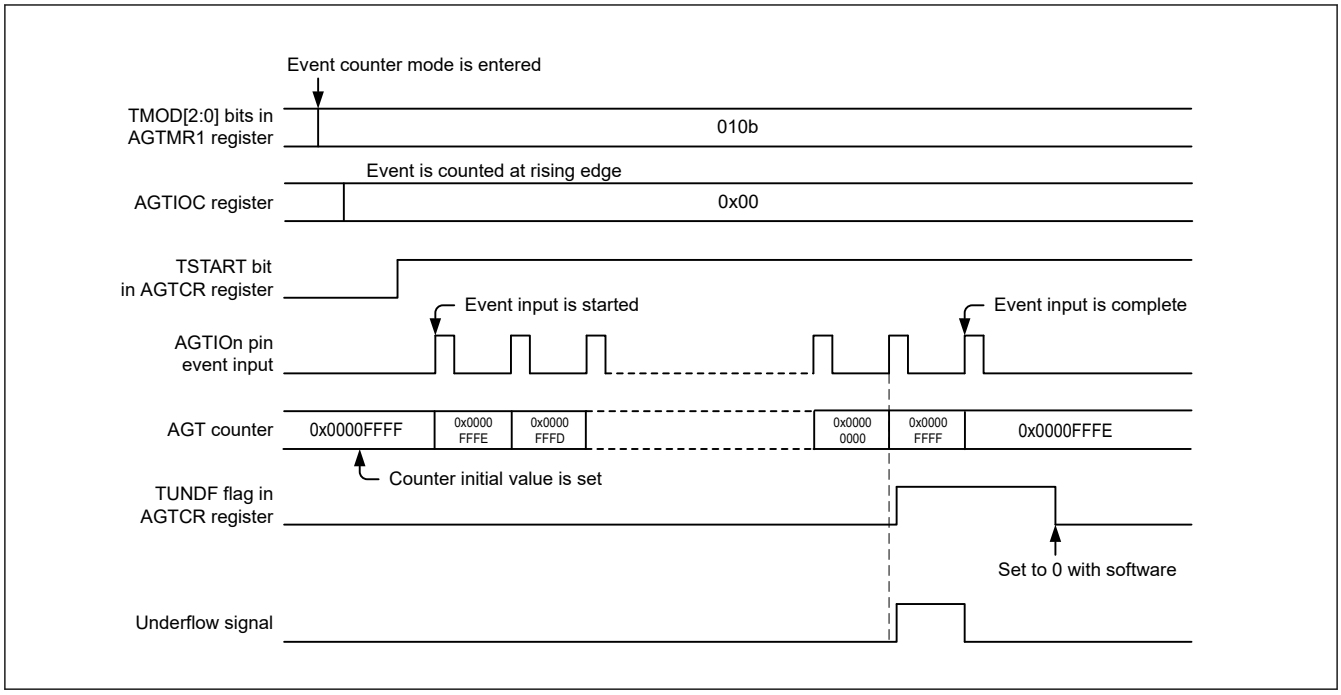
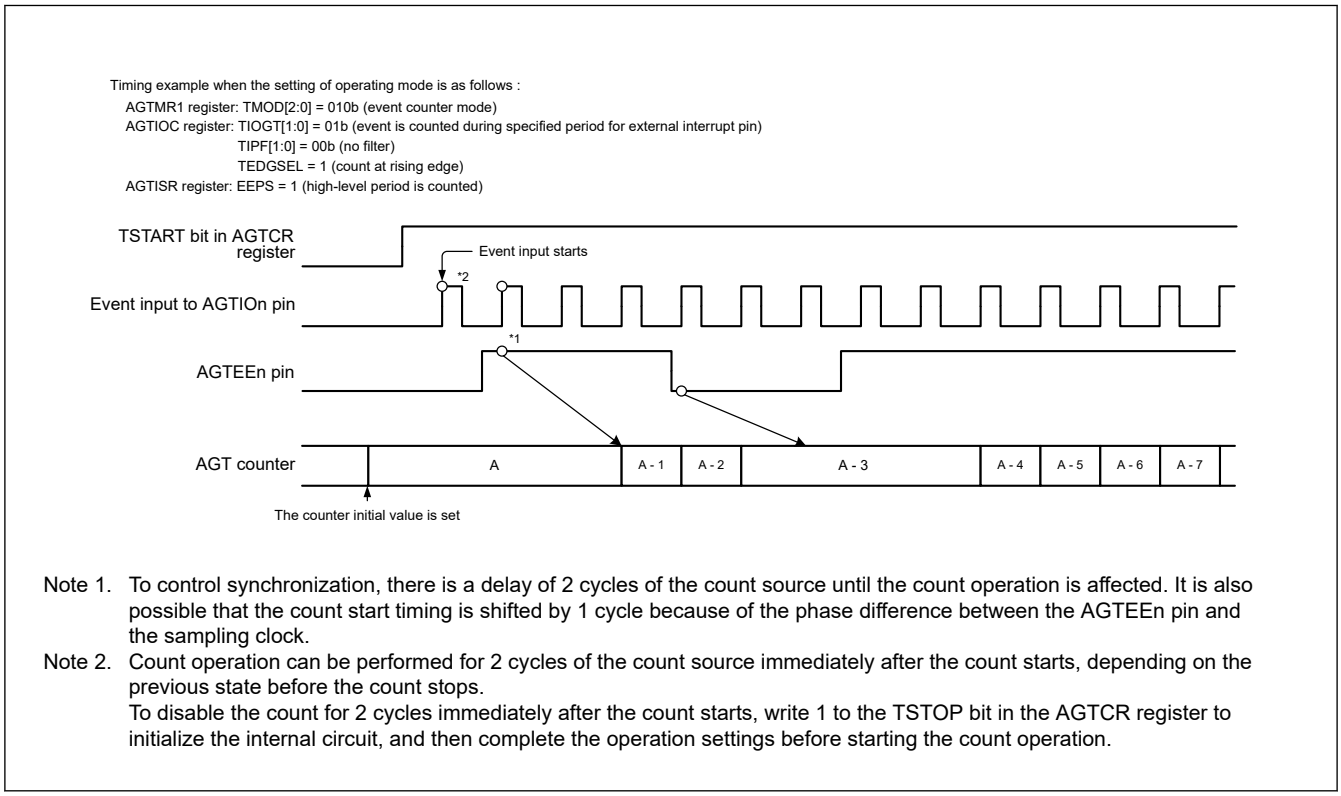


Figure 21.8 Operation example 1 in event counter mode

Figure 21.9 shows an operation example for counting during the specified period in event counter mode (TIOGT[1:0] bits in the AGTIOC register are set to 01b).



Note 1. To control synchronization, there is a delay of 2 cycles of the count source until the count operation is affected. It is also possible that the count start timing is shifted by 1 cycle because of the phase difference between the AGTEEn pin and the sampling clock.

Note 2. Count operation can be performed for 2 cycles of the count source immediately after the count starts, depending on the previous state before the count stops.
To disable the count for 2 cycles immediately after the count starts, write 1 to the TSTOP bit in the AGTCR register to initialize the internal circuit, and then complete the operation settings before starting the count operation.

Figure 21.9 Operation example 2 in event counter mode

21.3.6 Pulse Width Measurement Mode

In pulse width measurement mode, the pulse width of an external signal input to the AGTIOn pin is measured. When the level specified by the TEDGSEL bit in the AGTIOC register is input to the AGTIOn pin, the counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register. When the specified level on the AGTIOn pin ends,

the counter is stopped, the TEDGF flag in the AGTCR register is set to 1 (active edge received), and an interrupt request is generated. The measurement of pulse width data is performed by reading the count value while the counter is stopped. Also, when the counter underflows during measurement, the TUNDF flag in the AGTCR register is set to 1 and an interrupt request is generated.

Figure 21.10 shows the operation example in pulse width measurement mode.

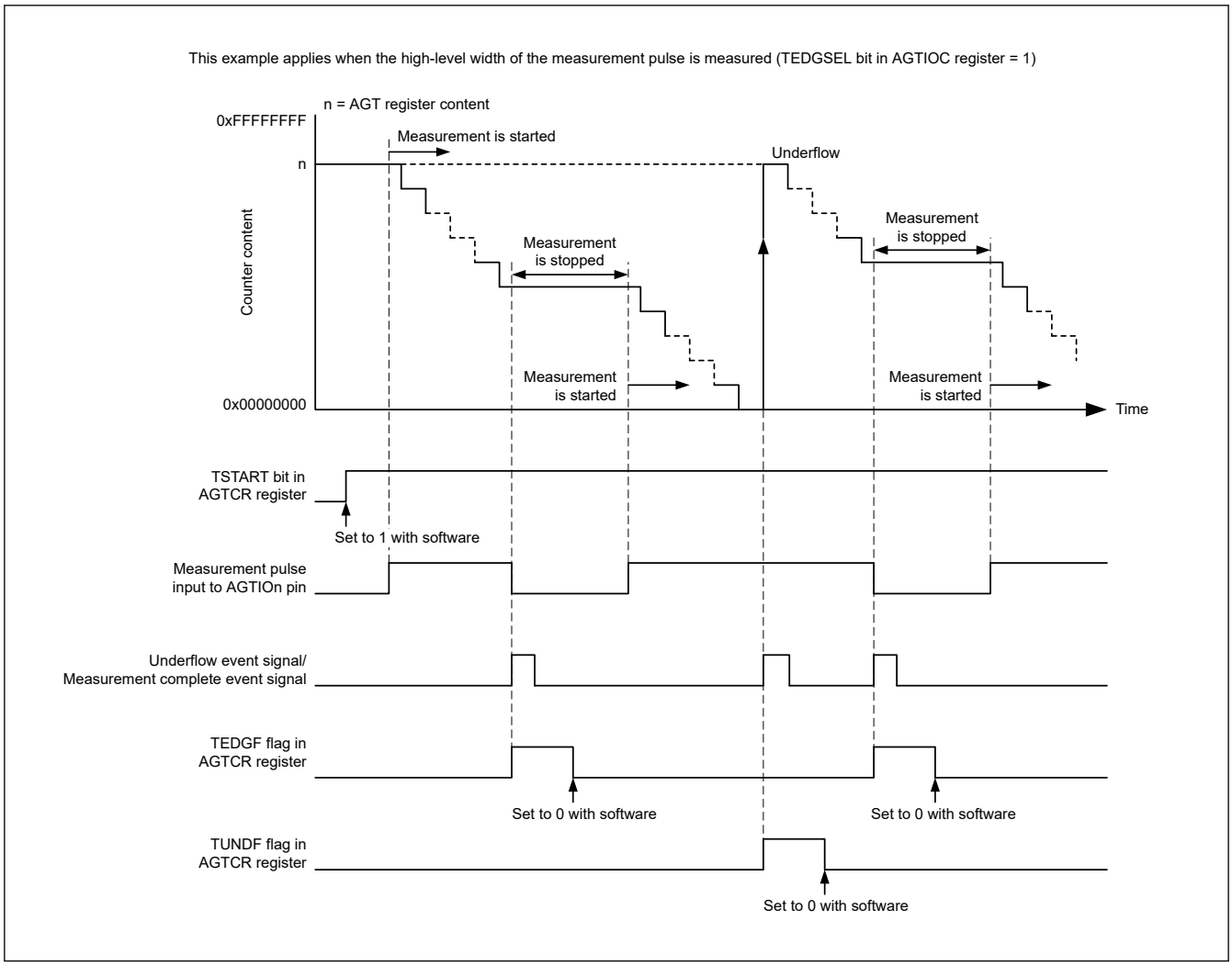


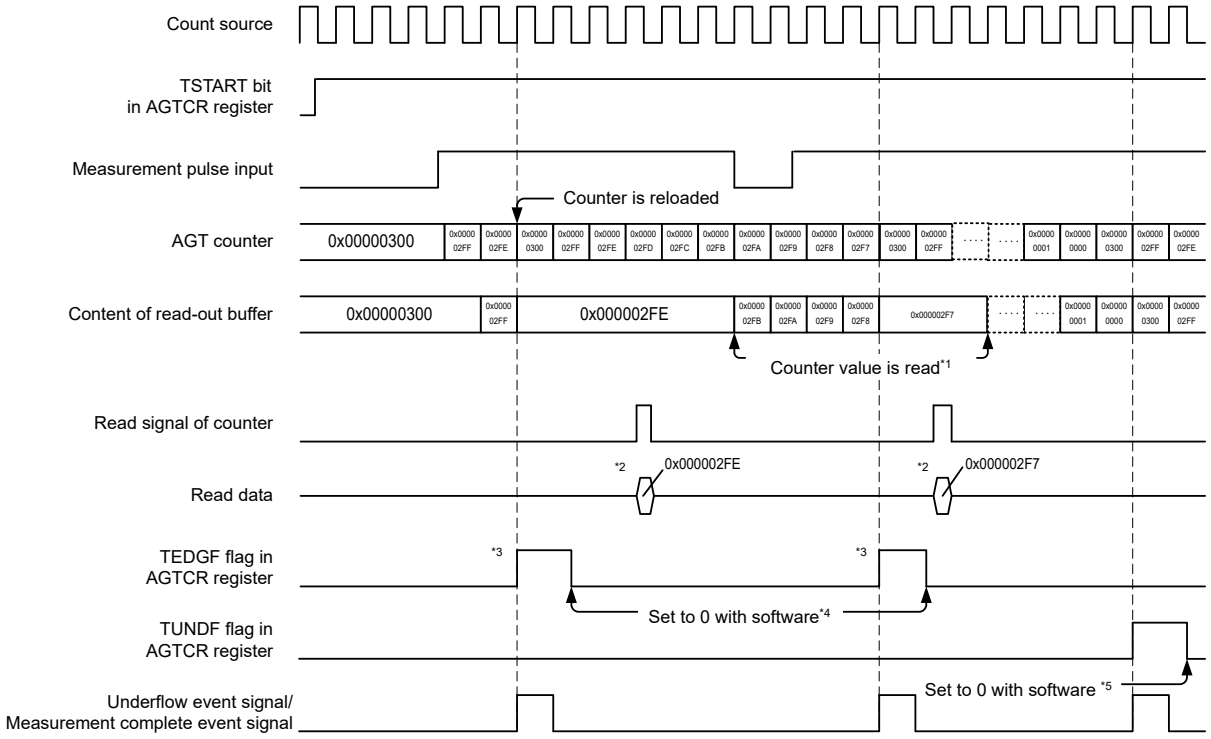
Figure 21.10 Operation example in pulse width measurement mode

21.3.7 Pulse Period Measurement Mode

In pulse period measurement mode, the pulse period of an external signal input to the AGTIOn pin is measured. The counter is decremented by the count source selected with TCK[2:0] bits in the AGTMR1 register. When a pulse with the period specified by the TEDGSEL bit in the AGTIOC register is input to the AGTIOn pin, the count value is transferred to the read-out buffer on the rising edge of the count source. The value in the reload register is loaded to the counter at the next rising edge. Simultaneously, the TEDGF flag in the AGTCR register is set to 1 (active edge received) and an interrupt request is generated. The read-out buffer (AGT register) is read at this time and the difference from the reload value (see section 21.4.6. How to Calculate Event Number, Pulse Width, and Pulse Period) is the period data of the input pulse. The period data is retained until the read-out buffer is read. When the counter underflows, the TUNDF flag in the AGTCR register is set to 1 (underflow) and an interrupt request is generated.

Figure 21.11 shows the operation example in pulse period measurement mode.

Only input pulses with a period longer than twice the period of the count source are measured. Also, the low-level and high-level widths must both be longer than the period of the count source. If a pulse period shorter than these conditions is input, the input might be ignored.



This example applies when the initial value of the AGT register is set to A, the TEDGSEL bit in the AGTIOC register is set to 0, and the period from one rising edge to the next edge of the measurement pulse is measured.

Note 1. Reading from the AGT register must be performed during the period from when the TEDGF flag is set to 1 (active edge received) until the next active edge is input. The content of the read-out buffer is retained until the AGT register is read. If it is not read before the active edge is input, the measurement result of the previous period is retained.
Note 2. When the AGT register is read in pulse period measurement mode, the content of the read-out buffer is read.
Note 3. When the active edge of the measurement pulse is input and then the set edge of an external pulse is input, the TEDGF flag in the AGTCR register is set to 1 (active edge received).
Note 4. To set to 0 with software, write 0 to the TEDGF flag in the AGTCR register with an 8-bit memory manipulation instruction.
Note 5. To set to 0 with software, write 0 to the TUNDF flag in the AGTCR register with an 8-bit memory manipulation instruction.

Figure 21.11 Operation example in pulse period measurement mode

21.3.8 Compare Match function

The compare match function detects matches (compare match) between the content of the AGTCMA or AGTCMB register and the content of the AGT register. This function is enabled when the TCMEA or TCMEB bit in the AGTCMSR register is 1 (compare match A register or compare match B register is valid). The counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register, and when the values of AGT and AGTCMA or AGTCMB match, the TCMAF/TCMBF flag in the AGTCR register is set to 1 (match), and an interrupt request is generated.

When the compare match function is enabled, the timing of the rewrite operation to the reload register and the counter differs. See section 21.3.1. Reload Register and Counter Rewrite Operation for details. In addition, the output level of the AGTOAn and AGTOBn pins is inverted by the match and by the underflow. The output level can be selected with the TOPOLA or TOPOLB bit in the AGTCMSR register.

Figure 21.12 shows the operation example in compare match function.

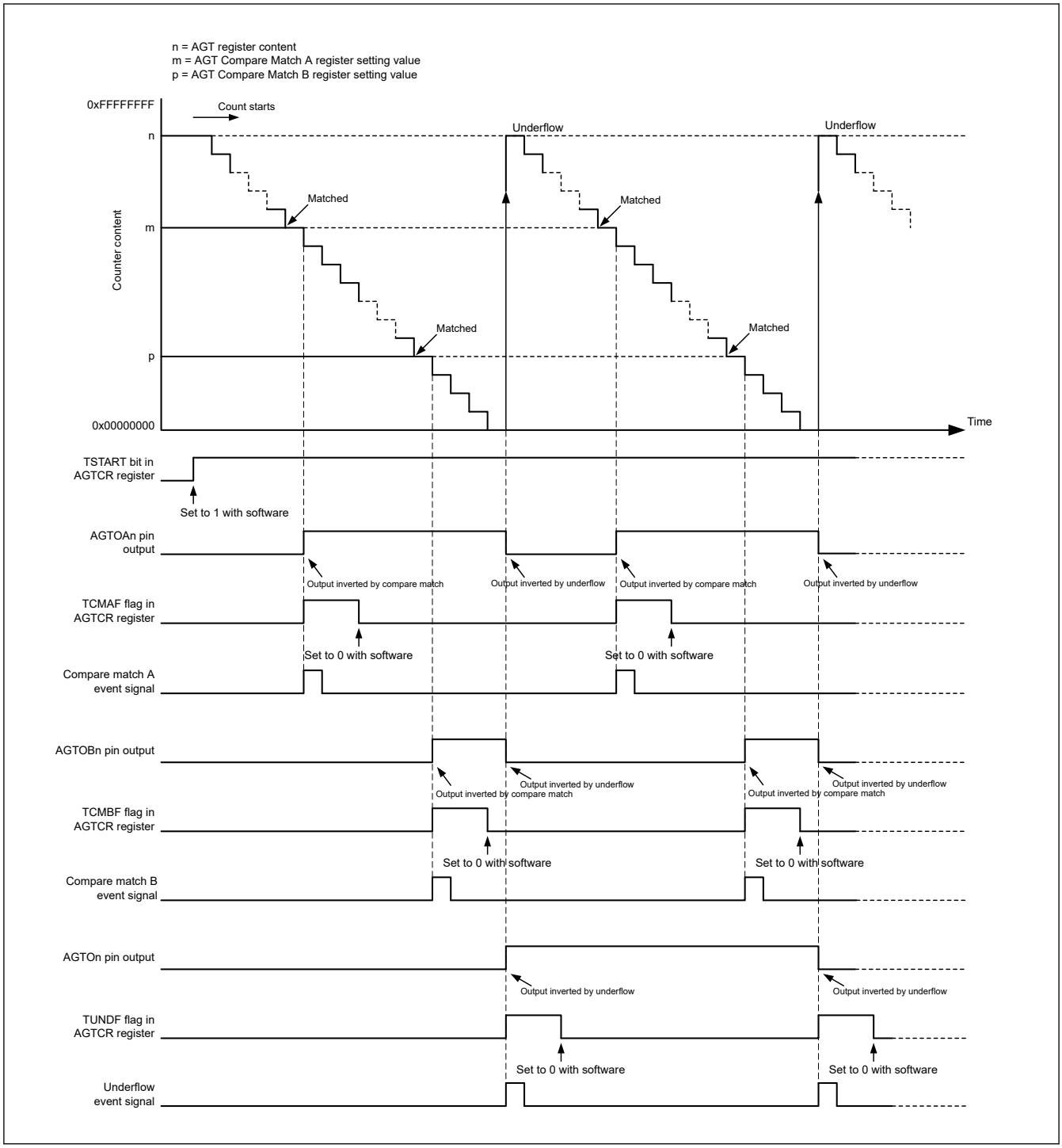


Figure 21.12 Operation example in compare match function (TOPOLA = 0, TOPOLB = 0)

21.3.9 Output Settings for Each Mode

Table 21.5 to Table 21.8 list the states of pins AGTOn, AGTIOn, AGTOAn, and AGTOBn pins in each mode.

Table 21.5 AGTOn pin setting

Operating mode	AGTIOC register		AGTOn pin output
	TOE bit	TEDGSEL bit	
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

Table 21.6 AGTIOn pin setting

Operating mode	AGTIOC register	AGTIOn pin I/O
	TEDGSEL bit	
Timer mode	0 or 1	Input (not used)
Pulse output mode	1	Normal output
	0	Inverted output
Event counter mode	0 or 1	Input
Pulse width measurement mode		
Pulse period measurement mode		

Table 21.7 AGTOAn pin setting

Operating mode	AGTCMSR register		AGTOAn pin output
	TOEA bit	TOPOLA bit	
Timer mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse output mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Event counter mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse width measurement mode	0	0	Prohibited
Pulse period measurement mode			

Table 21.8 AGTOBn pin setting (1 of 2)

Operating mode	AGTCMSR register		AGTOBn pin output
	TOEB bit	TOPOLB bit	
Timer mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse output mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Event counter mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)

Table 21.8 AGTOBn pin setting (2 of 2)

Operating mode	AGTCMSR register		AGTOBn pin output
	TOEB bit	TOPOLB bit	
Pulse width measurement mode	0	0	Prohibited
Pulse period measurement mode			

21.3.10 Standby Mode

The AGTW can operate in Software Standby mode. Set it to Software Standby mode with count operation start (TSTART = 1, TCSTF = 1).

Table 21.9 and Table 21.10 show the setting that can be used in Software Standby mode.

Table 21.9 Usable settings in Software Standby mode (AGTW0)

Operating mode	AGTMR1.TCK[2:0]	Operating clock	Resurgence factor of CPU
Timer mode	100b	AGTLCLK	—
Pulse output mode	100b	AGTLCLK	—
Event counter mode*2	—	AGTIOn (n = 0)*1	—
Pulse width measurement mode	100b	AGTLCLK	—
Pulse period measurement mode	100b	AGTLCLK	—

Note: —: invalid
Note 1. When using the AGTIOn pin for external event input in Software Standby mode, set AGTIOSEL.TIES = 1.
Note 2. When external event input is disabled during Software Standby mode (AGTIOSEL.TIES = 0), stop the count operation before entering Software Standby mode. And restart the count operation after returning from Software Standby mode as needed.

Table 21.10 Usable settings in Software Standby mode (AGTW1)

Operating mode	AGTMR1.TCK[2:0]	Operating clock	Resurgence factor of CPU
Timer mode	100b or 101b*1	AGTLCLK or AGTW0 underflow	• Underflow • Compare match A/B
Pulse output mode	100b or 101b*1	AGTLCLK or AGTW0 underflow	• Underflow • Compare match A/B
Event counter mode*3	—	AGTIOn (n = 1)*2	• Underflow • Compare match A/B
Pulse width measurement mode	100b or 101b*1	AGTLCLK or AGTW0 underflow	• Underflow • Active edge
Pulse period measurement mode	100b or 101b*1	AGTLCLK or AGTW0 underflow	• Underflow • Active edge

Note: —: invalid
Note: Release of Software Standby mode is only AGT1.
Note 1. Only when AGTW0 operates in Table 21.9.
Note 2. When using the AGTIOn pin for external event input in Software Standby mode, set AGTIOSEL.TIES = 1.
Note 3. When external event input is disabled during Software Standby mode (AGTIOSEL.TIES = 0), stop the count operation before entering Software Standby mode. And restart the count operation after returning from Software Standby mode as needed.

21.3.11 Interrupt Sources

The AGTWn has three interrupt sources as listed in Table 21.11.

Table 21.11 AGTW interrupt sources (1 of 2)

Name	Interrupt source	DTC activation
AGTn_AGTI	• When the counter underflows • When measurement of the active width of the external input pin (AGTIOn) is complete in pulse width measurement mode • When the set edge of the external input pin (AGTIOn) is input in pulse period measurement mode.	Possible
AGTn_AGTCMAI	• When the values of AGT register and AGTCMA register match	Possible

Table 21.11 AGTW interrupt sources (2 of 2)

Name	Interrupt source	DTC activation
AGTn_AGTCMBI	• When the values of AGT register and AGTCMB register match	Possible

Note: Channel number (n = 0, 1)

21.3.12 Event Signal Output to ELC

The AGTWn (n = 0, 1) uses the Event Link Controller (ELC) to perform a link operation to a specified module using the interrupt request signal as the event signal. The AGTWn (n = 0, 1) outputs compare match A, compare match B, and underflow/measurement complete signals as event signals. For details, see section 16, Event Link Controller (ELC).

21.4 Usage Notes

21.4.1 Count Operation Start and Stop Control

- When the operating mode (see Table 21.1) is set to other than the event counter mode, or the count source is set to other than AGTWn underflow event signal (TCK[2:0] = 101b):
  - After 1 (count starts) is written to the TSTART bit in the AGTCR register while the count is stopped, the TCSTF flag in the AGTCR register remains 0 (count stops) for 3 cycles of the count source. Do not access the registers associated with AGTW*1 other than the TCSTF flag until this flag is set to 1 (count in progress).
  - After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF flag remains 1 for 3 cycles of the count source. When the TCSTF flag is set to 0, the count is stopped. Do not access the registers associated with AGTW*1 other than the TCSTF flag until this flag is set to 0.
  - Clear the interrupt register before changing the TSTART bit from 0 to 1. For details, see section 12, Interrupt Controller Unit (ICU).
- When the operating mode (see Table 21.1) is set to event counter mode, or the count source is set to AGTW1 underflow event signal (TCK[2:0] = 101b):
  - After 1 (count starts) is written to the TSTART bit in the AGTCR register while the count is stopped, the TCSTF flag in the AGTCR register remains 0 (count stops) for 2 PCLKB cycles. Do not access the registers associated with AGTW*1 other than the TCSTF flag until this flag is set to 1 (count in progress).
  - After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF flag remains 1 for 2 PCLKB cycles. When the TCSTF flag is set to 0, the count is stopped. Do not access the registers associated with AGTW*1 other than the TCSTF flag until this flag is set to 0.
  - Clear the interrupt register before changing the TSTART bit from 0 to 1. For details, see section 12, Interrupt Controller Unit (ICU).

Note 1. Registers associated with AGT: AGT, AGTCMA, AGTCMB, AGTCR, AGTMR1, AGTMR2, AGTIOC, AGTISR, and AGTCMSR.

21.4.2 Access to Counter Register

When the TSTART bit and TCSTF flag in the AGTCR register are both 1 (count starts), allow at least 3 cycles of the count source clock between writes when writing to the AGT register successively.

21.4.3 When Changing Mode

The registers associated with AGT operating mode (AGTMR1, AGTMR2, AGTIOC, AGTISR, and AGTCMSR) can be changed only when the count is stopped with both the TSTART bit and TCSTF flag set to 0 (count stops). Do not change these registers during count operation.

When the registers associated with AGT operating mode are changed, the values of TEDGF, TUNDF, TCMAF, and TCMBF flags are undefined. Before starting the count, write 0 to the following flags:

- TEDGF (no active edge received)
- TUNDF (no underflow)

- TCMAF (no match)
- TCMBF (no match).

21.4.4 Output pin setting

When using the AGTOn, AGTIOn, AGTOAn, or AGTOBn as an output pin, set up the Operation and determine the initial output values. Then set an output mode in the port register.

When using the AGTIOn as an input pin in pulse width measurement mode or pulse period measurement mode, set up the Operation and start count operation. Then start to enter external events from the AGTIOn pin. Invalidate the first measurement and validate the second and later completed measurements.

21.4.5 Digital Filter

When using the digital filter, do not start the timer operation for 5 cycles of the digital filter clock after setting TIPF[1:0] bits and when the TEDGSEL bit in the AGTIOC register changes.

21.4.6 How to Calculate Event Number, Pulse Width, and Pulse Period

- In event counter mode, event number is expressed mathematically as follows:
  Event number = initial value of counter [AGT register] - counter value of active event end
- In pulse width measurement mode, pulse width is expressed mathematically as follows:
  Pulse width = counter value of stopping measurement - counter value of next stopping measurement
- In pulse period measurement mode, input pulse period is expressed mathematically as follows:
  Period of input pulse = (initial value of counter [AGT register] - reading value of the read-out buffer) + 1.

21.4.7 When Count is Forcibly Stopped by TSTOP Bit

After the counter is forcibly stopped by the TSTOP bit in the AGTCR register, do not access the following I/O registers for 1 cycle of the count source:

- AGT
- AGTCMA
- AGTCMB
- AGTCR
- AGTMR1
- AGTMR2.

21.4.8 When Selecting AGTW0 Underflow as the Count Source

Operate according to the following procedures described in this section when selecting the underflow event signal as the count source.

(1) Procedure for starting operation

1. Set AGTW.
2. Start the count operation of AGTW1.
3. Start the count operation of AGTW0.

(2) Procedure for stopping operation

1. Stop the count operation of AGTW0.
2. Stop the count operation of AGTW1.
3. Stop the count source clock of AGTW1 (write 000b in the AGTMR1.TCK[2:0] bits).

21.4.9 Module-stop function

AGTW operation can be disabled or enabled using Module Stop Control Register D (MSTPCRD). The AGTW module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

22. Watchdog Timer (WDT)

22.1 Overview

The Watchdog Timer (WDT) is a 14-bit down counter that can be used to reset the MCU when the counter underflows because the system has run out of control and is unable to refresh the WDT. In addition, the WDT can be used to generate a non-maskable interrupt or an underflow interrupt or watchdog timer reset.

Table 22.1 lists the WDT specifications and Figure 22.1 shows a block diagram.

Table 22.1 WDT specifications

Parameter	Specifications
Count source*1	Peripheral clock (PCLKB)
Clock division ratio	Division by 4, 64, 128, 512, 2048, or 8192
Counter operation	Counting down using a 14-bit down-counter
Condition for starting the counter	• Auto start mode: Counting automatically starts after a reset or after an underflow or refresh error occurs • Register start mode: Counting is started with a refresh by writing to the WDTRR register
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error is generated
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Watchdog timer reset sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt/interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading of the counter value	The down-counter value can be read by the WDTSR register
Event link function (output)	• Down-counter underflow event output • Refresh error event output
Output signal (internal signal)	• Reset output • Interrupt request output • Sleep-mode count stop control output

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 4 × (the frequency of the count clock source after division).

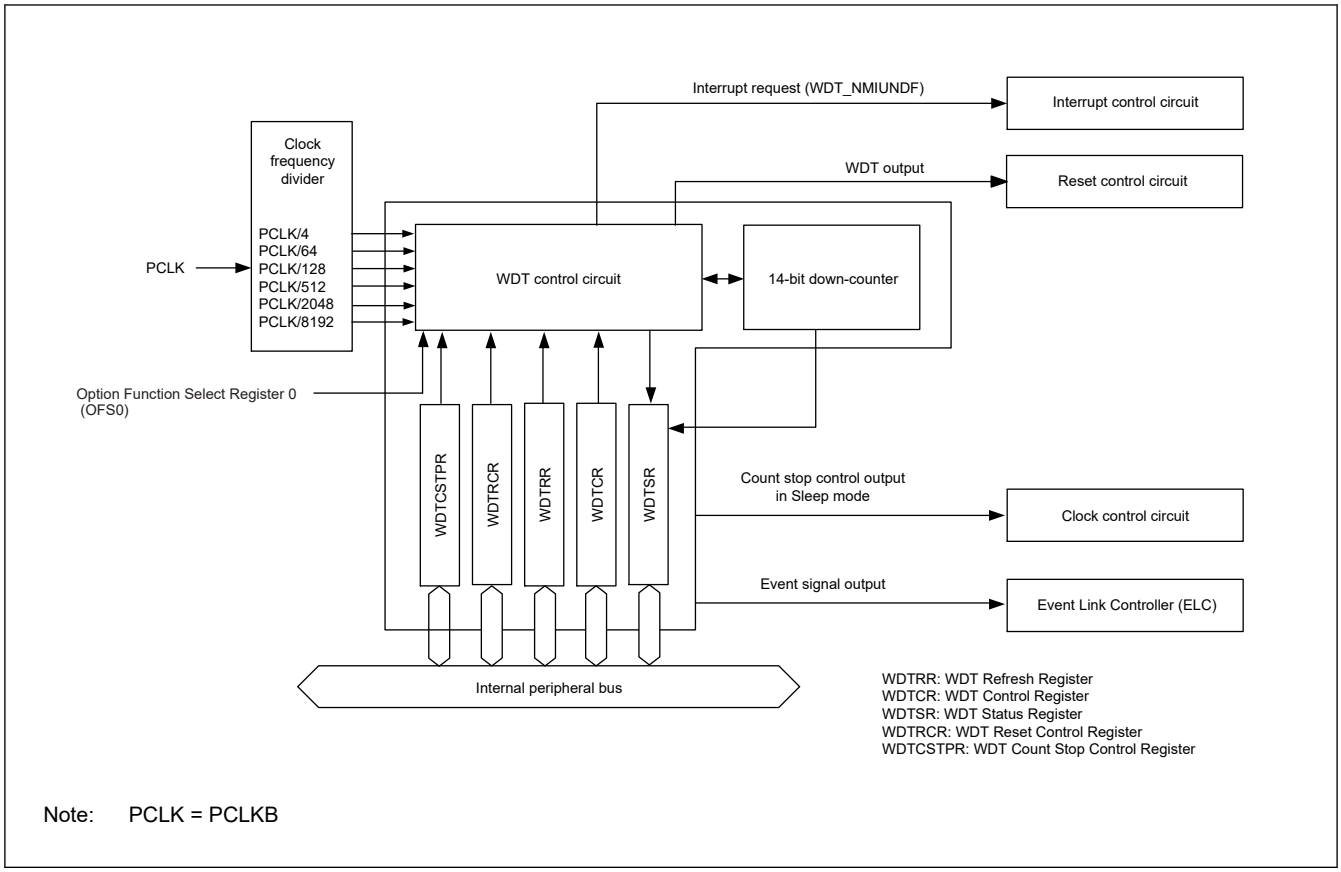


Figure 22.1 WDT block diagram

22.2 Register Descriptions

22.2.1 WDTRR : WDT Refresh Register

Base address: WDT = 0x4004_4200

Offset address: 0x00

Bit position:	7							0
Bit field:								
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	n/a	The down-counter is refreshed by writing 0x00 and then writing 0xFF to this register.	R/W

The WDTRR register refreshes the down-counter of the WDT.

The down-counter of the WDT is refreshed by writing 0x00 and then writing 0xFF to WDTRR register (refresh operation) within the refresh-permitted period.

After the down-counter is refreshed, it starts counting down from the value selected by setting the WDT Timeout Period Select bits (OFS0.WDTTOPS[1:0]) in the Option Function Select Register 0 in auto start mode. In register start mode, counting down starts from the value selected by setting the Timeout Period Select bits (WDTCR.TOPS[1:0]) in the WDT Control Register.

When 0x00 is written, the read value is 0x00. When a value other than 0x00 is written, the read value is 0xFF. For details of the refresh operation, see section 22.3.3. Refresh Operation.

22.2.2 WDTCR : WDT Control Register

Base address: WDT = 0x4004_4200

Offset address: 0x02

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

Bit	Symbol	Function	R/W
1:0	TOPS[1:0]	Timeout Period Select 0 0: 1024 cycles (0x03FF) 0 1: 4096 cycles (0x0FFF) 1 0: 8192 cycles (0x1FFF) 1 1: 16384 cycles (0x3FFF)	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	CKS[3:0]	Clock Division Ratio Select 0x1: PCLKB/4 0x4: PCLKB/64 0xF: PCLKB/128 0x6: PCLKB/512 0x7: PCLKB/2048 0x8: PCLKB/8192 Others: Setting prohibited	R/W
9:8	RPES[1:0]	Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (do not specify window end position).	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	RPSS[1:0]	Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (do not specify window start position).	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

The WDTCR register is used to set the clock division ratio, and window start and end positions for refresh, and the timeout period until the down-counter underflows in register start mode.

Some constraints apply to writing to the WDTCR register. For details, see section 22.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the settings in the WDTCR register are disabled, and the settings in the Option Function Select Register 0 (OFS0) are enabled. The settings for the WDTCR register can also be made in the OFS0 register. For details, see section 22.3.8. Association between Option Function Select Register 0 (OFS0) and WDT Registers.

TOPS[1:0] bits (Timeout Period Select)

The TOPS[1:0] bits select the timeout period, the period until the down-counter underflows, from 1024, 4096, 8192, and 16384 cycles, taking the divided clock specified in the CKS[3:0] bits as 1 cycle. After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the number of PCLKB cycles until the counter underflows.

Table 22.2 lists the relationship between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLKB cycles.

Table 22.2 Timeout period settings

CKS[3:0] bits	TOPS[1:0] bits	Clock division ratio	Timeout period (number of cycles)	PCLKB clock cycles
0x1	00b	PCLKB/4	1024	4096
	01b		4096	16384
	10b		8192	32768
	11b		16384	65536
0x4	00b	PCLKB/64	1024	65536
	01b		4096	262144
	10b		8192	524288
	11b		16384	1048576
0xF	00b	PCLKB/128	1024	131072
	01b		4096	524288
	10b		8192	1048576
	11b		16384	2097152
0x6	00b	PCLKB/512	1024	524288
	01b		4096	2097152
	10b		8192	4194304
	11b		16384	8388608
0x7	00b	PCLKB/2048	1024	2097152
	01b		4096	8388608
	10b		8192	16777216
	11b		16384	33554432
0x8	00b	PCLKB/8192	1024	8388608
	01b		4096	33554432
	10b		8192	67108864
	11b		16384	134217728

CKS[3:0] bits (Clock Division Ratio Select)

The CKS[3:0] bits specify the division ratio of the clock used for the down-counter. The division ratio can be selected from the PCLKB divided by 4, 64, 128, 512, 2048, and 8192. Combined with the TOPS[1:0] bit setting, this allows the WDT to be configured to a count period between 4096 and 134217728 PCLKB clock cycles.

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

The RPES[1:0] bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. Set the window end position to a value less than the value for the window start position (window start position > window end position). If the window start position is set to a value less than or equal to the window end position, the window start position setting is enabled and the window end position is set to 0%.

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

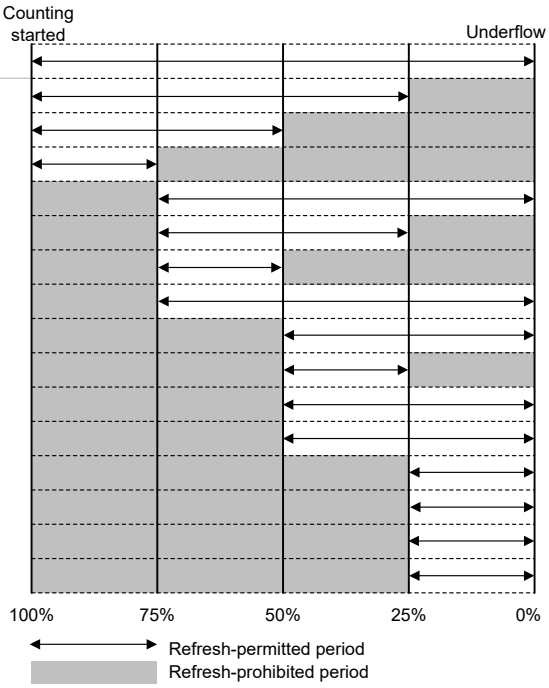
The RPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window start position. Set the window start position to a value greater than the value for the window end position. If the window start position is set to a value less than or equal to the window end position, the window start position setting is enabled and the window end position is set to 0%.

Table 22.3 lists the counter values for the window start and end positions, and Figure 22.2 shows the refresh-permitted period set in the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 22.3 Relationship between the timeout period and window start and end counter values

TOPS[1:0]	Timeout period		Window start and end counter value			
	Cycles	Counter value	100%	75%	50%	25%
00b	1024	0x03FF	0x03FF	0x02FF	0x01FF	0x00FF
01b	4096	0x0FFF	0x0FFF	0x0BFF	0x07FF	0x03FF
10b	8192	0x1FFF	0x1FFF	0x17FF	0x0FFF	0x07FF
11b	16384	0x3FFF	0x3FFF	0x2FFF	0x1FFF	0x0FFF

RPSS[1:0] bits		RPES[1:0] bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75



Note: If window end position setting ≥ window start position setting, the window end position setting is set to 0%.

Figure 22.2 RPSS[1:0] and RPES[1:0] bits setting and refresh-permitted period

22.2.3 WDTSR : WDT Status Register

Base address: WDT = 0x4004_4200

Offset address: 0x04

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

Bit	Symbol	Function	R/W
13:0	CNTVAL[13:0]	Down-Counter Value Value counted by the down-counter	R
14	UNDFF	Underflow Flag 0: No underflow occurred 1: Underflow occurred	R/W*1

Bit	Symbol	Function	R/W
15	REFEF	Refresh Error Flag 0: No refresh error occurred 1: Refresh error occurred	R/W*1

Note 1. Only 0 can be written to clear the flag.

The WDTSR register indicates the counter value of the down-counter and the status of whether an underflow or refresh error occurred in the down-counter.

CNTVAL[13:0] bits (Down-Counter Value)

Read the CNTVAL[13:0] bits to confirm the value of the down-counter. The read value might differ from the actual count by 1.

UNDFF flag (Underflow Flag)

Read the UNDFF flag to confirm whether an underflow occurred in the counter. A value of 1 indicates that the down counter underflowed. Write 0 to the flag to set the value to 0. Writing 1 has no effect.

Clearing of the UNDFF flag takes (N+1) PCLKB cycles. In addition, clearing of the flag is ignored for (N+1) PCLKB cycles after an underflow. N is specified in the WDTCR.CKS[3:0] bits as follows:

- When WDTCR.CKS[3:0] = 0x1, N = 4
- When WDTCR.CKS[3:0] = 0x4, N = 64
- When WDTCR.CKS[3:0] = 0xF, N = 128
- When WDTCR.CKS[3:0] = 0x6, N = 512
- When WDTCR.CKS[3:0] = 0x7, N = 2048
- When WDTCR.CKS[3:0] = 0x8, N = 8192

REFEF flag (Refresh Error Flag)

Read the REFEF flag to confirm whether a refresh error occurred, indicating that a refresh operation was performed during a prohibited period. A value of 1 indicates that a refresh error occurred. Write 0 to the flag to set the value to 0. Writing 1 has no effect.

Clearing of the REFEF flag takes (N+1) PCLKB cycles. In addition, clearing of the flag is ignored for (N+1) PCLKB cycles after a refresh error. N is specified in the WDTCR.CKS[3:0] bits as follows:

- When WDTCR.CKS[3:0] = 0x1, N = 4
- When WDTCR.CKS[3:0] = 0x4, N = 64
- When WDTCR.CKS[3:0] = 0xF, N = 128
- When WDTCR.CKS[3:0] = 0x6, N = 512
- When WDTCR.CKS[3:0] = 0x7, N = 2048
- When WDTCR.CKS[3:0] = 0x8, N = 8192

22.2.4 WDTRCR : WDT Reset Control Register

Base address: WDT = 0x4004_4200

Offset address: 0x06

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

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	RSTIRQS	WDT Behavior Selection 0: Interrupt 1: Reset	R/W

The WDTRCR register controls reset output by a WDT down-counter underflow or interrupt request output.

Some constraints apply to writing to the WDTRCR register. For details, see section 22.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the WDTRCR register settings are disabled, and the settings in the Option Function Select register 0 (OFS0) are enabled. The settings for the WDTRCR register can also be made for the OFS0 register. For details, see section 22.3.8. Association between Option Function Select Register 0 (OFS0) and WDT Registers.

22.2.5 WDTCSTPR : WDT Count Stop Control Register

Base address: WDT = 0x4004_4200

Offset address: 0x08

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

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	SLCSTP	Sleep-Mode Count Stop Control Register 0: Disable count stop 1: Stop count on transition to Sleep mode	R/W

The WDTCSTPR register controls whether to stop the WDT counter in Sleep mode. Some constraints apply to writing to the WDTCSTPR register. For details, see section 22.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.

In auto start mode, the WDTCSTPR register settings are disabled, and the settings in the Option Function Select register 0 (OFS0) are enabled. The settings for the WDTCSTPR register can also be made for the OFS0 register. For details, see section 22.3.8. Association between Option Function Select Register 0 (OFS0) and WDT Registers.

SLCSTP bit (Sleep-Mode Count Stop Control Register)

The SLCSTP bit selects whether to stop counting on transition to Sleep mode.

22.2.6 OFS0 : Option Function Select Register 0

For information on the OFS0 register, see section 22.3.8. Association between Option Function Select Register 0 (OFS0) and WDT Registers.

22.3 Operation

22.3.1 Count Operation in each Start Mode

The WDT has two start modes:

- Auto start mode, in which counting automatically starts after a release from the reset state
- Register start mode, in which counting starts with a refresh by writing to the register.

In auto start mode, counting automatically starts after a release from the reset state according to the settings in the Option Function Select register 0 (OFS0) in the flash.

In register start mode, counting starts with a refresh by writing to the WDTRR register after the respective registers are set after a release from the reset state.

Select auto start mode or register start mode by setting the WDT Start Mode Select bit (OFS0.WDTSTRT) in the OFS0 register.

When the auto start mode is selected, the settings in the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are disabled while the settings in the OFS0 register are enabled.

When the register start mode is selected, the setting for the OFS0 register is disabled while the settings for the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are enabled.

22.3.1.1 Register start mode

When the WDT Start Mode Select bit (OFS0.WDTSTRT) is 1, register start mode is selected, the OFS0 register setting is invalid, and the WDT control register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are enabled.

After the reset state is released, set the following:

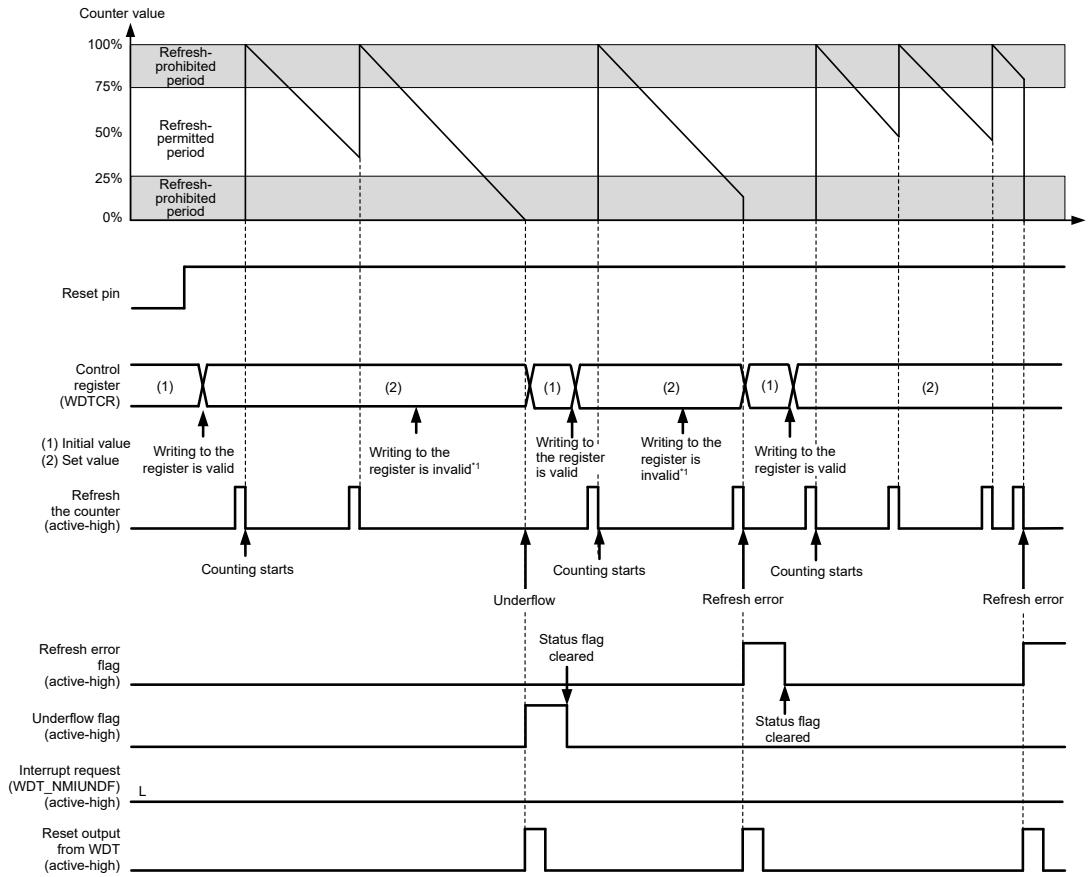
- Clock division ratio in the WDTCR register
- Window start and end positions in the WDTCR register
- Timeout period in the WDTCR register
- Reset output or interrupt request output in the WDTRCR register
- Counter stop control during transitions to Sleep mode in the WDTCSTPR register

The WDT refresh register (WDTRR) refreshes the down counter. As a result, the downcount starts at the value set by the timeout period selection bit (WDTCR.TOPS[1:0]).

Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The WDT does not output the reset signal or non-maskable interrupt request/interrupt request as long as counting continues. However, if the down-counter underflows because the down-counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the WDT outputs the reset signal or a non-maskable interrupt request/interrupt request (WDT_NMIUNDF). Reset output or interrupt request output can be selected in the WDT Reset Interrupt Request Select bit (WDTRCR.RSTIRQS). The interrupt enabled for operating the NMI can be selected in the WDT Underflow/Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 22.3 shows an example of operation under the following conditions:

- Register start mode (OFS0.WDTSTRT = 1)
- WDT reset interrupt request selection (WDTRCR.RSTIRQS = 1)
- The window start position is 75% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b)



Note: Refreshing the counter requires up to 4 count signal cycles from refresh request to execution.
Note 1. See section 22.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers.
Reset pin = RES

Figure 22.3 Operation example in register start mode

22.3.1.2 Auto start mode

When the WDT Start Mode Select bit (OFS0.WDTSTRT) in the Option Function Select Register 0 (OFS0) is 0, auto start mode is selected, the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSTPR) are disabled, and the settings in the OFS0 register are enabled.

Within the reset state, the setting values for the following in the Option Function Select Register 0 (OFS0) are set in the WDT registers:

- Clock division ratio
- Window start and end positions
- Timeout period
- Reset output or interrupt request
- Counter stop control during transition to Sleep mode

When the reset state is released, the down-counter automatically starts counting down from the value set in the WDT Timeout Period Select bits (OFS0.WDTTOPS[1:0]).

Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The WDT does not output the reset signal or non-maskable interrupt

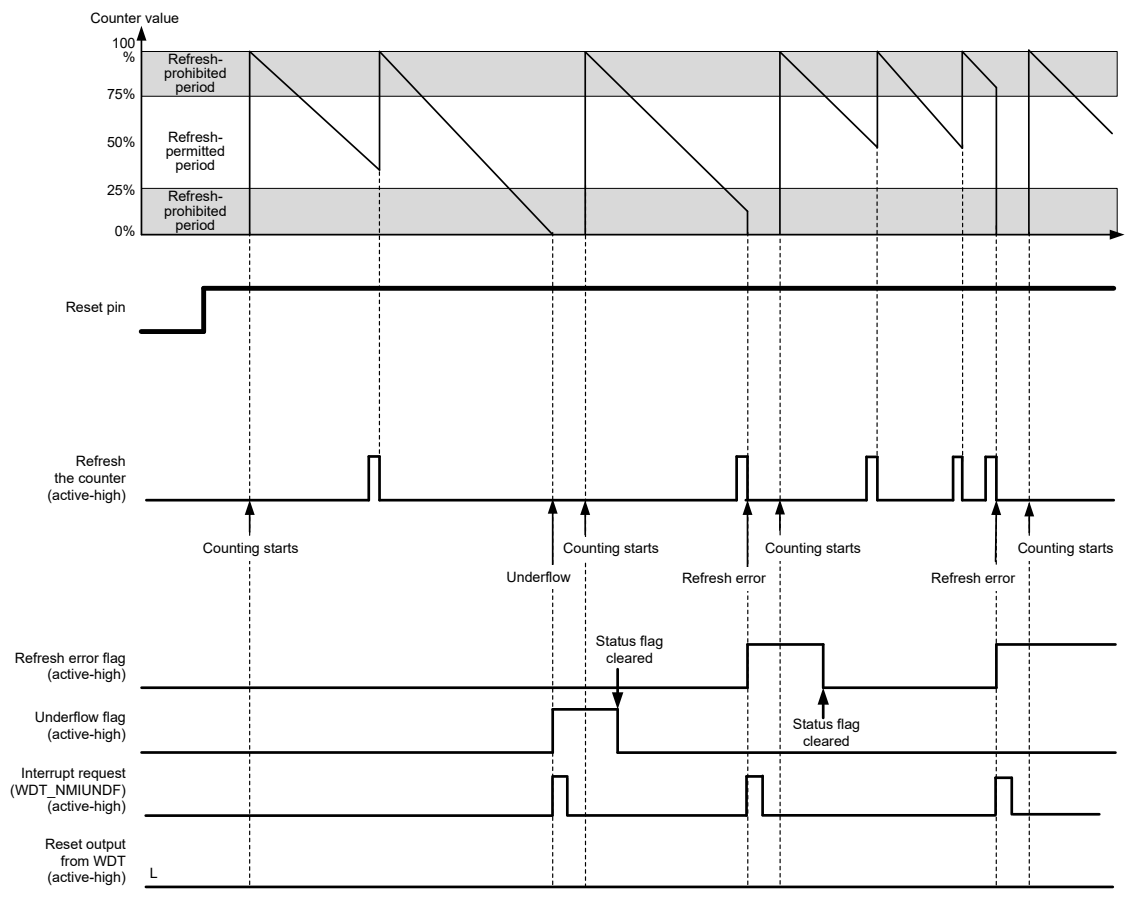
request/interrupt request (WDT_NMIUNDF) as long as the counting continues. However, if the down-counter underflows because refreshing of the down-counter is not possible due to a runaway program or if a refresh error occurs due to refreshing outside the refresh-permitted period, the WDT outputs the reset signal or non-maskable interrupt request/interrupt request (WDT_NMIUNDF).

After the reset signal or non-maskable interrupt request/interrupt request is generated, the counter reloads the timeout period after counting for 1 cycle. The value of the timeout period is set in the down-counter and counting restarts.

Reset output or interrupt request output can be selected by setting the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS). Non-maskable interrupt request or interrupt request can be selected in the WDT Underflow/ Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 22.4 shows an example of operation (non-maskable interrupt) under the following conditions:

- Auto start mode (OFS0.WDTSTRT = 0)
- WDT behavior selection: interrupt (OFS0.WDTRSTIRQS = 0)
- Non-maskable Interrupt: IWDT Underflow/Refresh Error Interrupt Enabled (NMIER.WDTEN = 1)
- The window start position is 75% (OFS0.WDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.WDTRPES[1:0] = 10b)



Note: Refreshing the counter requires up to 4 count signal cycles from refresh request to execution.
Note: Reset pin = RES

Figure 22.4 Operation example in auto start mode

22.3.2 Controlling Writes to the WDTCR, WDTRCR, and WDTCSTPR Registers

Writing to the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), or WDT Count Stop Control Register (WDTCSTPR) is possible once each between the release from the reset state and the first refresh operation.

After a refresh (counting starts) or a write to WDTCR, WDTRCR or WDTCSTPR register, the protection signal in the WDT becomes 1 to protect WDTCR, WDTRCR and WDTCSTPR register against subsequent write attempts. This protection is released by the reset source of the WDT. With other reset sources, the protection is not released.

Figure 22.5 shows control waveforms produced in response to writing to the WDTCR.

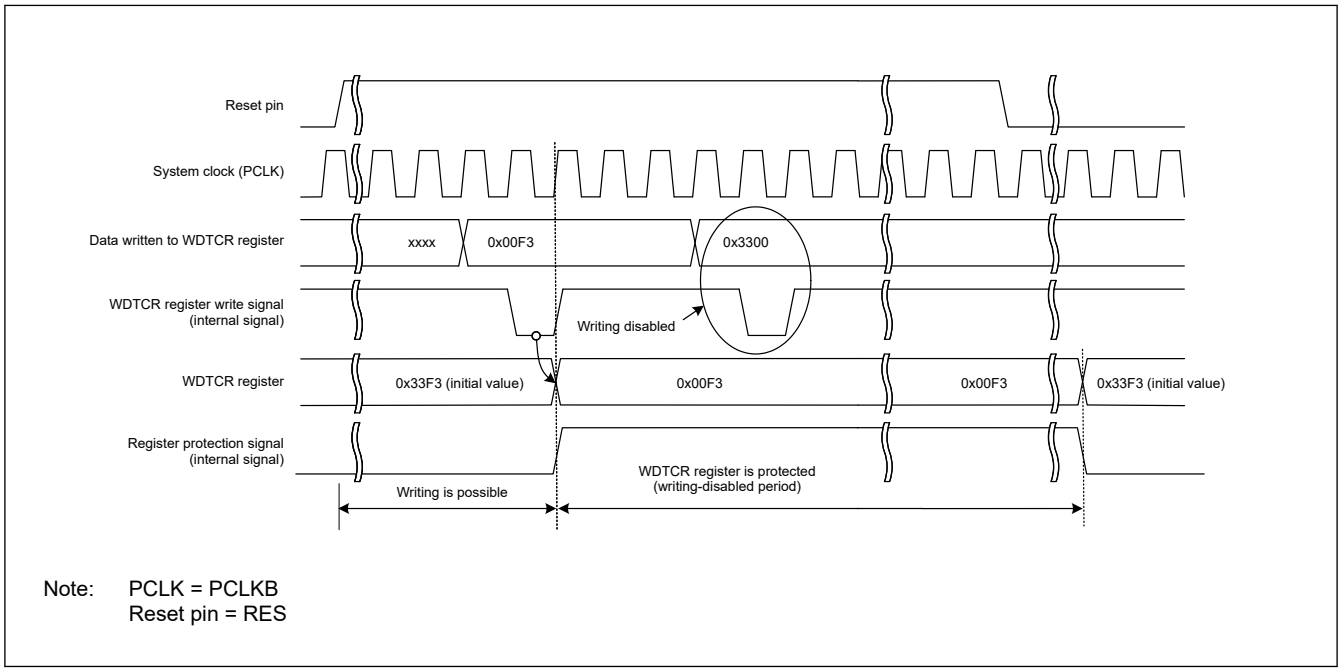


Figure 22.5 Control waveforms produced in response to writes to the WDTCR register

22.3.3 Refresh Operation

To refresh the down counter and start the counting operation, write to the WDT Refresh Register (WDTRR) in the order of values from 0x00 to 0xFF. If a value other than 0xFF is written after 0x00, the down-counter is not refreshed. If an invalid value is written, refreshing is performed normally by writing to the WDTRR register in the order of values from 0x00 to 0xFF.

Correct refreshing is also performed when a register other than WDTRR is accessed or WDTRR is read between writing 0x00 and writing 0xFF to WDTRR. Writes to refresh the counter must be made within the refresh-permitted period, and this is determined by the 0xFF write. For this reason, correct refreshing is performed even when 0x00 is written outside the refresh-permitted period.

[Example write sequences that are valid for refreshing the counter]

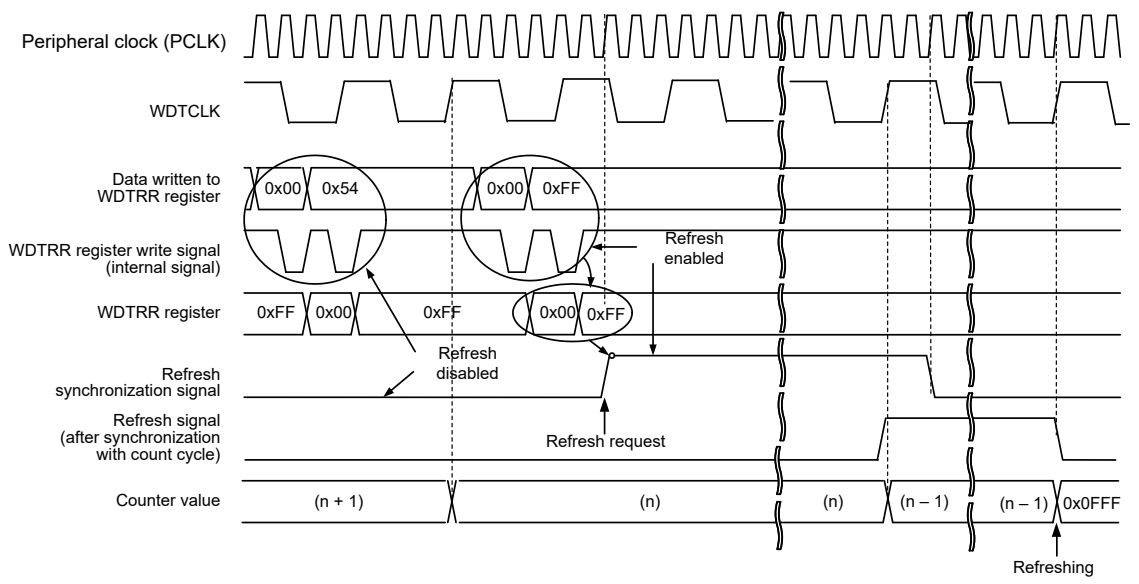
- 0x00 → 0xFF
- 0x00 ((n–1)th time) → 0x00 (nth time) → 0xFF
- 0x00 → access to another register or read from WDTRR → 0xFF

[Example write sequences that are invalid for refreshing the counter]

- 0x23 (a value other than 0x00) → 0xFF
- 0x00 → 0x54 (a value other than 0xFF)
- 0x00 → 0xAA (0x00 and a value other than 0xFF) → 0xFF

After 0xFF is written to the WDT Refresh Register (WDTRR), refreshing the down-counter requires up to 4 cycles of the signal for counting. To meet this requirement, complete writing 0xFF to WDTRR 4 count cycles before the down-counter underflows.

Figure 22.6 shows the WDT refresh-operation waveforms when the clock division ratio is PCLKB/64.



Note: PCLK = PCLKB

Figure 22.6 WDT refresh operation waveforms when WDTCR.CKS[3:0] = 0x4 and WDTCR.TOPS[1:0] = 01b

Note: When setting the refresh time, consider the oscillation accuracy of the clock sources of the PCLKB and WDTCLK. Set values which ensure that refreshing is possible even when the frequency varies in the range of error of the oscillation accuracy.

22.3.4 Status Flags

The refresh error (WDTSR.REFEF) and underflow (WDTSR.UNDFF) flags retain the source of the interrupt request from the WDT. After a release from the interrupt request generation, read the WDTSR.REFEF and WDTSR.UNDFF flags to check for the interrupt source. For each flag, writing 0 clears the bit. Writing 1 has no effect. Leaving the status flags unchanged does not affect operation. If the flags are not cleared at the next interrupt request from the WDT, the earlier interrupt source is cleared and the new interrupt source is written. For the time period between when 0 is written in each flag and when its value is reflected, see section 22.2.3. WDTSR : WDT Status Register.

22.3.5 Reset Output

When the Reset Interrupt Select bit (WDTRCR.RSTIRQS) is set to 1 in register start mode, or when the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 1 in auto start mode, a reset signal is output for 1 cycle count when an underflow in the down-counter or a refresh error occurs.

In register start mode, the down-counter is initialized (all bits set to 0) and stopped in that state after output of a reset signal. After the reset state is released and the program is restarted, the counter is set up again and counting down starts again with a refresh. In auto start mode, counting down starts automatically after the reset state is released.

22.3.6 Interrupt Sources

When the Reset Interrupt Select bit (WDTRCR.RSTIRQS) is set to 0 in register start mode or when the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 0 in auto start mode, an interrupt (WDT_NMIUNDF) signal is generated when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt or an interrupt. For details, see section 12, Interrupt Controller Unit (ICU).

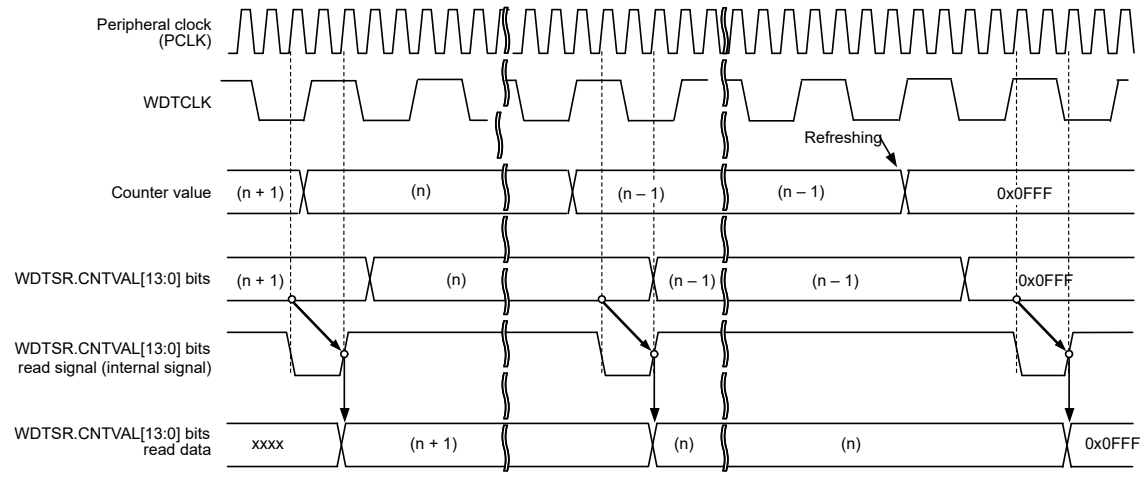
Table 22.4 WDT interrupt source

Name	Interrupt source	Interrupt to CPU	Start DTC
WDT_NMIUNDF	• Down-counter underflow • Refresh error	Possible	Not possible

22.3.7 Reading the Down-Counter Value

The WDT stores the counter value in the down-counter value bits (WDTSR.CNTVAL[13:0]) of the WDT Status Register. Check these bits to obtain the counter value. The read value of the down-counter might differ from the actual count by one.

Figure 22.7 shows the processing for reading the WDT down-counter value when the clock division ratio is PCLKB/64.



Note: PCLK = PCLKB

Figure 22.7 Processing for reading WDT down-counter value when WDTCR.CKS[3:0] = 0x4 and WDTCR.TOPS[1:0] = 01b

22.3.8 Association between Option Function Select Register 0 (OFS0) and WDT Registers

Table 22.5 lists the association between the Option Function Select Register 0 (OFS0) used in auto start mode, and the registers used in register start mode. Do not change the OFS0 register setting during WDT operation. For details on the Option Function Select Register 0 (OFS0), see section 6.2.1. OFS0 : Option Function Select Register 0.

Table 22.5 Association between Option Function Select Register 0 (OFS0) and the WDT registers

Control target	Function	OFS0 register (enabled in auto start mode) OFS0.WDTSTRT = 0	WDT registers (enabled in register start mode) OFS0.WDTSTRT = 1
Down-counter	Timeout period selection	OFS0.WDTTOPS[1:0]	WDTCR.TOPS[1:0]
	Clock division ratio selection	OFS0.WDTCKS[3:0]	WDTCR.CKS[3:0]
	Window start position selection	OFS0.WDTRPSS[1:0]	WDTCR.RPSS[1:0]
	Window end position selection	OFS0.WDTRPES[1:0]	WDTCR.RPES[1:0]
Reset output or interrupt request output	Select a reset interrupt request	OFS0.WDTRSTIRQS	WDTRCR.RSTIRQS
Count stop	Sleep mode count stop control	OFS0.WDTSTPCTL	WDTCSTPR.SLCSTP

22.4 Output to the Event Link Controller (ELC)

The WDT is capable of a link operation for the previously specified module when interrupt request signal is used as an event signal by the ELC. The event signal is output by the counter underflow and refresh error. An event signal is output regardless of the settings of the WDTRCR.RSTIRQS bit in register start mode or the OFS0.WDTRSTIRQS bit in auto start mode. An event signal can also be output when the next interrupt source is generated while the Refresh Error flag (WDTSR.REFEF) or Underflow flag (WDTSR.UNDFF) is 1. For details, see section 16, Event Link Controller (ELC).

22.5 Usage Notes

22.5.1 ICU Event Link Setting Register n (IELSRn) Setting

Setting 0x06 to ICU Event Link Setting Register n (ICU.IELSRn) is prohibited when WDT reset interrupt request selection resets (OFS0.WDTRSTIRQS = 1 or WDTRCR.RSTIRQS = 1) or when enabling event link operation (ELSRn.ELS[7:0] = 0x18).

23. Independent Watchdog Timer (IWDT)

23.1 Overview

The Independent Watchdog Timer (IWDT) consists of a 14-bit down counter that must be serviced periodically to prevent counter underflow. The IWDT provides functionality to reset the MCU or to generate a non-maskable interrupt or an underflow interrupt. Because the timer operates with an independent, dedicated clock source, it is particularly useful in returning the MCU to a known state as a fail-safe mechanism when the system runs out of control. The IWDT can be triggered automatically by a reset, underflow, refresh error, or a refresh of the count value in the registers.

The IWDT functions differ from those of the WDT in the following respects:

- The divided IWDT-dedicated clock (IWDTCLK) is used as the count source (not affected by PCLKB)
- IWDT does not support register start mode

Table 23.1 lists the IWDT specifications and Figure 23.1 shows a block diagram.

Table 23.1 IWDT specifications

Parameter	Description
Count source*1	IWDT-dedicated clock (IWDTCLK)
Clock division ratio	Division by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit down-counter
Condition for starting the counter	• Counting automatically starts after a reset
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error is generated (counting restarts automatically).
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Reset output sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error).
Non-maskable interrupt/interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error).
Reading the counter value	The down-counter value can be read by the IWDTSR register
Event link function	• Down-counter underflow event output • Refresh error event output.
Output signal (internal signal)	• Reset output • Interrupt request output • Sleep-mode count stop control output.
Auto start mode	Configurable to the following triggers: • Clock frequency division ratio after a reset (OFS0.IWDTCKS[3:0] bits) • Timeout period of the Independent Watchdog Timer (OFS0.IWDTTOPS[1:0] bits) • Window start position in the Independent Watchdog Timer (OFS0.IWDTRPSS[1:0] bits) • Window end position in the Independent Watchdog Timer (OFS0.IWDTRPES[1:0] bits) • Reset output or interrupt request output (OFS0.IWDTRSTIRQS bit) • Down-count stop function at transition to Sleep, Snooze, or Software Standby mode (OFS0.IWDTSTPCTL bit).

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 4 × (the frequency of the count clock source after division).

The bus interface and registers operate with PCLKB, and the 14-bit counter and control circuits operate with IWDTCLK.

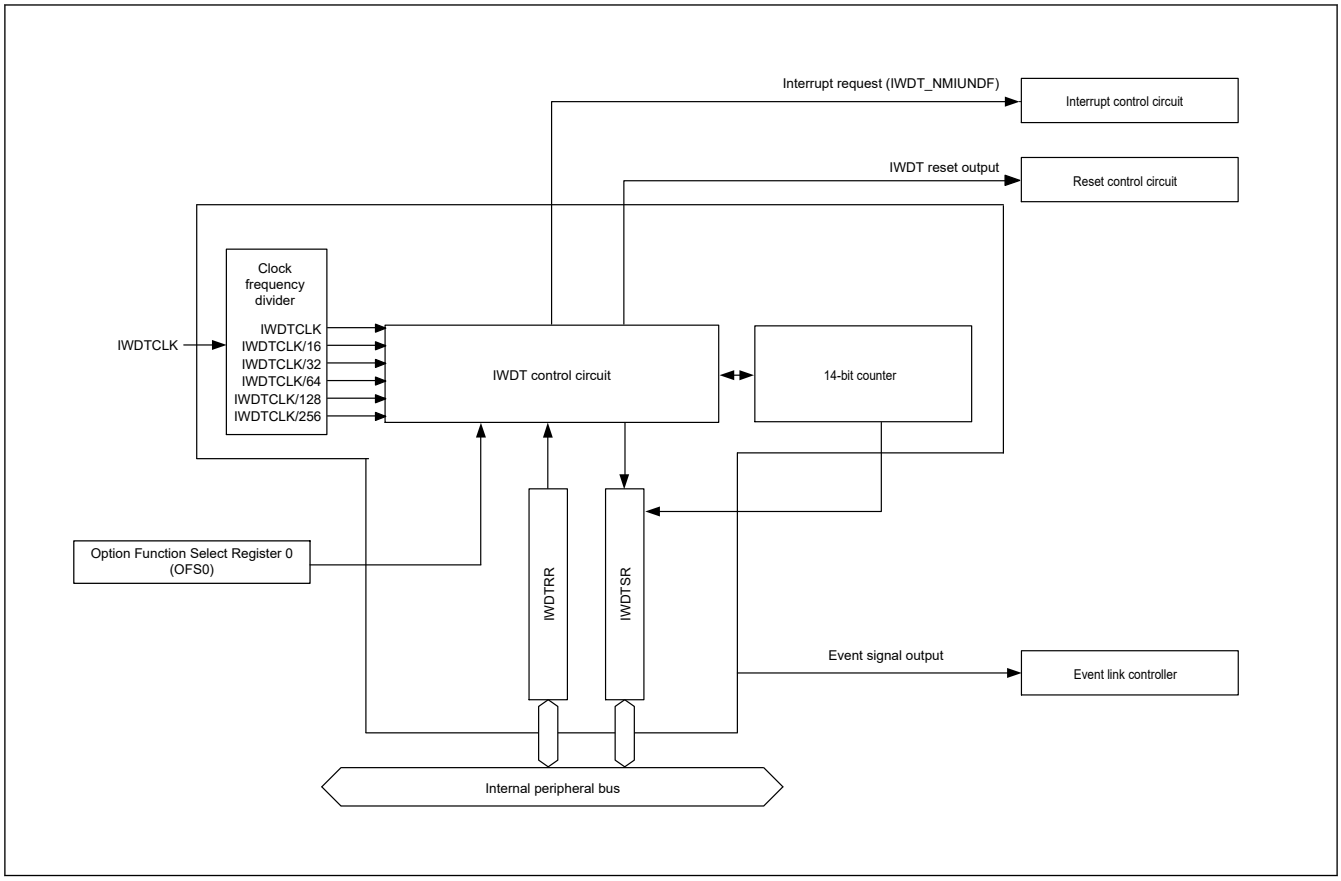


Figure 23.1 IWDT block diagram

23.2 Register Descriptions

23.2.1 IWDTRR : IWDT Refresh Register

Base address: IWDT = 0x4004_4400

Offset address: 0x00

Bit position:	7							0
Bit field:								
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	n/a	The down-counter is refreshed by writing 0x00 and then writing 0xFF to this register	R/W

The IWDTRR register refreshes the down-counter of the IWDT. The down-counter of the IWDT is refreshed by writing 0x00 and then writing 0xFF to IWDTRR (refresh operation) within the refresh-permitted period. After the down-counter is refreshed, it starts counting down from the value selected in the IWDT Timeout Period Select bits (OFS0.IWDTTOPS[1:0]) in the Option Function Select Register 0 (OFS0).

When 0x00 is written, the read value is 0x00. When a value other than 0x00 is written, the read value is 0xFF. For details of the refresh operation, see section 23.3.2. Refresh Operation.

23.2.2 IWDTSR : IWDT Status Register

Base address: IWDT = 0x4004_4400

Offset address: 0x04

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

Bit	Symbol	Function	R/W
13:0	CNTVAL[13:0]	Down-counter Value Value counted by the down-counter	R
14	UNDFF	Underflow Flag 0: No underflow occurred 1: Underflow occurred	R/W*1
15	REFEF	Refresh Error Flag 0: No refresh error occurred 1: Refresh error occurred	R/W*1

Note 1. Only 0 can be written to clear the flag.

The IWDTSR register indicates the counter value of the down-counter and whether an underflow or refresh error occurred in the down-counter.

CNTVAL[13:0] bits (Down-counter Value)

Read the CNTVAL[13:0] bits to confirm the value of the down-counter. The read value might differ from the actual count by 1.

UNDFF flag (Underflow Flag)

Read the UNDFF flag to confirm whether an underflow occurred in the down-counter. The value 1 indicates that the down-counter underflowed. Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

Clearing of the UNDFF flag takes (N + 2) IWDTCLK cycles and 2 PCLKB cycles. In addition, clearing of this flag is ignored for (N + 2) IWDTCLK cycles after an underflow. N is specified in the IWDTCKS[3:0] bits as follows:

- When OFS0.IWDTCKS[3:0] = 0x0, N = 1
- When OFS0.IWDTCKS[3:0] = 0x2, N = 16
- When OFS0.IWDTCKS[3:0] = 0x3, N = 32
- When OFS0.IWDTCKS[3:0] = 0x4, N = 64
- When OFS0.IWDTCKS[3:0] = 0xF, N = 128
- When OFS0.IWDTCKS[3:0] = 0x5, N = 256.

REFEF flag (Refresh Error Flag)

Read the REFEF flag to confirm whether a refresh error occurred. This indicates that a refresh operation was performed during a prohibited period. The value 1 indicates that a refresh error occurred. Write 0 to the REFEF flag to set the value to 0. Writing 1 has no effect.

Clearing of the REFEF flag takes (N + 2) IWDTCLK cycles and 2 PCLKB cycles. In addition, clearing of this flag is ignored for (N + 2) IWDTCLK cycles following a refresh error. N is specified in the IWDTCKS[3:0] bits as follows:

- When OFS0.IWDTCKS[3:0] = 0x0, N = 1
- When OFS0.IWDTCKS[3:0] = 0x2, N = 16
- When OFS0.IWDTCKS[3:0] = 0x3, N = 32
- When OFS0.IWDTCKS[3:0] = 0x4, N = 64
- When OFS0.IWDTCKS[3:0] = 0xF, N = 128
- When OFS0.IWDTCKS[3:0] = 0x5, N = 256.

23.2.3 OFS0 : Option Function Select Register 0

For information on the Option Function Select Register 0 (OFS0), see section 6.2.1. OFS0 : Option Function Select Register 0.

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

The IWDTTOPS[1:0] bits select the timeout period, that is, the period until the down-counter underflows, from 128, 512, 1024, or 2048 cycles, taking the divided clock specified in the IWDTCKS[3:0] bits as 1 cycle.

After the down-counter is refreshed, the combination of the IWDTCKS[3:0] and IWDTTOPS[1:0] bits determines the number of IWDTCLK cycles until the counter underflows.

Table 23.2 lists the relationship between the IWDTCKS[3:0] and IWDTTOPS[1:0] bit settings, the timeout period, and the number of IWDTCLK cycles.

Table 23.2 Timeout period settings

IWDTCKS[3:0] bits				IWDTTOPS[1:0] bits		Clock division ratio	Timeout period (number of cycles)	IWDTCLK cycles
b7	b6	b5	b4	b3	b2			
0	0	0	0	0	0	IWDTCLK	128	128
				0	1		512	512
				1	0		1024	1024
				1	1		2048	2048
0	0	1	0	0	0	IWDTCLK/16	128	2048
				0	1		512	8192
				1	0		1024	16384
				1	1		2048	32768
0	0	1	1	0	0	IWDTCLK/32	128	4096
				0	1		512	16384
				1	0		1024	32768
				1	1		2048	65536
0	1	0	0	0	0	IWDTCLK/64	128	8192
				0	1		512	32768
				1	0		1024	65536
				1	1		2048	131072
1	1	1	1	0	0	IWDTCLK/128	128	16384
				0	1		512	65536
				1	0		1024	131072
				1	1		2048	262144
0	1	0	1	0	0	IWDTCLK/256	128	32768
				0	1		512	131072
				1	0		1024	262144
				1	1		2048	524288

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

The IWDTCKS[3:0] bits specify the division ratio of the clock used for the down-counter. The division ratio can be selected from the IWDT-dedicated clock (IWDTCLK) divided by 1, 16, 32, 64, 128, and 256. Combined with the IWDTTOPS[1:0] bit setting, the IWDT can be configured to a count period between 128 and 524,288 IWDTCLK cycles.

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

The IWDTRPES[1:0] bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. Set the window end position to a value less than the window start position (window start position > window end position). If the window start position is set to a value less than or equal to the window end position, the window start position setting is enabled and the window end position is set to 0%.

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

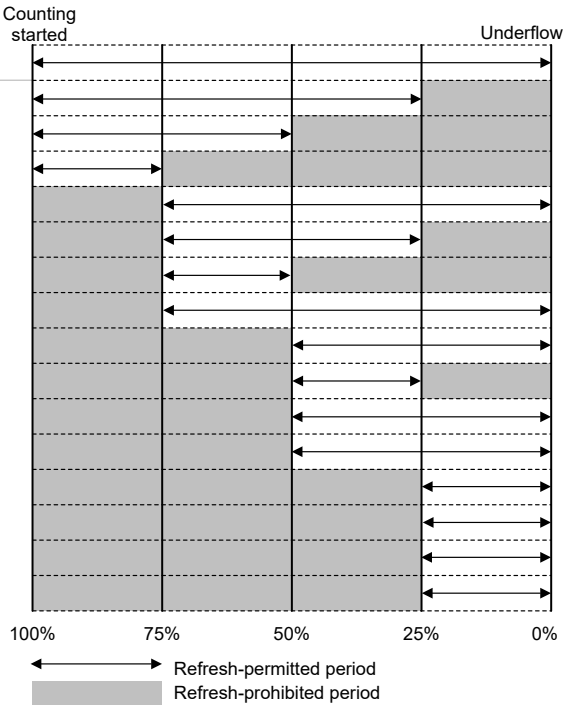
The IWDTRPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window start position. Set the window start position to a value greater than the window end position. If the window start position is set to a value less than or equal to the window end position, the window start position setting is enabled and the window end position is set to 0%.

Table 23.3 lists the counter values for the window start and end positions, and Figure 23.2 shows the refresh-permitted period set in the IWDTRPSS[1:0], IWDTRPES[1:0], and IWDTTOPS[1:0] bits.

Table 23.3 Relationship between the timeout period and window start and end counter values

IWDTTOPS[1:0] bits		Timeout period		Window start and end counter value			
b3	b2	Cycles	Counter value	100%	75%	50%	25%
0	0	128	0x007F	0x007F	0x005F	0x003F	0x001F
0	1	512	0x01FF	0x01FF	0x017F	0x00FF	0x007F
1	0	1024	0x03FF	0x03FF	0x02FF	0x01FF	0x00FF
1	1	2048	0x07FF	0x07FF	0x05FF	0x03FF	0x01FF

IWDTRPSS[1:0] bits		IWDTRPES[1:0] bits		Window	
b11	b10	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75



Note: If window end position setting ≥ window start position setting, the window end position setting is set to 0%.

Figure 23.2 IWDTRPSS[1:0] and IWDTRPES[1:0] bit settings and refresh-permitted period

IWDTRSTIRQS bit (IWDT Reset Interrupt Request Select)

The IWDTRSTIRQS bit specifies the behavior when an underflow or a refresh error occurs. Setting 1 selects reset output. Setting 0 selects interrupt.

IWDTSTPCTL bit (IWDT Stop Control)

The IWDTSTPCTL bit selects whether to stop counting on transition to Sleep, Snooze, or Software Standby mode.

23.3 Operation

23.3.1 Auto Start Mode

When the IWDT Start Mode Select bit (OFS0.IWDTSTRT) in the Option Function Select Register 0 is 0, auto start mode is selected, otherwise the IWDT is disabled.

Within the reset state, the setting values for the following in the Option Function Select Register 0 (OFS0) are set in the IWDT registers:

- Clock division ratio (OFS0.IWDTCKS[3:0])
- Window start and end positions (OFS0.IWDTRPSS[1:0], OFS0.IWDTRPES[1:0])
- Timeout period (OFS0.IWDTTOPS[1:0])
- Reset output or interrupt request (OFS0.IWDTRSTIRQS)

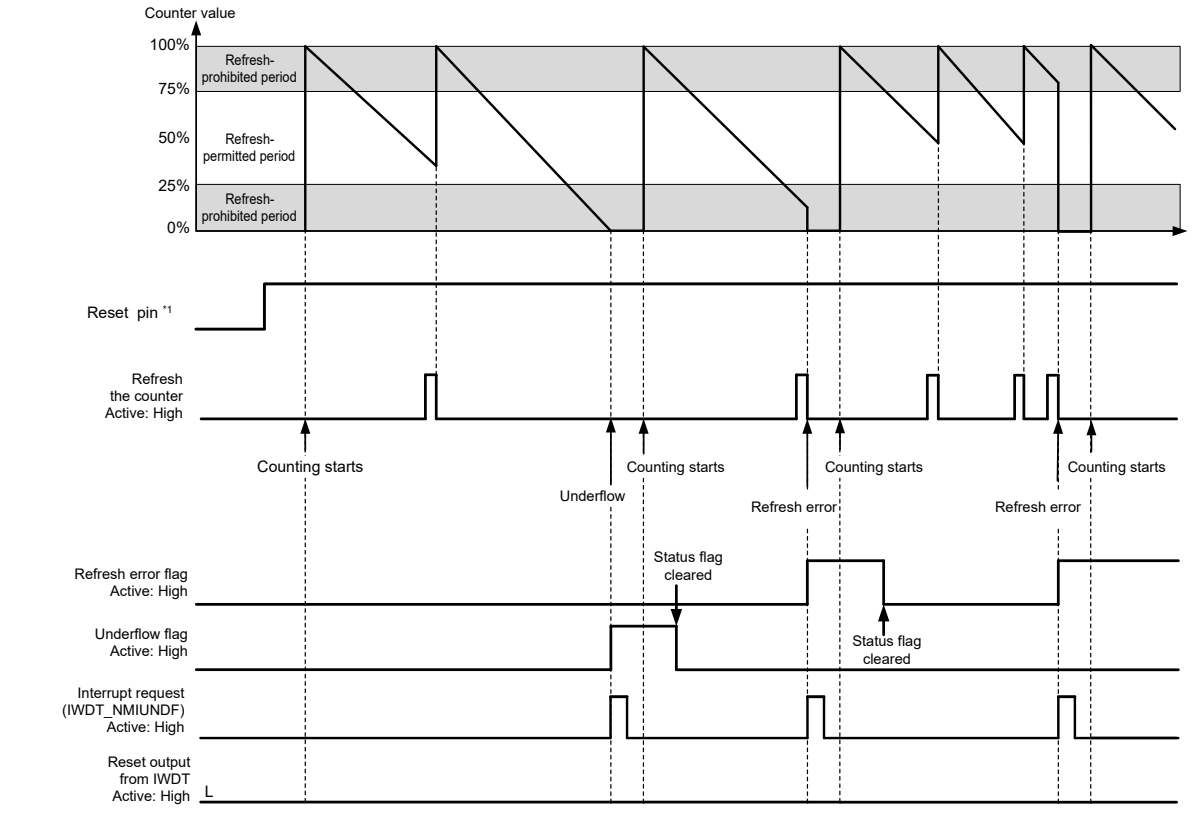
When the reset state is released, the counter automatically starts counting down from the value selected in the IWDT Timeout Period Select bits (OFS0.IWDTTOPS[1:0]).

After that, as long as the program continues normal operation and the counter is refreshed within the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and down-counting continues. The IWDT does not output the reset signal as long as this procedure continues. However, if the counter underflows because the program crashed or because a refresh error occurred when an attempt is made to refresh outside the refresh-permitted period, the IWDT asserts the reset signal or non-maskable interrupt request/interrupt request (IWDT_NMIUNDF).

After the reset signal or non-maskable interrupt request/interrupt request is generated, the counter reloads the timeout period after counting for 1 cycle, the value of the timeout period is set in the down-counter and counting starts. The reset output or interrupt request output can be selected with the IWDT Reset Interrupt Request Select bit (OFS0.IWDTRSTIRQS). The interrupt enabled for operating the NMI can be selected with the IWDT Underflow/Refresh Error Interrupt Enable bit (NMIER.IWDTEN).

Figure 23.3 shows an example of operation under the following conditions:

- Auto start mode (OFS0.IWDTSTRT = 0)
- IWDT behavior selection: interrupt (OFS0.IWDTRSTIRQS = 0)
- Non-maskable Interrupt: IWDT Underflow/Refresh Error Interrupt Enabled (NMIER.IWDTEN = 1)
- The window start position is 75% (OFS0.IWDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.IWDTRPES[1:0] = 10b).



Note 1. Reset pin = RES

Figure 23.3 Operation example in auto start mode

23.3.2 Refresh Operation

To refresh the down counter and start the counting operation, write to the IWDT Refresh Register (IWDTRR) in the order of values from 0x00 to 0xFF. If a value other than 0xFF is written after 0x00, the down-counter is not refreshed. If an invalid value is written, refreshing is performed normally by writing to the IWDTRR register in the order of values from 0x00 to 0xFF.

When writes are made in the order of 0x00 (first time) → 0x00 (second time), and if 0xFF is written after that, the writing order 0x00 → 0xFF is satisfied. Writing 0x00 ((n - 1)th time) →0x00 (nth time) → 0xFF is valid, and the refresh is performed correctly. Even when the first value written before 0x00 is not 0x00, correct refreshing is performed as long as the operation contains the write sequence of 0x00 → 0xFF.

Correct refreshing is also performed regardless of whether a register other than IWDTRR is accessed or IWDTRR is read between writing 0x00 and writing 0xFF to IWDTRR. Writes to refresh the counter must be made within the refresh-permitted period. Whether writing is done within the refresh-permitted period is determined when 0xFF is written. For this reason, correct refreshing is performed even when 0x00 is written outside the refresh-permitted period.

[Example write sequences that are valid to refresh the counter]

- 0x00 → 0xFF
- 0x00 ((n - 1)th time) → 0x00 (nth time) → 0xFF
- 0x00 → access to another register or read from IWDTRR → 0xFF.

[Example write sequences that are not valid to refresh the counter]

- 0x23 (a value other than 0x00) → 0xFF

- 0x00 → 0x54 (a value other than 0xFF)
- 0x00 → 0xAA (0x00 and a value other than 0xFF) → 0xFF.

After 0xFF is written to the IWDTRR register, refreshing the counter requires up to 4 cycles of the signal for counting (the IWDT-Dedicated Clock Frequency Division Ratio Select bits (OFS0.IWDTCKS[3:0]) to determine how many cycles of the IWDT-dedicated clock (IWDTCLK) make up 1 cycle for counting. To meet this requirement, writing 0xFF to the IWDTRR must be completed 4 count cycles before the end of the refresh-permitted period or a down-counter underflow. The value of the counter can be checked with the counter bits (IWDTSR.CNTVAL[13:0]).

[Example refreshing timings]

- When the window start position is set to 0x01FF, even if 0x00 is written to IWDTRR before 0x01FF is reached at (0x0202, for example), refreshing occurs if 0xFF is written to IWDTRR after the value of the IWDTSR.CNTVAL[13:0] bits reaches 0x01FF
- When the window end position is set to 0x01FF, refreshing occurs if 0x0203 (4 count cycles before 0x01FF) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 0x00 → 0xFF to IWDTRR
- When the refresh-permitted period continues until count 0x0000, refreshing can be performed immediately before an underflow. In this case, if 0x0003 (4 count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 0x00 → 0xFF to IWDTRR, no underflow occurs and refreshing is performed.

Figure 23.4 shows the IWDT refresh-operation waveforms when PCLKB > IWDTCLK and the clock division ratio is IWDTCLK.

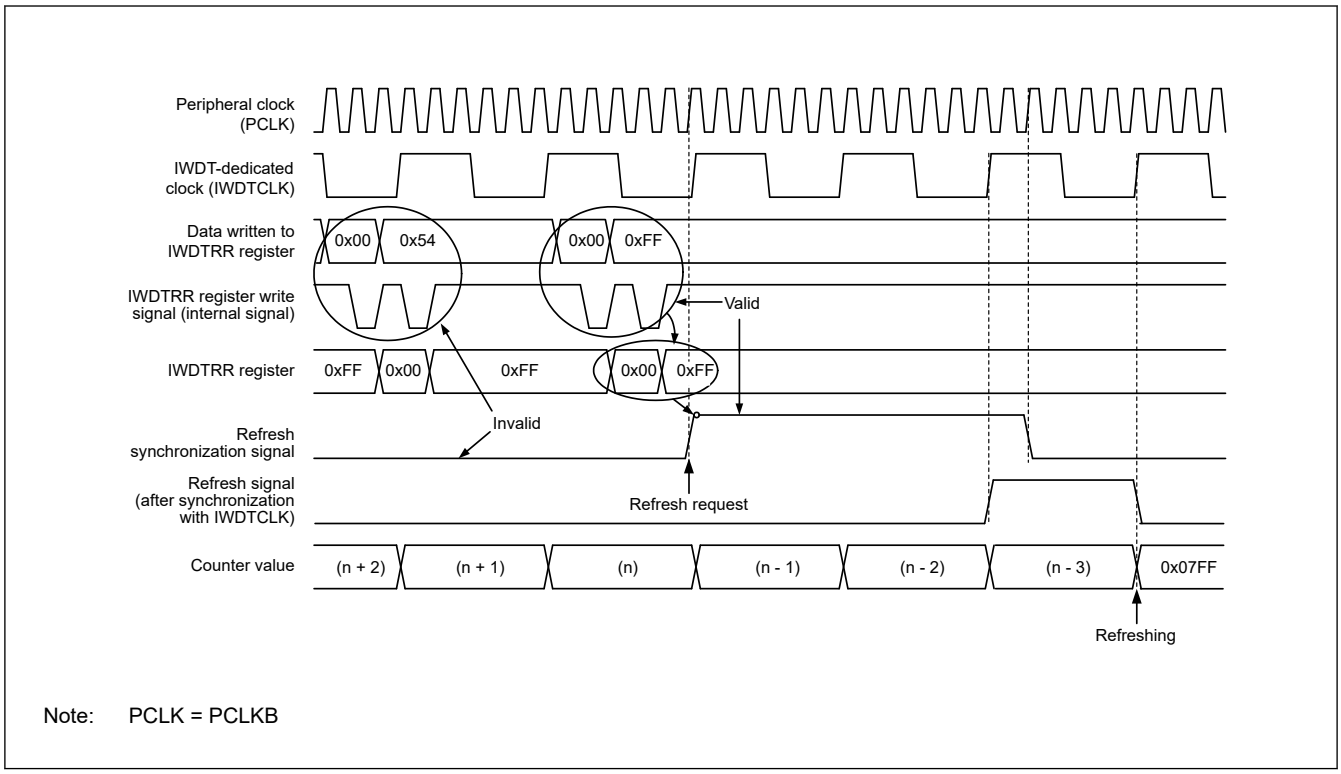


Figure 23.4 IWDT refresh operation waveforms when OFS0.IWDTCKS[3:0] = 0000b, OFS0.IWDTTOPS[1:0] = 11b

23.3.3 Status Flags

The refresh error (IWDTSR.REFEF) and underflow (IWDTSR.UNDFF) flags retain the source of the interrupt request from the IWDT. Therefore, after a release from the interrupt request generation, read the IWDTSR.REFEF and UNDFF flags to check for the interrupt source. For each flag, writing 0 clears the bit and writing 1 has no effect.

Leaving the status flags unchanged does not affect operation. If the flags are not cleared at the time of the next interrupt request from the IWDT, the earlier interrupt source is cleared and the new interrupt source is written. For the time period between when 0 is written in each flag and when its value is reflected, see section 23.2.2. IWDTSR : IWDT Status Register.

23.3.4 Reset Output

When the IWDT Reset Interrupt Request Select bit (OFS0.IWDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 1, a reset signal is output when an underflow in the counter or a refresh error occurs. Counting down automatically starts after the reset output.

23.3.5 Interrupt Sources

When the IWDT Reset Interrupt Request Select bit (OFS0.IWDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 0, an interrupt (IWDT_NMIUNDF) signal occurs when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt or an interrupt. For details, see section 12, Interrupt Controller Unit (ICU).

Table 23.4 IWDT interrupt source

Name	Interrupt source	Interrupt to CPU	Start DTC
IWDT_NMIUNDF	• Down-counter underflow • Refresh error	Possible	Not possible

23.3.6 Reading the Down-Counter Value

As the counter is a IWDT-dedicated clock (IWDTCLK), the counter value cannot be read directly. The IWDT synchronizes the counter value with the peripheral clock (PCLKB) and stores it in the down-counter value bits (IWDTSR.CNTVAL[13:0]) of the IWDT Status Register. Check these bits to obtain the counter value indirectly.

Reading the counter value requires multiple PCLKB clock cycles (up to 4 clock cycles), and the read counter value might differ from the actual counter value by a value of one count.

Figure 23.5 shows the processing for reading the IWDT counter value when PCLKB > IWDTCLK and the clock division ratio is IWDTCLK.

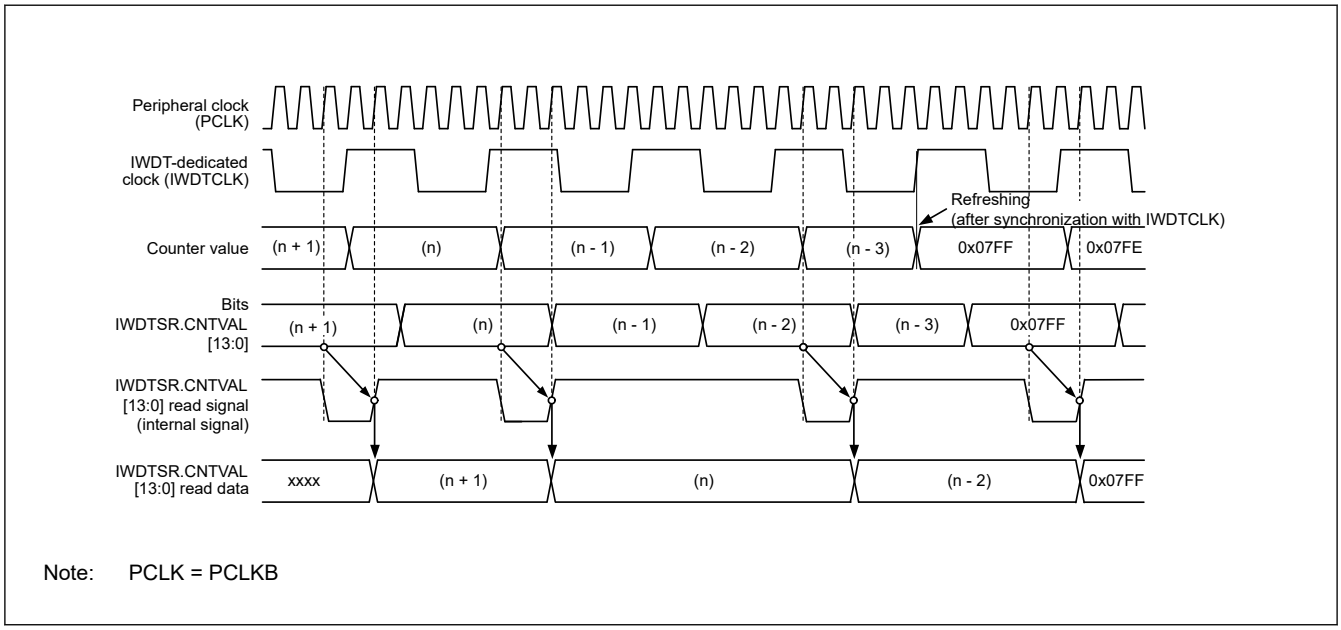


Figure 23.5 Processing for reading IWDT counter value when OFS0.IWDTCKS[3:0] = 0000b, OFS0.IWDTTOPS[1:0] = 11b

23.4 Output to the Event Link Controller (ELC)

The IWDT is capable of link operation for a specified module when the interrupt request signal is used as an event signal by the event link controller (ELC). The event signal is output by the counter underflow or refresh error.

An event signal is output regardless of the setting of the OFS0.IWDTRSTIRQS bit. An event signal can also be output at generation of the next interrupt source while the Refresh Error flag (IWDTSR.REFEF) or Underflow flag (IWDTSR.UNDFF) is 1. For details, see section 16, Event Link Controller (ELC).

23.5 Usage Notes

23.5.1 Refresh Operations

While configuring the refresh time, consider variations in the range of errors given the accuracy of PCLKB and IWDTCLK. Set values that ensure refreshing is possible.

23.5.2 Clock Division Ratio Setting

Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 4 × (the frequency of the count clock source after division).

23.5.3 Constraints on the ICU Event Link Setting Register n (IELSRn) Setting

Setting 0x03 to ICU Event Link Setting Register n (IELSRn.IELS[4:0]) is prohibited when enabling the IWDT reset assertion (OFS0.IWDTRSTIRQS = 0) or when enabling event link operation (ELSRn.ELS[7:0] = 0x17).

24. Serial Communications Interface (SCI)

24.1 Overview

The Serial Communications Interface (SCI) × 4 channels have asynchronous and synchronous serial interfaces:

- Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA))
- 8-bit clock synchronous interface
- Simple IIC (master-only)
- Simple SPI
- Smart card interface

The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. SCIn (n = 0) has FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using an on-chip baud rate generator.

In this section, PCLK refers to PCLKB.

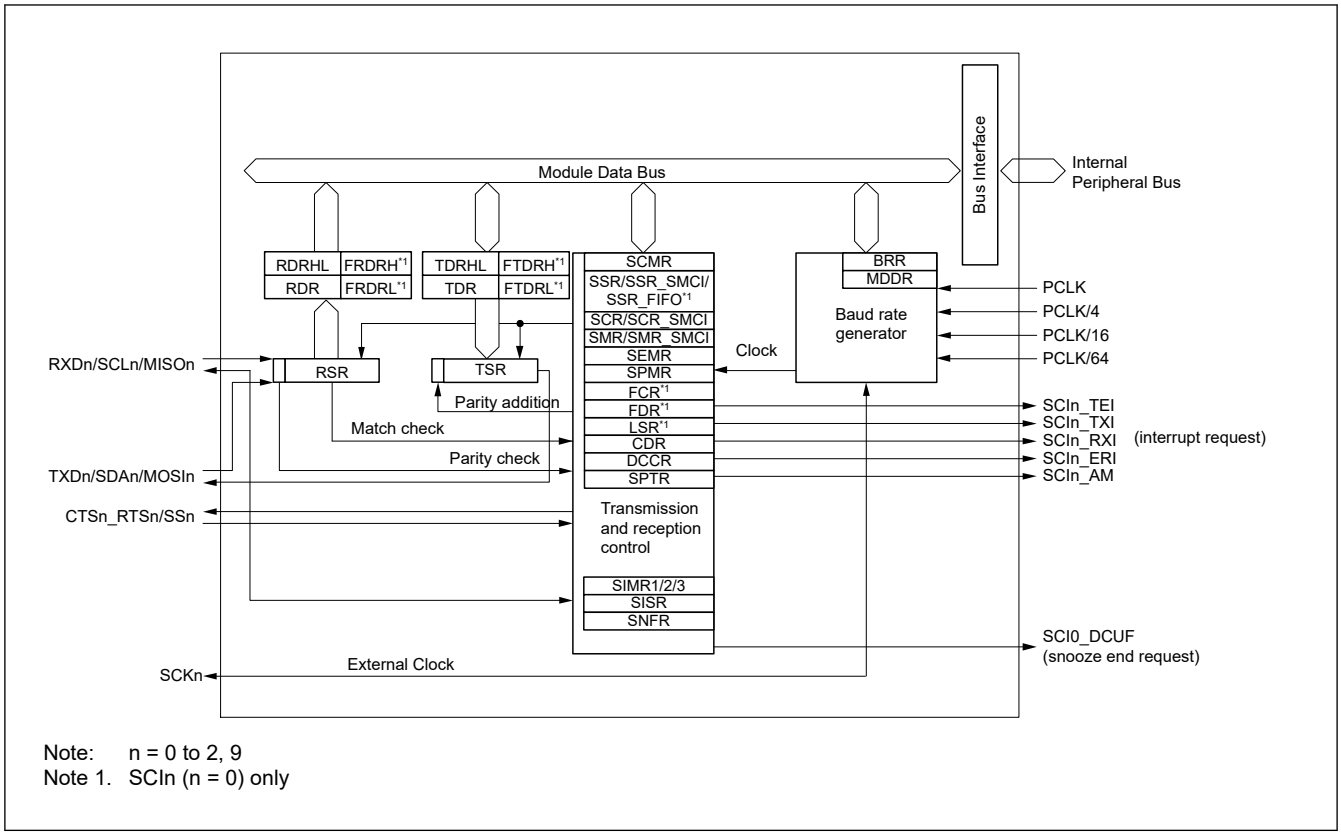
Table 24.1 lists the SCI specifications, Figure 24.1 shows a block diagram of SCI, and Table 24.3 lists the I/O pins.

Table 24.1 SCI specifications (1 of 2)

Parameter		Specifications
Number of modules		4 (SCIn (n = 0 to 2, 9))
Serial communication modes		• Asynchronous • Clock synchronous • Simple IIC • Simple SPI • Smart card interface
Transfer speed		Bit rate specifiable with the on-chip baud rate generator
Full-duplex communications		• Transmitter: Continuous transmission possible using double-buffering • Receiver: Continuous reception possible using double-buffering
Data transfer		Selectable as LSB-first or MSB-first transfer
Interrupt sources		Transmit end, transmit data empty, receive data full, receive error, receive data ready, address match. Completion of generation of a start condition, restart condition, or stop condition. (for simple IIC mode)
Module-stop function		Module-stop state can be set for each channel
Snooze end request		SCI0 address mismatch (SCI0_DCUF)
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pins
	Transmission and reception	Selectable to 1-stage register or 16-stage FIFO (only SCIn (n = 0) supports FIFO)
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	• Parity error • Overrun error • Framing error
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pins
	Transmission and reception	Selectable to 1-stage register or 16-stage FIFO (SCIn (n = 0) supports FIFO)

Table 24.1 SCI specifications (2 of 2)

Parameter		Specifications
	Address match	Interrupt request/event output can be issued upon detecting a match between received data and the value in the compare match register
	Address mismatch (SCI0 only) receive data	Snooze end request can be issued when detecting a mismatch between the received data and the value in the compare match register
	Start-bit detection	Selectable to low level or falling edge detection
	Break detection	Breaks from framing errors detectable by read from SPTR register
	Clock source	Selectable to internal or external clock
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Multi-processor communications function	Serial communication enabled among multiple processors
	Noise cancellation	Digital noise filters included on signal paths from the RXDn pin inputs
Smart card interface mode	Error processing	Error signal can be automatically transmitted upon detecting a parity error during reception
		Data can be automatically retransmitted upon receiving an error signal during transmission
	Data type	Both direct and inverse convention supported
Simple IIC mode	Transfer format	I^2C bus format (MSB-first only)
	Operating mode	Master (single-master operation only)
	Transfer rate	Up to 400 kbps
	Noise cancellation	The signal paths from input on the SCLn and SDAn pins incorporate digital noise filters and provide an adjustable interval for noise cancellation
Simple SPI mode	Data length	8 bits
	Error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	SSn input pin function	High impedance state can be invoked on the output pins by driving the SSn pin high.
	Clock settings	Configurable among four clock phase and clock polarity settings
Bit rate modulation function		Error reduction through correction of outputs from the on-chip baud rate generator
Event link function		Error event output for receive error or error signal detection (SCIn_ERI) (n = 0 to 2, 9)
		Receive data full event output (SCIn_RXI) (n = 0 to 2, 9)
		Transmit data empty event output (SCIn_TXI) (n = 0 to 2, 9)
		Address match event output (SCIn_AM) (n = 0 to 2, 9)
		Transmit end event output (SCIn_TEI) (n = 0 to 2, 9)



Note: n = 0 to 2, 9
Note 1. SCIn (n = 0) only

Figure 24.1 SCI block diagram

Table 24.2 Product and support channels

SCI channel	Product	
	48 pin	32 pin / 24 pin
SCI0	✓	✓
SCI1	✓	✓
SCI2	✓	✓*1
SCI9	✓	✓*1

Note: ✓ : Support
—: No support
Note 1. SCI2/9 only support Asynchronous mode and simple I^2C

Table 24.3 SCI I/O pins

Function	Pin name	Input/ Output	Description
SCIn (n = 0 to 2, 9)	SCKn	Input/ Output	SCIn clock input/output
	RXDn/SCLn/MISOn	Input/ Output	SCIn receive data input SCIn I^2C clock input/output SCIn slave transmit data input/output
	TXDn/SDAn/MOSIn	Input/ Output	SCIn transmit data output SCIn I^2C data input/output SCIn master transmit data input/output
	SSn/CTSn_RTSn	Input/ Output	SCIn chip select input, active-low SCIn transfer start control input/output, active-low

24.2 Register Descriptions

24.2.1 RSR : Receive Shift Register

RSR is a shift register that receives serial data input from the RXDn pin and converts it into parallel data. When one frame of data is received, the data is automatically transferred to the RDR, RDRHL, or the receive FIFO register. The RSR register cannot be directly accessed by the CPU.

24.2.2 RDR : Receive Data Register

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x05

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

RDR is an 8-bit register that stores received data. When one frame of serial data is received, it is transferred from RSR to RDR, and the RSR register can receive more data. Because RSR and RDR function as a double buffer, continuous received operations can be performed.

Read the RDR only once after a receive data full interrupt (SCIn_RXI) occurs.

Note: If the next frame of data is received before reading the received data from RDR, an overrun error occurs. The CPU cannot write to the RDR.

24.2.3 RDRHL : Receive Data Register

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x10

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

Bit	Symbol	Function	R/W
8:0	RDAT[8:0]	Serial Receive Data	R
15:9	—	These bits are read as 0.	R

RDRHL is a 16-bit register that stores received data. Use this register when asynchronous mode and 9-bit data length are selected.

The lower 8 bits of RDRHL are the shadow register of RDR, so access to RDRHL affects the RDR register. Access to the RDRHL register is prohibited if 7-bit or 8-bit data length is selected.

After one frame of data is received, the received data is transferred from the RSR register to the RDR/RDRHL registers, allowing the RSR register to receive more data.

The RSR and RDRHL registers form a double-buffered structure to enable continuous reception. RDRHL should be read only when a receive data full interrupt (SCIn_RXI) request is issued. An overrun error occurs when the next frame of data is received before the received data is read from RDRHL. The CPU cannot write to the RDRHL register.

24.2.4 FRDRHL/FRDRH/FRDRL : Receive FIFO Data Register

Base address: SCI0 = 0x4007_0000

Offset address: 0x10 (FRDRHL/FRDRH)
0x11 (FRDRL)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	RDF	ORER	FER	PER	DR	MPB	RDAT[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	RDAT[8:0]	Serial receive data Stores the serial receive data. Valid only in asynchronous mode, including multi-processor mode, and clock synchronous mode, and with FIFO selected.	R
9	MPB	Multi-Processor Bit Flag Stores the value of the multi-processor bit in the serial receive data, RDAT[8:0]. Valid only in asynchronous mode with SMR.MP = 1, and with FIFO selected. 0: Data transmission cycle 1: ID transmission cycle	R
10	DR	Receive Data Ready Flag This flag is the same as SSR_FIFO.DR. 0: Receiving is in progress, or no received data remains in the FRDRH and FRDRL registers after successfully completed reception 1: Next receive data is not received for a period after successfully completed reception	R*1
11	PER	Parity Error Flag 0: No parity error occurred in the first data of FRDRH and FRDRL 1: Parity error occurred in the first data of FRDRH and FRDRL	R
12	FER	Framing Error Flag 0: No framing error occurred in the first data of FRDRH and FRDRL 1: Framing error occurred in the first data of FRDRH and FRDRL	R
13	ORER	Overrun Error Flag This flag is the same as SSR_FIFO.ORER. 0: No overrun error occurred 1: Overrun error occurred	R*1
14	RDF	Receive FIFO Data Full Flag This flag is the same as SSR_FIFO.RDF. 0: The amount of receive data written in FRDRH and FRDRL is less than the specified receive triggering number 1: The amount of receive data written in FRDRH and FRDRL is equal to or greater than the specified receive triggering number	R*1
15	—	This bit is read as 0.	R

Note 1. If this flag is read, it indicates the same value as that read from the SSR_FIFO register. Write 0 to the SSR_FIFO register to clear the flag.

FRDRHL is a 16-bit register that consists of the 8-bit FRDRH and FRDRL registers. FRDRH is assigned to the FRDRHL[15:8] bits, and allocated to the same address as FRDRHL. FRDRL is assigned to the FRDRHL[7:0] bits, and allocated to (the address of FRDRHL + 1) address.

FRDRH and FRDRL constitute a 16-stage FIFO register that stores serial receive data and related status information readable by software. This register is only valid in asynchronous mode, including multi-processor mode, or clock synchronous mode.

The SCI completes reception of one frame of serial data by transferring the received data from the Receive Shift Register (RSR) into FRDRH and FRDRL for storage. Continuous reception is executed until 16 stages are stored. If data is read when there is no received data in FRDRH and FRDRL, the value is undefined. When FRDRH and FRDRL are full, subsequent serial receive data is lost. The CPU can read from the FRDRH and FRDRL registers but cannot write to them.

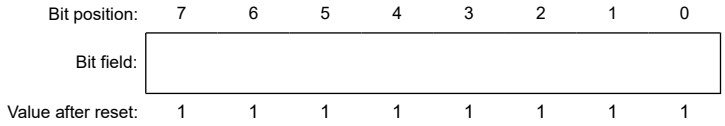
Reading 1 from the RDF, ORER, or DR flags of the FRDRH register is the same as reading from those bits in the SSR_FIFO register. When writing 0 to clear a flag in the SSR_FIFO register after reading the FRDRH register, write 0 only to the flag that is to be cleared and write 1 to the other flags.

When reading both the FRDRH and FRDRL registers, read in order from FRDRH to FRDRL. The FRDRHL register can be accessed in 16-bit units.

24.2.5 TDR : Transmit Data Register

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x03



Bit	Symbol	Function	R/W
7:0	n/a	Serial Transmit Data	R/W

TDR is an 8-bit register that stores transmit data.

When the SCI detects that the TSR register is empty, it transfers the transmit data written in the TDR register to the TSR register and starts transmission.

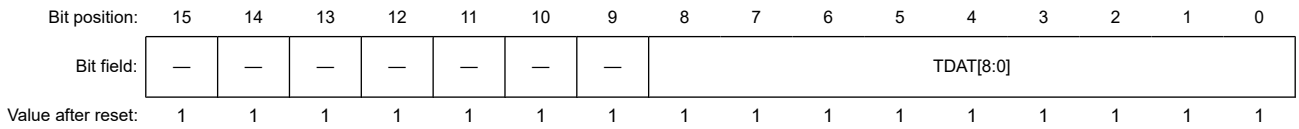
The double-buffered structure of the TDR and TSR registers enables continuous serial transmission. If the next transmit data is already written to TDR when one frame of data is transmitted, the SCI transfers the written data to the TSR register to continue transmission.

The CPU can read from or write to TDR at any time. Only write transmit data to TDR once after each instance of the transmit data empty interrupt (SCIn_TXI).

24.2.6 TDRHL : Transmit Data Register

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x0E



Bit	Symbol	Function	R/W
8:0	TDAT[8:0]	Serial Transmit Data	R/W
15:9	—	This bit is read as 1. The write value should be 1.	R/W

TDRHL is a 16-bit register that stores transmit data. Use this register when asynchronous mode and 9-bit data length are selected.

The lower 8 bits of TDRHL are the shadow register of TDR, so access to TDRHL affects the TDR register. Access to the TDRHL register is prohibited if 7-bit or 8-bit data length is selected.

When empty space is detected in the TSR register, the transmit data stored in the TDRHL registers is transferred to TSR and transmission starts.

The TSR and TDRHL registers have a double-buffered structure to support continuous transmission. When the next data to be transmitted is stored in TDRHL after one frame of data is transmitted, the transmitting operation continues by transferring the data to the TSR register.

The CPU can read and write to the TDRHL register. Bits [15:9] in TDRHL are fixed to 1. These bits are read as 1. The write value should be 1.

Write transmit data to theTDRHL register only once when a transmit data empty interrupt (SCIn_TXI) request is issued.

24.2.7 FTDRHL/FTDRH/FTDRL : Transmit FIFO Data Register

Base address: SCI0 = 0x4007_0000

Offset address: 0x0E (FTDRHL/FTDRH)
0x0F (FTDRL)

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

Bit	Symbol	Function	R/W
8:0	TDAT[8:0]	Serial transmit data Specifies the serial transmit data. Valid only in asynchronous mode, including multi-processor mode, and clock synchronous mode, and with FIFO selected.	W
9	MPBT	Multi-Processor Transfer Bit Flag Specifies the multi-processor bit in the transmission frame. Valid only in asynchronous mode and SMR.MP = 1, and with FIFO selected. Valid only in asynchronous mode, including multi-processor mode, and clock synchronous mode, and with FIFO selected. 0: Data transmission cycle 1: ID transmission cycle	W
15:10	—	The write value should be 1.	W

FTDRHL is a 16-bit register that consists of the 8-bit FTDRH and FTDRL registers. FTDRH is assigned to the FTDRHL[15:8] bits, and allocated to the same address as FTDRHL. FTDRL is assigned to the FTDRHL[7:0] bits, and allocated to (the address of FTDRHL + 1) address.

FTDRH and FTDRL constitute a 16-stage FIFO register that stores data for serial transmission and a multi-processor transfer bit. This register is only valid in asynchronous mode, including multi-processor mode, or clock synchronous mode.

When the SCI detects that the Transmit Shift Register (TSR) is empty, it transfers data written in the FTDRH and FTDRL registers to the TSR register and starts serial transmission. Continuous serial transmission is executed until no transmit data is left in FTDRH and FTDRL. When FTDRHL is full of transmit data, no more data can be written. If writing new data is attempted, the data is ignored. The CPU can write to the FTDRH and FTDRL registers but cannot read them.

When writing to both the FTDRH and FTDRL registers, write in order from FTDRH to FTDRL.

TDAT[8:0] bits (Serial transmit data)

The TDAT[8:0] bits set the serial transmission data. This is valid only when FIFO is selected in asynchronous mode (including multiprocessor) or clock synchronous mode.

MPBT flag (Multi-Processor Transfer Bit Flag)

The MPBT flag specifies the value of the multi-processor bit of the transmit frame. When FCR.FM = 1, SSR.MPBT is invalid.

24.2.8 TSR : Transmit Shift Register

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI first automatically transfers transmit data from TDR, TDRHL, or transmit FIFO to TSR, then sends the data to the TXDn pin. The CPU cannot directly access the TSR.

24.2.9 SMR : Serial Mode Register for Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CKS[1:0]	Clock Select 0 0: PCLK clock (n = 0)*1 0 1: PCLK/4 clock (n = 1)*1 1 0: PCLK/16 clock (n = 2)*1 1 1: PCLK/64 clock (n = 3)*1	R/W*4
2	MP	Multi-Processor Mode Valid only in asynchronous mode. 0: Disable multi-processor communications function 1: Enable multi-processor communications function	R/W*4
3	STOP	Stop Bit Length Valid only in asynchronous mode. 0: 1 stop bit 1: 2 stop bits	R/W*4
4	PM	Parity Mode Valid only when the PE bit is 1. 0: Even parity 1: Odd parity	R/W*4
5	PE	Parity Enable Valid only in asynchronous mode. 0: When transmitting: Do not add parity bit When receiving: Do not check parity bit 1: When transmitting: Add parity bit When receiving: Check parity bit	R/W*4
6	CHR	Character Length Valid only in asynchronous mode.*2 Selects the transmit/receive character length in combination with the SCMR.CHR1 bit. 0: SCMR.CHR1 = 0: Transmit/receive in 9-bit data length SCMR.CHR1 = 1: Transmit/receive in 8-bit data length (initial value) 1: SCMR.CHR1 = 0: Transmit/receive in 9-bit data length SCMR.CHR1 = 1: Transmit/receive in 7-bit data length*3	R/W*4
7	CM	Communication Mode 0: Asynchronous mode or simple IIC mode 1: Clock synchronous mode or simple SPI mode	R/W*4

Note 1. n is the decimal notation of the value of n in the BRR register. See section 24.2.17. BRR : Bit Rate Register.
Note 2. In any mode other than asynchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.
Note 3. LSB-first is fixed and the MSB (bit [7]) in the TDR register is not transmitted in transmit mode.
Note 4. Writable only when SCR.TE = 0 and SCR.RE = 0 (both serial transmission and reception are disabled).

The SMR register sets the communication format and clock source for the on-chip baud rate generator.

CKS[1:0] bits (Clock Select)

The CKS[1:0] bits select the clock source for the on-chip baud rate generator. For the relationship between the settings of these bits and the baud rate, see section 24.2.17. BRR : Bit Rate Register.

MP bit (Multi-Processor Mode)

The MP bit disables or enables the multi-processor communications function. The PE and PM bit settings are invalid in multi-processor mode.

STOP bit (Stop Bit Length)

The STOP bit selects the stop bit length in transmission.

In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

PM bit (Parity Mode)

The PM bit selects the parity mode (even or odd) for transmission and reception. The PM bit setting is invalid in multiprocessor mode.

PE bit (Parity Enable)

When the PE bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception. Regardless of the PE bit setting, the parity bit is not added or checked in multi-processor format.

CHR bit (Character Length)

The CHR bit selects the data length for transmission and reception in combination with the SCMR.CHR1 bit. In modes other than asynchronous, a fixed data length of 8 bits is used.

CM bit (Communication Mode)

The CM bit selects the communication mode:

- Asynchronous mode or simple IIC mode
- Clock synchronous mode or simple SPI mode

24.2.10 SMR_SMCI : Serial Mode Register for Smart Card Interface Mode (SCMR.SMIF = 1)

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CKS[1:0]	Clock Select 0 0: PCLK clock (n = 0)*1 0 1: PCLK/4 clock (n = 1)*1 1 0: PCLK/16 clock (n = 2)*1 1 1: PCLK/64 clock (n = 3)*1	R/W*2
3:2	BCP[1:0]	Base Clock Pulse Selects the number of base clock cycles in combination with the SCMR.BCP2 bit. Table 24.4 lists the combinations of the SCMR.BCP2 and SMR.BCP[1:0] bits.	R/W*2
4	PM	Parity Mode Valid only when the PE bit is 1. 0: Even parity 1: Odd parity	R/W*2
5	PE	Parity Enable When this bit is set to 1, a parity bit is added to transmit data, and the parity of received data is checked. Set this bit to 1 in smart card interface mode.	R/W*2
6	BLK	Block Transfer Mode 0: Normal mode operation 1: Block transfer mode operation	R/W*2

Bit	Symbol	Function	R/W
7	GM	GSM Mode 0: Normal mode operation 1: GSM mode operation	R/W*2

Note 1. n is the decimal notation of the value of n in the BRR register. See section 24.2.17. BRR : Bit Rate Register.
Note 2. Writable only when SCR_SMCI.TE = 0 and SCR_SMCI.RE = 0 (both serial transmission and reception are disabled).

The SMR_SMCI register sets the communication format and clock source for the on-chip baud rate generator.

CKS[1:0] bits (Clock Select)

The CKS[1:0] bits select the clock source for the on-chip baud rate generator. For the relationship between the settings of these bits and the baud rate, see section 24.2.17. BRR : Bit Rate Register.

BCP[1:0] bits (Base Clock Pulse)

The BCP[1:0] bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set these bits in combination with the SCMR.BCP2 bit.

For details, see section 24.6.4. Receive Data Sampling Timing and Reception Margin.

Table 24.4 Combinations of SCMR.BCP2 and SMR_SMCI.BCP[1:0] bits

SCMR.BCP2 bit	SMR_SMCI.BCP[1:0] bits	Number of base clock cycles for 1-bit transfer period*1
0	00b	93 clock cycles (S = 93)
0	01b	128 clock cycles (S = 128)
0	10b	186 clock cycles (S = 186)
0	11b	512 clock cycles (S = 512)
1	00b	32 clock cycles (S = 32) (initial value)
1	01b	64 clock cycles (S = 64)
1	10b	372 clock cycles (S = 372)
1	11b	256 clock cycles (S = 256)

Note 1. S is the value of S in BRR (see section 24.2.17. BRR : Bit Rate Register).

PM bit (Parity Mode)

The PM bit selects the parity mode for transmission and reception (even or odd). For details on the usage of this bit in smart card interface mode, see section 24.6.2. Data Format (Except in Block Transfer Mode).

PE bit (Parity Enable)

Set the PE bit to 1. The parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

BLK bit (Block Transfer Mode)

Setting the BLK bit to 1 enables block transfer mode operation. For details, see section 24.6.3. Block Transfer Mode.

GM bit (GSM Mode)

Setting the GM bit to 1 enables GSM mode operation. In GSM mode, the SSR_SMCI.TEND flag set timing is moved forward to 11.0 etus (elementary time unit = 1-bit transfer time) from the start bit, and clock output control is added. For details, see section 24.6.6. Serial Data Transmission (Except in Block Transfer Mode) and section 24.6.8. Clock Output Control.

24.2.11 SCR : Serial Control Register for Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CKE[1:0]	Clock Enable 0 0: In asynchronous mode, the SCKn pin is available for use as an I/O port based on the I/O port settings. In clock synchronous mode, the SCKn pin functions as the clock output pin. 0 1: In asynchronous mode, a clock with the same frequency as the bit rate is output from the SCKn pin. In clock synchronous mode, the SCKn pin functions as the clock output pin. Others: In asynchronous mode, input a clock with a frequency 16 times the bit rate from the SCKn pin when the SEMR.ABCS bit is 0. Input a clock signal with a frequency eight times the bit rate when the SEMR.ABCS bit is 1. In clock synchronous mode, the SCKn pin functions as the clock input pin.	R/W*1
2	TEIE	Transmit End Interrupt Enable 0: Disable SCIn_TEI interrupt requests 1: Enable SCIn_TEI interrupt requests	R/W
3	MPIE	Multi-Processor Interrupt Enable Valid in asynchronous mode when SMR.MP = 1. 0: Normal reception 1: When data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags RDRF, ORER, and FER in SSR to 1. When data with the multi-processor bit set to 1 is received, the MPIE bit is automatically set to 0, and normal reception is resumed.	R/W*3
4	RE	Receive Enable 0: Disable serial reception 1: Enable serial reception	R/W*2
5	TE	Transmit Enable 0: Disable serial transmission 1: Enable serial transmission	R/W*2
6	RIE	Receive Interrupt Enable 0: Disable SCIn_RXI and SCIn_ERI interrupt requests 1: Enable SCIn_RXI and SCIn_ERI interrupt requests	R/W
7	TIE	Transmit Interrupt Enable 0: Disable SCIn_TXI interrupt requests 1: Enable SCIn_TXI interrupt requests	R/W

Note 1. Writable only when TE = 0 and RE = 0.
Note 2. 1 can be written only when TE = 0 and RE = 0, when the SMR.CM bit is 1. After setting TE or RE to 1, only 0 can be written to TE and RE. When the SMR.CM bit is 0 and the SIMR1.IICM bit is 0, writing is enabled under any condition.
Note 3. When writing a new value to a bit other than the MPIE bit of this register in multi-processor mode (SMR.MP bit = 1), write 0 to the MPIE bit using the store instruction to avoid accidentally setting the MPIE bit to 1 by a read-modify-write operation when using a bit manipulation instruction.

The SCR register controls operation and clock source selection for transmission and reception.

CKE[1:0] bits (Clock Enable)

The CKE[1:0] bits select the clock source and the SCKn pin function.

TEIE bit (Transmit End Interrupt Enable)

The TEIE bit enables or disables SCIn_TEI interrupt requests. Set TEIE to 0 to disable an SCIn_TEI interrupt request.

In simple IIC mode, SCIn_TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STIn). In this case, the TEIE bit can be used to enable or disable the STI.

MPIE bit (Multi-Processor Interrupt Enable)

When the MPIE bit is set to 1 and data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags RDRF, ORER, FER, RDF, and DR in SSR/SSR_FIFO to 1 is disabled. When data with the multi-processor bit set to 1 is received, the MPIE bit is automatically set to 0, and normal reception resumes. For details, see section 24.4. Multi-Processor Communication Function.

When the MPB bit in the SSR register is 0, the receive data is not transferred from the RSR register to the RDR register, a receive error is not detected, and setting the flags ORER and FER to 1 is disabled.

When the MPB bit is set to 1, the MPIE bit is automatically set to 0, SCIn_RXI and SCIn_ERI interrupt requests are enabled (if the RIE bit in SCR is set to 1), and setting of the ORER and FER flags to 1 is enabled.

Set MPIE to 0 if the multi-processor communications function is not used.

RE bit (Receive Enable)

The RE bit enables or disables serial reception. When the RE bit is set to 1, serial reception starts by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Set the reception format in the SMR register before setting the RE bit to 1.

In non-FIFO operation, when reception is halted by setting the RE bit to 0, the RDRF, ORER, FER, and PER flags in the SSR register are not affected, and the previous values are retained.

When FIFO operation is selected and reception is halted by setting the RE bit to 0, the RDF, ORER, FER, PER, and DR flags in SSR_FIFO are not affected and the previous values are retained.

TE bit (Transmit Enable)

The TE bit enables or disables serial transmission.

When the TE bit is set to 1, serial transmission is started by writing transmit data to the TDR register. Set the transmission format in the SMR register before setting the TE bit to 1.

RIE bit (Receive Interrupt Enable)

The RIE bit enables or disables SCIn_RXI and SCIn_ERI interrupt requests.

SCIn_RXI and SCIn_ERI interrupt requests are disabled by setting the RIE bit to 0.

An SCIn_ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in SSR/SSR_FIFO then setting the flag to 0, or by setting the RIE bit to 0.

TIE bit (Transmit Interrupt Enable)

The TIE bit enables or disables SCIn_TXI interrupt requests. SCIn_TXI interrupt requests are disabled by setting the TIE bit to 0.

Note: To switch the TIE bit value from 0 to 1 in FIFO mode, set the TIE and TE bits to 1 simultaneously or set the TIE bit to 1 when TE = 1. When TE = 0 in FIFO mode, setting the TIE bit to 1 is prohibited.

24.2.12 SCR_SMCI : Serial Control Register for Smart Card Interface Mode (SCMR.SMIF = 1)

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CKE[1:0]	Clock Enable 0 0: When SMR_SMCI.GM = 0: Disable output The SCKn pin is available for use as an I/O port if set up in the I/O port settings When SMR_SMCI.GM = 1: Fix output low 0 1: When SMR_SMCI.GM = 0: Output clock When SMR_SMCI.GM = 1: Output clock 1 0: When SMR_SMCI.GM = 0: Setting prohibited When SMR_SMCI.GM = 1: Fix output high 1 1: When SMR_SMCI.GM = 0: Setting prohibited When SMR_SMCI.GM = 1: Output clock	R/W*1
2	TEIE	Transmit End Interrupt Enable Set this bit to 0 in smart card interface mode	R/W
3	MPIE	Multi-Processor Interrupt Enable Set this bit to 0 in smart card interface mode	R/W
4	RE	Receive Enable 0: Disable serial reception 1: Enable serial reception	R/W*2
5	TE	Transmit Enable 0: Disable serial transmission 1: Enable serial transmission	R/W*2
6	RIE	Receive Interrupt Enable 0: Disable SCIn_RXI and SCIn_ERI interrupt requests 1: Enable SCIn_RXI and SCIn_ERI interrupt requests	R/W
7	TIE	Transmit Interrupt Enable 0: Disable SCIn_TXI interrupt requests 1: Enable SCIn_TXI interrupt requests	R/W

Note 1. Writable only when TE = 0 and RE = 0.
Note 2. 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written to TE and RE.

The SCR_SMCI register sets transmission and reception control, interrupt control, and clock source selection for transmission and reception.

For details on interrupt requests, see section 24.10. Interrupt Sources.

CKE[1:0] bits (Clock Enable)

The CKE[1:0] bits control the clock output from the SCKn pin. In GSM mode, clock output can be dynamically switched. For details, see section 24.6.8. Clock Output Control.

TEIE bit (Transmit End Interrupt Enable)

Set the TEIE bit to 0 in smart card interface mode.

MPIE bit (Multi-Processor Interrupt Enable)

Set the MPIE bit to 0 in smart card interface mode.

RE bit (Receive Enable)

The RE bit enables or disables serial reception. When the RE bit is set to 1, serial reception starts by detecting the start bit. Set the reception format in the SMR_SMCI register before setting the RE bit to 1.

If reception is halted by setting the RE bit to 0, the ORER, FER, and PER flags in SSR_SMCI are not affected and the previous values are retained.

TE bit (Transmit Enable)

The TE bit enables or disables serial transmission. When the TE bit is set to 1, serial transmission is started by writing transmit data to TDR. Set the transmission format in the SMR_SMCI register before setting the TE bit to 1.

RIE bit (Receive Interrupt Enable)

The RIE bit enables or disables SCIn_RXI and SCIn_ERI interrupt requests.

SCIn_RXI and SCIn_ERI interrupt requests are disabled by setting the RIE bit to 0.

An SCIn_ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR_SMCI register, and then setting the flag to 0, or by setting the RIE bit to 0.

TIE bit (Transmit Interrupt Enable)

The TIE bit enables or disables SCIn_TXI interrupt requests. SCIn_TXI interrupt requests are disabled by setting the TIE bit to 0.

24.2.13 SSR : Serial Status Register for Non-Smart Card Interface and Non-FIFO Mode (SCMR.SMIF = 0, FCR.FM = 0)

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
0	MPBT	Multi-Processor Bit Transfer Sets the value of the multi-processor bit in the transmission frame. 0: Data transmission cycle 1: ID transmission cycle	R/W
1	MPB	Multi-Processor Value of the multi-processor bit in the reception frame. 0: Data transmission cycle 1: ID transmission cycle	R
2	TEND	Transmit End Flag 0: A character is being transmitted 1: Character transfer is complete	R
3	PER	Parity Error Flag 0: No parity error occurred 1: Parity error occurred	R/(W)*1
4	FER	Framing Error Flag 0: No framing error occurred 1: Framing error occurred	R/(W)*1
5	ORER	Overrun Error Flag 0: No overrun error occurred 1: Overrun error occurred	R/(W)*1
6	RDRF	Receive Data Full Flag 0: No received data in RDR register 1: Received data in RDR register	R/(W)*1
7	TDRE	Transmit Data Empty Flag 0: Transmit data in TDR register 1: No transmit data in TDR register	R/(W)*1

Note 1. Only 0 can be written to clear the flag after reading 1.

The SSR register provides SCI status flags and transmission and reception multi-processor bits.

MPBT bit (Multi-Processor Bit Transfer)

The MPBT bit sets the value of the multi-processor bit in the transmit frame.

MPB bit (Multi-Processor)

The MPB bit holds the value of the multi-processor bit in the reception frame. This bit does not change when the SCR.RE bit is 0.

TEND flag (Transmit End Flag)

The TEND flag indicates completion of transmission.

[Setting conditions]

- When the SCR.TE bit is set to 0 (serial transmission is disabled) and the FCR.FM bit is set to 0 (non-FIFO selected). When the SCR.TE bit is set to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated on transmission of the tail-end bit of a character being transmitted.

[Clearing conditions]

- When transmit data is written to the TDR register while the SCR.TE bit is 1
- When 0 is written to TDRE after reading TDRE = 1 while the SCR.TE bit is 1

PER flag (Parity Error Flag)

The PER flag indicates that a parity error occurred during reception in asynchronous mode and the reception ended abnormally.

[Setting condition]

- When a parity error is detected during reception in asynchronous mode when the address match function is disabled (DCCR.DCME = 0).

  Although receive data is transferred to the RDR register when the parity error occurs, no SCIn_RXI interrupt request occurs. When the PER flag is set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to PER after reading PER = 1. After writing 0 to the PER flag, read the PER flag to check that it is actually set to 0.

When the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

FER flag (Framing Error Flag)

The FER flag indicates that a framing error occurred during reception in asynchronous mode and the reception ended abnormally.

[Setting condition]

- When 0 is sampled as the stop bit during reception in asynchronous mode when the address match function is disabled (DCCR.DCME = 0).

  In 2-stop-bit mode, only the first stop bit is checked. The second stop bit is not checked. Although receive data is transferred to the RDR register when the framing error occurs, no SCIn_RXI interrupt request occurs. When the FER flag is to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to FER after reading FER = 1. After writing 0 to the FER flag, read the FER flag to check that it is actually set to 0.

When the SCR.RE bit is set to 0 (serial reception is disabled), the FER flag is not affected and retains its previous value.

ORER flag (Overrun Error Flag)

The ORER flag indicates that an overrun error occurred during reception and the reception ended abnormally.

[Setting condition]

- When the next data is received before receive data that does not have a parity error and a framing error is read from the RDR register.

  The data received before an overrun error occurred is saved in the RDR register, but data received after the error is lost. When the ORER flag is set to 1, receive data is not forwarded to the RDR register. In clock synchronous mode, serial transmission and reception are stopped.

[Clearing condition]

- When 0 is written to ORER after reading ORER = 1. After writing 0 to the ORER flag, read the ORER flag to check that it is actually set to 0.

When the SCR.RE bit is set to 0 (serial reception is disabled), the ORER flag is not affected and retains its previous value.

RDRF flag (Receive Data Full Flag)

The RDRF flag indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing conditions]

- When 0 is written to RDRF after reading RDRF = 1
- When data is forwarded from the RDR register

TDRE flag (Transmit Data Empty Flag)

The TDRE flag indicates the presence of transmit data in the TDR register.

[Setting conditions]

- When the SCR.TE bit is 0
- When data is transmitted from the TDR register to the TSR register

[Clearing conditions]

- When 0 is written to TDRE after reading TDRE = 1
- When the SCR.TE bit is 1 and data is written to the TDR register

24.2.14 SSR_FIFO : Serial Status Register for Non-Smart Card Interface and FIFO Mode (SCMR.SMIF = 0, FCR.FM = 1)

Base address: SCI0 = 0x4007_0000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TDFE	RDF	ORER	FER	PER	TEND	—	DR
Value after reset:	1	0	0	0	0	0	x	0

Bit	Symbol	Function	R/W
0	DR	Receive Data Ready Flag 0: Receiving is in progress, or no received data remains in FRDRHL after successfully completed reception (receive FIFO empty) 1: Next receive data is not received for a period after normal receiving is complete, when the amount of data stored in the FIFO is equal to or less than the receive triggering number	R/W*1
1	—	The read value is undefined. The write value should be 1.	R/W
2	TEND	Transmit End Flag 0: A character is being transmitted 1: Character transfer is complete	R/W*1
3	PER	Parity Error Flag 0: No parity error occurred 1: Parity error occurred	R/W*1
4	FER	Framing Error Flag 0: No framing error occurred 1: Framing error occurred	R/W*1

Bit	Symbol	Function	R/W
5	ORER	Overrun Error Flag 0: No overrun error occurred 1: Overrun error occurred	R/W*1
6	RDF	Receive FIFO Data Full Flag 0: The amount of receive data written in FRDRHL is less than the specified receive triggering number 1: The amount of receive data written in FRDRHL is equal to or greater than the specified receive triggering number	R/W*1
7	TDFE	Transmit FIFO Data Empty Flag 0: The amount of transmit data written in FTDRHL exceeds the specified transmit triggering number 1: The amount of transmit data written in FTDRHL is equal to or less than the specified transmit triggering number	R/W*1

Note 1. Only 0 can be written, to clear the flag after reading 1.

The SSR_FIFO register provides the SCI with FIFO mode status flags.

DR flag (Receive Data Ready Flag)

The DR flag indicates that the amount of data stored in the Receive FIFO Data Register (FRDRHL) falls below the specified receive triggering number, and that no next data is received after 15 etus (elementary time units) from the last stop bit in asynchronous mode. This flag is valid only in asynchronous mode, including multi-processor mode, when FIFO operation is selected.

In clock synchronous mode, the DR flag is not set to 1.

[Setting condition]

- When FRDRHL contains less data than the specified receive triggering number, and no next data is received after 15 etus*1 from the last stop bit, and the SSR_FIFO.FER and SSR_FIFO.PER flags are 0.

[Clearing conditions]

- When 1 is read from DR, after all received data is read
- When the FCR.FM bit is changed from 0 to 1

Note 1. This is equivalent to 1.5 frames in the 8-bit format with one stop bit.

The DR flag is only set to 1 when FIFO is selected in asynchronous mode, including multi-processor mode. It is not set to 1 in other operation modes.

TEND flag (Transmit End Flag)

The TEND flag indicates that FTDRHL does not contain valid data when transmitting the last bit of a serial character, so the transmission is halted.

[Setting condition]

- When FTDRHL does not contain transmit data when the last bit of a 1-byte serial character is transmitted.

[Clearing conditions]

- When transmit data is written to FTDRHL*1 while the SCR.TE bit is 1
- When 0 is written to TEND after 1 is read from TEND, when the SCR.TE bit is 1
- When the FCR.FM bit is changed from 0 to 1

Note 1. Do not use the TEND bit as a transmit end flag when the DTC writes data to FTDRHL in response to an SCIn_TXI interrupt request.

PER flag (Parity Error Flag)

The PER flag indicates whether there is a parity error in the data read from the FRDRHL register in asynchronous mode when the address match function is disabled (DCCR.DCME = 0).

[Setting condition]

- When data is received and a parity error is detected, when the address match function is disabled (DCCR.DCME = 0).

[Clearing condition]

- When 0 is written to PER after reading PER = 1.

The reception operation is continuous, and the receive data is stored in the FRDRHL register, even when a parity error occurs during reception.

When the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

FER flag (Framing Error Flag)

The FER flag indicates whether there is a framing error in the data read from the FRDRHL register in asynchronous mode when the address match function is disabled (DCCR.DCME = 0).

[Setting condition]

- When 0 is sampled as the stop bit during reception when the address match function is disabled (DCCR.DCME = 0).

[Clearing condition]

- When 0 is written to FER after reading FER = 1.

The reception operation is continuous, and the receive data is stored in the FRDRHL register, even when a framing error occurs during reception.

When the SCR.RE bit is set to 0 (serial reception is disabled), the FER flag is not affected and retains its previous value.

ORER flag (Overrun Error Flag)

The ORER flag indicates that the receive operation stopped abnormally because an overrun error occurred.

[Setting condition]

- When the next serial reception completes while the receive FIFO is full with 16-byte receive data.

[Clearing condition]

- When 0 is written to ORER after reading ORER = 1.

When the SCR.RE bit is set to 0 (serial reception is disabled), the ORER flag is not affected and retains its previous value.

RDF flag (Receive FIFO Data Full Flag)

The RDF flag indicates that receive data was transferred to the FRDRHL register, and the amount of data in FRDRHL is equal to or exceeds the specified receive triggering number. When RTRG is set to 0, the RDF flag is not set even when the amount of data in the receive FIFO is equal to 0.

[Setting condition]

- When the amount of receive data equal to or greater than the specified receive triggering number is stored in FRDRHL,[*1] and the FIFO is not empty.

[Clearing conditions]

- When 0 is written to RDF after reading RDF = 1
- When FRDRHL is read by the DTC, but only when the block transfer is the last transmission
- When the setting and clearing conditions occur at the same time, the RDF bit is set to 0. After that, when the amount of data stored in the FRDRHL register is the same as or greater than the RTRG value, RDF is set to 1 after 1 PCLK.

Note 1. Because FRDRHL is a 16-stage FIFO register, the maximum amount of data that can be read when RDF is 1 is equivalent to the specified receive triggering number. If an attempt is made to read after all the data in FRDRHL is read, the data is undefined.

TDFE flag (Transmit FIFO Data Empty Flag)

The TDFE flag indicates that data is transferred from the FTDRHL register into the TSR register, the amount of data in FTDRHL is below the specified transmit triggering number, and writing of transmit data to FTDRHL is enabled.

[Setting conditions]

- When the TE bit in SCR is 0
- When the amount of transmit data written in FTDRHL is equal to or less than the specified transmit triggering number*1

[Clearing conditions]

- When writing to FTDRHL is executed on the last transmission while the DTC is activated
- When 0 is written to the TDFE flag after reading TDFE = 1.*2
  The setting conditions are given priority when TE = 0. When the setting condition and clearing condition occur at the same time, the TDFE flag is set to 0. After that, when the amount of data stored in the FTDRHL register is equal to or less than the TTRG value, TDFE is set to 1 after 1 PCLK.

Note 1. Because the FTDRHL register is a 16-stage FIFO register, when the TDFE flag is 1, the maximum amount of data that can be written to the FTDRHL register is 16 minus FDR.T[4:0] bytes. If more data is written, data is discarded.
Note 2. Do not clear the TDFE flag during block transfer processing by the DTC.

24.2.15 SSR_SMCI : Serial Status Register for Smart Card Interface Mode (SCMR.SMIF = 1)

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
0	MPBT	Multi-Processor Bit Transfer Set this bit to 0 in smart card interface mode	R/W
1	MPB	Multi-Processor Set this bit to 0 in smart card interface mode	R
2	TEND	Transmit End Flag 0: A character is being transmitted 1: Character transfer is complete	R
3	PER	Parity Error Flag 0: No parity error occurred 1: Parity error occurred	R/W*1
4	ERS	Error Signal Status Flag 0: No low error signal response 1: Low error signal response occurred	R/W*1
5	ORER	Overrun Error Flag 0: No overrun error occurred 1: Overrun error occurred	R/W*1
6	RDRF	Receive Data Full Flag 0: No received data in RDR register 1: Received data in RDR register	R/W*1
7	TDRE	Transmit Data Empty Flag 0: Transmit data in TDR register 1: No transmit data in TDR register	R/W*1

Note 1. Only 0 can be written, to clear the flag after reading 1.

The SSR_SMCI register provides the SCI with smart card interface mode status flags.

TEND flag (Transmit End Flag)

When there is no error signal from the receiving side, the TEND flag is set to 1 when more data for transfer is ready to be transferred to the TDR register.

[Setting conditions]

- When the SCR_SMCI.TE bit = 0 (serial transmission is disabled).
  When the SCR_SMCI.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period elapses after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated.

  The set timing is determined by the following register settings:
  - When SMR_SMCI.GM = 0 and SMR_SMCI.BLK = 0, 12.5 etus after the start of transmission
  - When SMR_SMCI.GM = 0 and SMR_SMCI.BLK = 1, 11.5 etus after the start of transmission
  - When SMR_SMCI.GM = 1 and SMR_SMCI.BLK = 0, 11.0 etus after the start of transmission
  - When SMR_SMCI.GM = 1 and SMR_SMCI.BLK = 1, 11.0 etus after the start of transmission

[Clearing conditions]

- When transmit data is written to the TDR register while the SCR_SMCI.TE bit is 1
- When 0 is written to TDRE after reading TDRE = 1 while the SCR_SMCI.TE bit is 1

PER flag (Parity Error Flag)

The PER flag indicates that a parity error occurred during reception in asynchronous mode and the reception ended abnormally.

[Setting condition]

- When a parity error is detected during reception. Although receive data is transferred to RDR when a parity error occurs, no SCIn_RXI interrupt request occurs. After the PER flag is set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1. After writing 0 to the PER flag, read the flag to check that it is actually set to 0.

When the RE bit in SCR_SMCI is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

ERS flag (Error Signal Status Flag)

[Setting condition]

- When a low error signal is sampled.

[Clearing condition]

- When 0 is written to ERS after reading ERS = 1.

ORER flag (Overrun Error Flag)

The ORER flag indicates that an overrun error occurred during reception and the reception ended abnormally.

[Setting condition]

- When the next data is received before receive data that does not have a parity error is read from the RDR register. The data received before an overrun error occurred is saved in the RDR, but data received after the error is lost. When the ORER flag is set to 1, receive data is not forwarded to the RDR register.

[Clearing condition]

- When 0 is written to ORER after reading ORER = 1. After writing 0 to the ORER flag, read the flag to check that it is actually set to 0.

When the RE bit in SCR_SMCI is set to 0, the ORER flag is not affected and retains its previous value.

RDRF flag (Receive Data Full Flag)

The RDRF flag indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing conditions]

- When 0 is written to RDRF after reading RDRF = 1
- When data is forwarded from the RDR register

TDRE flag (Transmit Data Empty Flag)

The TDRE flag indicates the presence of transmit data in the TDR register.

[Setting conditions]

- When the SCR_SMCI.TE bit is 0
- When data is transmitted from the TDR register to the TSR register

[Clearing conditions]

- When 0 is written to TDRE after reading TDRE = 1
- When the SCR_SMCI.TE bit is 1 and data is written to the TDR register

24.2.16 SCMR : Smart Card Mode Register

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BCP2	—	—	CHR1	SDIR	SINV	—	SMIF
Value after reset:	1	1	1	1	0	0	1	0

Bit	Symbol	Function	R/W
0	SMIF	Smart Card Interface Mode Select 0: Non-smart card interface mode (asynchronous mode, clock synchronous mode, simple SPI mode, or simple IIC mode) 1: Smart card interface mode	R/W*1
1	—	This bit is read as 1. The write value should be 1.	R/W
2	SINV	Transmitted/Received Data Invert Set the SINV bit to 0 for operation in simple IIC mode. The SINV bit can be used in the following modes: ● Smart card interface mode ● Asynchronous mode (including multi-processor mode) ● Clock synchronous mode ● Simple SPI mode 0: TDR contents are transmitted as they are. Received data is stored as received in the RDR register. 1: TDR register contents are inverted before transmission. Receive data is stored in inverted form in the RDR register.	R/W*1

Bit	Symbol	Function	R/W
3	SDIR	Transmitted/Received Data Transfer Direction Set the SDIR bit to 1 for operation in simple IIC mode. The SDIR bit can be used in the following modes: ● Smart card interface mode ● Asynchronous mode (including multi-processor mode) ● Clock synchronous mode ● Simple SPI mode 0: Transfer LSB-first 1: Transfer MSB-first	R/W*1
4	CHR1	Character Length 1 Valid only in asynchronous mode.*2 Selects the transmit/receive character length in combination with the SMR.CHR bit. 0: SMR.CHR = 0: Transmit/receive in 9-bit data length SMR.CHR = 1: Transmit/receive in 9-bit data length 1: SMR.CHR = 0: Transmit/receive in 8-bit data length (initial value) SMR.CHR = 1: Transmit/receive in 7-bit data length*3	R/W*1
6:5	—	These bits are read as 1. The write value should be 1.	R/W
7	BCP2	Base Clock Pulse 2 Selects the number of base clock cycles in combination with the SMR_SMCI.BCP[1:0] bits. Table 24.5 lists the combinations of the SCMR.BCP2 and SMR_SMCI.BCP[1:0] bits.	R/W*1

Note 1. Writable only when the TE and RE bits in SCR/SCR_SMCI are 0 (both serial transmission and reception are disabled).
Note 2. The setting is invalid and a fixed data length of 8 bits is used in modes other than asynchronous mode.
Note 3. LSB-first must be selected and the value of the MSB (bit [7]) in TDR cannot be transmitted.

The SCMR register selects the smart card interface and communication format.

SMIF bit (Smart Card Interface Mode Select)

Setting the SMIF bit to 1 selects smart card interface mode. Setting it to 0 selects all other modes:

- Asynchronous mode, including multi-processor mode
- Clock synchronous mode
- Simple SPI mode
- Simple IIC mode

SINV bit (Transmitted/Received Data Invert)

The SINV bit inverts the transmit and receive data logic level. It does not affect the logic level of the parity bit. To invert the parity bit, invert the PM bit in SMR or SMR_SMCI.

CHR1 bit (Character Length 1)

The CHR1 bit selects the data length of transmit and receive data in combination with the CHR bit in the SMR register. A fixed data length of 8 bits is used in modes other than asynchronous mode.

BCP2 bit (Base Clock Pulse 2)

The BCP2 bit selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the SMR_SMCI.BCP[1:0] bits.

Table 24.5 Combinations of the SCMR.BCP2 and SMR_SMCI.BCP[1:0] bits (1 of 2)

SCMR.BCP2 bit	SMR_SMCI.BCP[1:0] bits	Number of base clock cycles for 1-bit transfer period
0	00b	93 clock cycles (S = 93)*1
0	01b	128 clock cycles (S = 128)*1
0	10b	186 clock cycles (S = 186)*1
0	11b	512 clock cycles (S = 512)*1
1	00b	32 clock cycles (S = 32) (Initial Value)*1
1	01b	64 clock cycles (S = 64)*1

Table 24.5 Combinations of the SCMR.BCP2 and SMR_SMCI.BCP[1:0] bits (2 of 2)

SCMR.BCP2 bit	SMR_SMCI.BCP[1:0] bits	Number of base clock cycles for 1-bit transfer period
1	10b	372 clock cycles (S = 372)*1
1	11b	256 clock cycles (S = 256)*1

Note 1. S is the value of S in section 24.2.17. BRR : Bit Rate Register.

24.2.17 BRR : Bit Rate Register

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x01

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

BRR is an 8-bit register that adjusts the bit rate.

As each SCI channel has independent baud rate generator control, different bit rates can be set for each channel. Table 24.6 shows the relationship between the setting (N) in the BRR and the bit rate (B) for normal asynchronous mode, multi-processor transfer, clock synchronous mode, smart card interface mode, simple SPI mode, and simple IIC mode.

The initial value of the BRR register is 0xFF. The BRR register can be read by the CPU, but it can be written to only when the TE and RE bits in SCR/SCR_SMCI are 0.

Table 24.6 Relationship between N setting in BRR and bit rate B

Mode	SEMR settings			BRR setting	Error
	BGDM bit	ABCS bit	ABCS E bit		
Asynchronous, multi-processor transfer	0	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \left\{\frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
	1	0	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \left\{\frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \left\{\frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
	Don't care	Don't care	1	$N = \frac{PCLK \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \left\{\frac{PCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (N+1)} - 1\right\} \times 100$
Clock synchronous, simple SPI				$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	—
Smart card interface				$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error\ (\%) = \left\{\frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1\right\} \times 100$
Simple IIC*1				$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	—

Note: B: Bit rate (bps)
N: BRR setting for on-chip baud rate generator (0 ≤ N ≤ 255)
PCLK: Operating frequency (MHz)
n and S: Determined by the SMR/SMR_SMCI and SCMR register settings as listed in Table 24.8 and Table 24.9.

Note 1. Adjust the bit rate so that the widths of high and low level of the SCLn output in simple IIC mode satisfy the I^2C bus standard.

Table 24.7 Calculating widths of SCLn high and low levels

Mode	SCLn	Formula (result in seconds)
IIC	Width at high level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLK \times 10^6}$
	Width at low level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLK \times 10^6}$

Table 24.8 Clock source settings

SMR or SMR_SMCI.CKS[1:0] bits setting		
CKS[1:0] bits	Clock source	n
00b	PCLK clock	0
01b	PCLK/4 clock	1
10b	PCLK/16 clock	2
11b	PCLK/64 clock	3

Table 24.9 Base clock settings in smart card interface mode

SCMR.BCP2 bit setting	SMR_SMCI.BCP[1:0] bits setting		
BCP2 bit	BCP[1:0] bits	Base clock cycles for 1-bit period	S
0	00b	93 clock cycles	93
0	01b	128 clock cycles	128
0	10b	186 clock cycles	186
0	11b	512 clock cycles	512
1	00b	32 clock cycles	32
1	01b	64 clock cycles	64
1	10b	372 clock cycles	372
1	11b	256 clock cycles	256

Table 24.10 lists examples of BRR (N) settings in normal asynchronous mode. Table 24.11 lists the maximum bit rate settable for each operating frequency. Table 24.15 lists examples of BRR (N) settings in smart card interface mode.

In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, see section 24.6.4. Receive Data Sampling Timing and Reception Margin. Table 24.12 and Table 24.14 list the maximum bit rates with external clock input.

When either the Asynchronous Mode Base Clock Select bit (ABCS) or the Baud Rate Generator Double-speed Mode Select bit (BGDM) in the Serial Extended Mode Register (SEMR) is set to 1 in asynchronous mode, the bit rate becomes twice the value listed in Table 24.16. When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 24.10 Examples of BRR settings for different bit rates in asynchronous mode (1) (1 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	-0.26	2	177	-0.25	2	212	0.03	2	217	0.08
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00

Table 24.10 Examples of BRR settings for different bit rates in asynchronous mode (1) (2 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	25	0.16	0	31	0.00	0	32	-1.36	0	38	0.16	0	39	0.00
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	-2.34	0	19	0.00
31250	0	7	0.00	0	9	-1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	-2.34	0	9	0.00

Table 24.10 Examples of BRR settings for different bit rates in asynchronous mode (1) (3 of 3)

Bit rate (bps)	Operating frequency PCLK (MHz)																	
	14			16			17.2032			18			19.6608			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	-0.17	3	70	0.03	3	75	0.48	3	79	-0.12	3	86	0.31	3	88	-0.25
150	2	181	0.16	2	207	0.16	2	223	0.00	2	233	0.16	2	255	0.00	3	64	0.16
300	2	90	0.16	2	103	0.16	2	111	0.00	2	116	0.16	2	127	0.00	2	129	0.16
600	1	181	0.16	1	207	0.16	1	223	0.00	1	233	0.16	1	255	0.00	2	64	0.16
1200	1	90	0.16	1	103	0.16	1	111	0.00	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	181	0.16	0	207	0.16	0	223	0.00	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	90	0.16	0	103	0.16	0	111	0.00	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	45	-0.93	0	51	0.16	0	55	0.00	0	58	-0.69	0	63	0.00	0	64	0.16
19200	0	22	-0.93	0	25	0.16	0	27	0.00	0	28	1.02	0	31	0.00	0	32	-1.36
31250	0	13	0.00	0	15	0.00	0	16	1.20	0	17	0.00	0	19	-1.70	0	19	0.00
38400	—	—	—	0	12	0.16	0	13	0.00	0	14	-2.34	0	15	0.00	0	15	1.73

Note: In this example, SEMR.ABCS = 0, SEMR.ABCSE = 0, and SEMR.BGDM = 0.
When either the ABCS or BGDM bit is set to 1, the bit rate doubles.
When both ABCS and BGDM are set to 1, the bit rate increases four times.

Table 24.11 maximum bit rate for each operating frequency in asynchronous mode (1 of 2)

PCLK (MHz)	SEMR settings					Maximum bit rate (bps)	PCLK (MHz)	SEMR settings					Maximum bit rate (bps)
	BGDM bit	ABCS bit	ABCSE bit	n	N			BGDM bit	ABCS bit	ABCSE bit	n	N	
8	0	0	0	0	0	250000	17.2032	0	0	0	0	0	537600
		1	0	0	0	500000			1	0	0	0	1075200
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1000000			1	0	0	0	2150400
	Don't care	Don't care	1	0	0	1333333		Don't care	Don't care	1	0	0	2867200
9.8304	0	0	0	0	0	307200	18	0	0	0	0	0	562500
		1	0	0	0	614400			1	0	0	0	1125000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1228800			1	0	0	0	2250000
	Don't care	Don't care	1	0	0	1638400		Don't care	Don't care	1	0	0	3000000

Table 24.11 maximum bit rate for each operating frequency in asynchronous mode (2 of 2)

PCLK (MHz)	SEMR settings					Maximum bit rate (bps)	PCLK (MHz)	SEMR settings					Maximum bit rate (bps)
	BGDM bit	ABCS bit	ABCSE bit	n	N			BGDM bit	ABCS bit	ABCSE bit	n	N	
10	0	0	0	0	0	312500	19.6608	0	0	0	0	0	614400
		1	0	0	0	625000			1	0	0	0	1228800
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1250000			1	0	0	0	2457600
	Don't care	Don't care	1	0	0	1666666		Don't care	Don't care	1	0	0	3276800
12	0	0	0	0	0	375000	20	0	0	0	0	0	625000
		1	0	0	0	750000			1	0	0	0	1250000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1500000			1	0	0	0	2500000
	Don't care	Don't care	1	0	0	2000000		Don't care	Don't care	1	0	0	3333333
12.288	0	0	0	0	0	384000							
		1	0	0	0	768000							
	1	0	0	0	0								
		1	0	0	0	1536000							
	Don't care	Don't care	1	0	0	2048000							
14	0	0	0	0	0	437500							
		1	0	0	0	875000							
	1	0	0	0	0								
		1	0	0	0	1750000							
	Don't care	Don't care	1	0	0	2333333							
16	0	0	0	0	0	500000							
		1	0	0	0	1000000							
	1	0	0	0	0								
		1	0	0	0	2000000							
	Don't care	Don't care	1	0	0	2666666							

Table 24.12 Maximum bit rate with external clock input in asynchronous mode

Maximum bit rate (bps)			
PCLK (MHz)	External input clock (MHz)	SEMR.ABCS = 0	SEMR.ABCS = 1
8	2.0000	125000	250000
9.8304	2.4576	153600	307200
10	2.5000	156250	312500
12	3.0000	187500	375000
12.288	3.0720	192000	384000
14	3.5000	218750	437500
16	4.0000	250000	500000
17.2032	4.3008	268800	537600
18	4.5000	281250	562500
19.6608	4.9152	307200	614400
20	5.0000	312500	625000

Table 24.13 BBR settings for different bit rates in clock synchronous and simple SPI modes

	Operating frequency PCLK (MHz)							
	8		10		16		20	
Bit rate (bps)	n	N	n	N	n	N	n	N
2.5 k	1	199	1	249	2	99	2	124
5 k	1	99	1	124	1	199	1	249
10 k	0	199	0	249	1	99	1	124
25 k	0	79	0	99	0	159	0	199
50 k	0	39	0	49	0	79	0	99
100 k	0	19	0	24	0	39	0	49
250 k	0	7	0	9	0	15	0	19
500 k	0	3	0	4	0	7	0	9
1 M	0	1			0	3	0	4
2.5 M			0	0*1			0	1
5 M							0	0*1
7.5 M								

Note: Space: Setting prohibited.
Note 1. Continuous transmission or reception is not possible. After transmitting or receiving one frame of data, a 1-bit period elapses before starting to transmit or receive the next frame of data. The output of the synchronization clock is stopped for a 1-bit period. Therefore, it takes 9 bits worth of time to transfer one frame (8 bits) of data, and the average transfer rate is 8/9 times the bit rate.

Table 24.14 Maximum bit rate with external clock input in clock synchronous mode and simple SPI mode

PCLK (MHz)	External input clock (MHz)	Maximum bit rate (Mbps)
8	1.3333	1.3333333
10	1.6667	1.6666667
12	2.0000	2.0000000
14	2.3333	2.3333333
16	2.6667	2.6666667
18	3.0000	3.0000000
20	3.3333	3.3333333

Table 24.15 BBR settings for different bit rates in smart card interface mode (n = 0, S = 372) (1 of 2)

	Operating frequency PCLK (MHz)											
	7.1424			10.00			10.7136			13.00		
Bit rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	0	0.00	0	1	-30	0	1	-25	0	1	-8.99

Table 24.15 BBR settings for different bit rates in smart card interface mode (n = 0, S = 372) (2 of 2)

	Operating frequency PCLK (MHz)											
	14.2848			16.00			18.00			20.00		
Bit rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	0.00	0	1	12.01	0	2	-15.99	0	2	-6.66

Table 24.16 Maximum bit rate for each operating frequency in smart card interface mode (S = 32) (1 of 2)

PCLK (MHz)	Maximum bit rate (bps)	n	N
10.00	156250	0	0

Table 24.16 Maximum bit rate for each operating frequency in smart card interface mode (S = 32) (2 of 2)

PCLK (MHz)	Maximum bit rate (bps)	n	N
10.7136	167400	0	0
13.00	203125	0	0
16.00	250000	0	0
18.00	281250	0	0
20.00	312500	0	0

Table 24.17 BBR settings for different bit rates in simple IIC mode

Bit rate (bps)	Operating frequency PCLK (MHz)											
	8			10			16			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	0	24	0.0	0	30	0.8	1	12	-3.8	1	15	-2.3
25 k	0	9	0.0	0	12	-3.8	1	4	0.0	1	5	4.2
50 k	0	4	0.0	0	5	4.2	1	2	-16.7	1	2	4.2
100 k[*1]	0	2	-16.7	0	3	-21.9	0	4	0.0	0	6	-10.7
250 k	0	0	0.0	0	0	25.0	0	1	0.0	0	2	-16.7
350 k	—	—	—	—	—	—	—	—	—	0	1	-10.7
400 k[*1]	—	—	—	—	—	—	—	—	—	0	1	-21.9

Note 1. The bit rate of 100 kbps and 400 kbps indicates the set value at which the error is on the minus side.

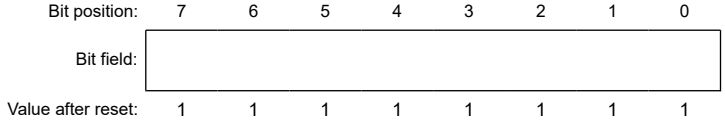
Table 24.18 Minimum widths at SCLn high and low levels for different bit rates in simple IIC mode

Bit rate (bps)	Operating frequency PCLK (MHz)											
	8			10			16			20		
	n	N	Minimum widths at SCLn high/low levels (µs)	n	N	Minimum widths at SCLn high/low levels (µs)	n	N	Minimum widths at SCLn high/low levels (µs)	n	N	Minimum widths at SCLn high/low levels (µs)
10 k	0	24	43.75/50.00	0	30	43.40/49.60	1	12	45.5/52.00	1	15	44.80/51.20
25 k	0	9	17.50/20.00	0	12	18.2/20.80	1	4	17.50/20.00	1	5	16.80/19.20
50 k	0	4	8.75/10.00	0	5	8.40/9.60	1	2	10.50/12.00	1	2	8.40/9.60
100 k	0	2	5.25/6.00	0	3	5.60/6.40	0	4	4.37/5.00	0	6	4.90/5.60
250 k	0	0	1.75/2.00	0	0	1.40/1.60	0	1	1.75/2.00	0	2	2.10/2.40
350 k	—	—	—	—	—	—	—	—	—	0	1	1.40/1.60
400 k	—	—	—	—	—	—	—	—	—	0	1	1.40/1.60

24.2.18 MDDR : Modulation Duty Register

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x12



MDDR corrects the bit rate adjusted by the BRR register.

When the BRME bit in SEMR is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected using the settings in MDDR (M/256). Table 24.19 shows the relationship between the MDDR setting (M) and the bit rate (B).

The initial value of MDDR is 0xFF. Bit [7] in this register is fixed to 1.

The CPU can read the MDDR register, but this register is only writable when the TE and RE bits in SCR/SCR_SMCI are 0.

Table 24.19 Relationship between MDDR setting (M) and bit rate (B) when bit rate modulation function is used

B: Bit rate (bps)
M: MDDR setting (128 ≤ MDDR ≤ 256)
N: BRR setting for baud rate generator (0 ≤ N ≤ 255)
PCLK: Operating frequency (MHz)
n and S: Determined by the SMR/SMR_SMCI and SCMR register settings as listed in Table 24.8 and Table 24.9 in section 24.2.17. BRR : Bit Rate Register.
x: Don't care

Mode	SEMR settings			BRR setting	Error
	BGDM bit	ABCS bit	ABCSE bit		
Asynchronous multiprocessor transfer	0	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \left(256/M\right) \times B} - 1$	$Error(\%) = \left\{\frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times \left(256/M\right) \times \left(N+1\right)} - 1\right\} \times 100$
	1	0	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times \left(256/M\right) \times B} - 1$	$Error(\%) = \left\{\frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times \left(256/M\right) \times \left(N+1\right)} - 1\right\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times \left(256/M\right) \times B} - 1$	$Error(\%) = \left\{\frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times \left(256/M\right) \times \left(N+1\right)} - 1\right\} \times 100$
	x	x	1	$N = \frac{PCLK \times 10^6}{12 \times 2^{2n-1} \times \left(256/M\right) \times B} - 1$	$Error(\%) = \left\{\frac{PCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times \left(256/M\right) \times \left(N+1\right)} - 1\right\} \times 100$
Clock synchronous, simple SPI*1				$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times \left(256/M\right) \times B} - 1$	—
Smart card interface				$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times \left(256/M\right) \times B} - 1$	$Error(\%) = \left\{\frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times \left(256/M\right) \times \left(N+1\right)} - 1\right\} \times 100$
Simple IIC*2				$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \left(256/M\right) \times B} - 1$	—

Note 1. Do not use this function in clock synchronous mode or in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).
Note 2. Adjust the bit rate so that the widths at high and low level of the SCLn output in simple IIC mode satisfy the IIC standard.

Table 24.20 and Table 24.21 list examples of N settings in BRR and M settings in MDDR in normal asynchronous mode.

Table 24.20 Examples of BRR and MDDR settings for multiple bit rates in asynchronous mode (1) (1 of 4)

Bit rate (bps)	Operating frequency PCLK (MHz)														
	8					9.8304					10				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	5	236	0	0.03	0	7	(256)*1	0	0.00	0	10	173	1	-0.01
57600	0	3	236	0	0.03	0	4	240	0	0.00	0	4	236	0	0.03
115200	0	1	236	0	0.03	0	1	192	0	0.00	0	4	236	1	0.03

Table 24.20 Examples of BRR and MDDR settings for multiple bit rates in asynchronous mode (1) (2 of 4)

Bit rate (bps)	Operating frequency PCLK (MHz)													
	8					9.8304					10				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
230400	0	0	236	0	0.03	0	0	192	0	0.00	0	1	189	1	0.14
460800	0	0	236	1	0.03	0	0	192	1	0.00	0	0	189	1	0.14

Table 24.20 Examples of BRR and MDDR settings for multiple bit rates in asynchronous mode (1) (3 of 4)

Bit rate (bps)	Operating frequency PCLK (MHz)													
	12					12.288					14				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	8	236	0	0.03	0	9	(256)*1	0	0.00	0	16	191	1	0.00
57600	0	5	236	0	0.03	0	4	192	0	0.00	0	13	236	1	0.03
115200	0	2	236	0	0.03	0	4	192	1	0.00	0	6	236	1	0.03
230400	0	2	236	1	0.03	0	2	230	1	-0.17	0	2	202	1	-0.11
460800	0	0	157	1	-0.18	0	0	154	1	-0.26	0	0	135	1	0.14

Table 24.20 Examples of BRR and MDDR settings for multiple bit rates in asynchronous mode (1) (4 of 4)

Bit rate (bps)	Operating frequency PCLK (MHz)													
	16					17.2032					18				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	11	236	0	0.03	0	13	(256)*1	0	0.00	0	18	166	1	-0.01
57600	0	7	236	0	0.03	0	6	192	0	0.00	0	18	249	1	-0.01
115200	0	3	236	0	0.03	0	6	192	1	0.00	0	8	236	1	0.03
230400	0	1	236	0	0.03	0	3	219	1	-0.20	0	1	210	0	0.14
460800	0	1	236	1	0.03	0	1	219	1	-0.20	0	0	210	0	0.14

Note 1. In this example, ABCS and ABCSE in SEMR are 0.
SEMR.BRME = 0 (M = 256) disables the bit rate modulation function.

Table 24.21 Examples of BRR and MDDR settings for different bit rates in asynchronous mode (2)

Bit rate (bps)	Operating frequency PCLK (MHz)									
	19.6608					20				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	15	(256)*1	0	0.00	0	10	173	0	-0.01
57600	0	9	240	0	0.00	0	9	236	0	0.03
115200	0	4	240	0	0.00	0	4	236	0	0.03
230400	0	1	192	0	0.00	0	4	236	1	0.03
460800	0	0	192	0	0.00	0	0	189	0	0.14

Note 1. In this example, ABCS and ABCSE in SEMR are 0.
SEMR.BRME = 0 (M = 256) disables the bit rate modulation function.

24.2.19 SEMR : Serial Extended Mode Register

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x07

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RXDESEL	BGDM	NFEN	ABCS	ABCSE	BRME	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	BRME	Bit Rate Modulation Enable 0: Disable bit rate modulation function 1: Enable bit rate modulation function	R/W*1
3	ABCSE	Asynchronous Mode Extended Base Clock Select 1 Valid only in asynchronous mode with SCR.CKE[1] = 0. 0: Clock cycles for 1-bit period determined by combination of the BGDM and ABCS bits in the SEMR register 1: Baud rate is 6 base clock cycles for 1-bit period	R/W*1
4	ABCS	Asynchronous Mode Base Clock Select Valid only in asynchronous mode. 0: Select 16 base clock cycles for 1-bit period 1: Select 8 base clock cycles for 1-bit period	R/W*1
5	NFEN	Digital Noise Filter Function Enable The NFEN bit must be 0 in all other modes. 0: In asynchronous mode: Disable noise cancellation function for RXDn input signal In simple I2C mode: Disable noise cancellation function for SCLn and SDAn input signals 1: In asynchronous mode: Enable noise cancellation function for RXDn input signal In simple I2C mode: Enable noise cancellation function for SCLn and SDAn input signals	R/W*1
6	BGDM	Baud Rate Generator Double-Speed Mode Select Valid only in asynchronous mode with SCR.CKE[1] = 0. 0: Output clock from baud rate generator with normal frequency 1: Output clock from baud rate generator with doubled frequency	R/W*1
7	RXDESEL	Asynchronous Start Bit Edge Detection Select Valid only in asynchronous mode. 0: Detect low level on RXDn pin as start bit 1: Detect falling edge of RXDn pin as start bit	R/W*1

Note 1. Writable only when the TE and RE bits in SCR/SCR_SMCI are 0 (both serial transmission and reception are disabled).

The SEMR register selects the clock source for the 1-bit period in asynchronous mode.

BRME bit (Bit Rate Modulation Enable)

The BRME bit enables or disables the bit rate modulation function. The bit rate generated by the on-chip baud rate generator is evenly corrected when this function is enabled.

ABCSE bit (Asynchronous Mode Extended Base Clock Select 1)

The ABCSE bit sets the pulse number for the base clock in a 1-bit period to 6, and the double-frequency clock is output from the baud rate generator. When the bit rate is set to 6 while dividing the bus clock frequency, use this bit and set SMR.CKS[1:0] = 00b and BRR = 0.

Set it to 0 in modes other than asynchronous mode. Even in asynchronous mode, set it to 0 when using external clock.

ABCS bit (Asynchronous Mode Base Clock Select)

The ABCS bit selects the number of clock cycles for a 1-bit period.

NFEN bit (Digital Noise Filter Function Enable)

The NFEN bit enables or disables the digital noise filter function.

When the digital noise filter function is enabled:

- Noise cancellation is applied to the RXDn input signal in asynchronous mode
- Noise cancellation is applied to the SDAn and SCLn input signals in simple I^2C mode

In all other modes, set the NFEN bit to 0 to disable the digital noise filter function. When the function is disabled, input signals are transferred as received.

BGDM bit (Baud Rate Generator Double-Speed Mode Select)

The BGDM bit selects whether or not to double the base clock frequency output from the baud rate generator.

The BGDM bit is valid when the on-chip baud rate generator is selected as the clock source (SCR.CKE[1] = 0) in asynchronous mode (SMR.CM = 0). When external clock is selected (SCR.CKE[1] = 1), set it to 0. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved and the bit rate is doubled.

Set this bit to 0 in modes other than asynchronous mode.

RXDESEL bit (Asynchronous Start Bit Edge Detection Select)

The RXDESEL bit selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data reception operation depends on the setting of this bit. Set this bit to 1 when reception must be stopped while a break occurs or when reception must be started without keeping the RXDn pin input at the high level for the period of one data frame or longer after completion of the break.

Set this bit to 0 in modes other than asynchronous mode.

24.2.20 SNFR : Noise Filter Setting Register

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x08

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

Bit	Symbol	Function	R/W
2:0	NFCS[2:0]	Noise Filter Clock Select In asynchronous mode, selects the standard setting for the base clock. In simple I^2C mode, selects the standard settings for the clock source of the on-chip baud rate generator selected in the SMR.CKS[1:0] bits. 0 0 0: In asynchronous mode: Use clock signal divided by 1 with noise filter In simple I^2C mode: Setting prohibited 0 0 1: In asynchronous mode: Setting prohibited In simple I^2C mode: Use clock signal divided by 1 with noise filter 0 1 0: In asynchronous mode: Setting prohibited In simple I^2C mode: Use clock signal divided by 2 with noise filter 0 1 1: In asynchronous mode: Setting prohibited In simple I^2C mode: Use clock signal divided by 4 with noise filter 1 0 0: In asynchronous mode: Setting prohibited In simple I^2C mode: Use clock signal divided by 8 with noise filter Others: Setting prohibited	R/W*1
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in SCR/SCR_SMCI are 0 (serial reception and transmission disabled).

The SNFR register sets the digital noise filter clock.

NFCS[2:0] bits (Noise Filter Clock Select)

The NFCS[2:0] bits select the sampling clock for the digital noise filter. To use the noise filter in asynchronous mode, set these bits to 000b. In simple I^2C mode, when 32 clocks are selected as one bit period in the basic clock selection bits of the SEMR register, set the NFCS [2: 0] bits in the range from 001b to 100b. When any other value is selected for the basic clock selection bit, set the NFCS bit to 001b.

24.2.21 SIMR1 : IIC Mode Register 1

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x09

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

Bit	Symbol	Function	R/W
0	IICM	Simple IIC Mode Select 0: SCMR.SMIF = 0: Asynchronous mode (including multi-processor mode), clock synchronous mode, or simple SPI mode SCMR.SMIF = 1: Smart card interface mode 1: SCMR.SMIF = 0: Simple IIC mode SCMR.SMIF = 1: Setting prohibited	R/W*1
2:1	—	These bits are read as 0. The write value should be 0.	R/W
7:3	IICDL[4:0]	SDAn Delay Output Select SDAn signal output delay in cycles of the clock signal from the on-chip baud rate generator. 0x00: No output delay Others: (IICDL - 1) to (IICDL) cycles	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (both serial transmission and reception are disabled).

SIMR1 selects simple IIC mode and the number of delay stages for the SDAn output.

IICM bit (Simple IIC Mode Select)

In combination with the SCMR.SMIF bit, the IICM bit selects the operating mode.

IICDL[4:0] bits (SDAn Delay Output Select)

The IICDL[4:0] bits specify an output delay on the SDAn pin relative to the falling edge of the output on the SCLn pin.

The available delay settings range from no delay to 31 cycles, with the clock signal from the on-chip baud rate generator as the base. The signal obtained by frequency-dividing PCLK by the divisor set in SMR.CKS[1:0] is supplied as the clock signal from the on-chip baud rate generator. Set 00000b to IICDL[4:0] bits unless operation is in simple IIC mode. In simple IIC mode, set the bits to a value in the range from 00001b to 11111b.

Table 24.22 Settable value of IICDL[4: 0] bits in each communication mode

Communication mode	ABCS	Settable value of IICDL[4:0] bits
Other than simple IIC mode	Don't care	00000b
Simple IIC mode	0	00001b to 11111b
	1	00001b to 00100b

24.2.22 SIMR2 : IIC Mode Register 2

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x0A

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

Bit	Symbol	Function	R/W
0	IICINTM	IIC Interrupt Mode Select 0: Use ACK/NACK interrupts 1: Use reception and transmission interrupts	R/W*1
1	IICCSC	Clock Synchronization 0: Do not synchronize with clock signal 1: Synchronize with clock signal	R/W*1
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	IICACKT	ACK Transmission Data 0: ACK transmission 1: NACK transmission and ACK/NACK reception	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (serial reception and transmission disabled).

SIMR2 selects how reception and transmission are controlled in simple IIC mode.

IICINTM bit (IIC Interrupt Mode Select)

The IICINTM bit selects the sources of interrupt requests in simple IIC mode.

IICCSC bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SCLn clock signal is to be synchronized when the SCLn pin is driven low because a wait was inserted by another other device.

The SCLn clock signal is not synchronized if the IICCSC bit is 0. The SCLn clock signal is generated according to the rate selected in the BRR register regardless of the level being input on the SCLn pin.

Set the IICCSC bit to 1 except during debugging.

IICACKT bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set the IICACKT bit to 1 when ACK and NACK bits are received.

24.2.23 SIMR3 : IIC Mode Register 3

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x0B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IICSCLS[1:0]		IICSDAS[1:0]		IICSTIF	IICSTPREQ	IICRSTAREQ	IICSTAREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IICSTAREQ	Start Condition Generation 0: Do not generate start condition 1: Generate start condition*1 *3 *5 *6	R/W

Bit	Symbol	Function	R/W
1	IICRSTAREQ	Restart Condition Generation 0: Do not generate restart condition 1: Generate restart condition*2 *3 *5 *6	R/W
2	IICSTPREQ	Stop Condition Generation 0: Do not generate stop condition 1: Generate stop condition*2 *3 *5 *6	R/W
3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag 0: No requests are being made for generating conditions, or a condition is being generated 1: Generation of start, restart, or stop condition is complete. When 0 is written to IICSTIF, it is set to 0*4	R/W*4
5:4	IICSDAS[1:0]	SDAn Output Select 0 0: Output serial data 0 1: Generate start, restart, or stop condition 1 0: Output low on SDAn pin 1 1: Drive SDAn pin to high-impedance state	R/W
7:6	IICSCLS[1:0]	SCLn Output Select 0 0: Output serial clock 0 1: Generate start, restart, or stop condition 1 0: Output low on SCLn pin 1 1: Drive SCLn pin to high-impedance state	R/W

Note 1. Only generate a start condition after checking the bus state and confirming that the bus is free.
Note 2. Generate a restart or stop condition after checking the bus state and confirming that the bus is busy.
Note 3. Do not set more than one of the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.
Note 4. Write only 0. When 1 is written, the value is ignored.
Note 5. Execute the generation of a condition after the value of the IICSTIF flag is 0.
Note 6. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

The SIMR3 register is used to control the start, restart, and stop conditions in the simple I^2C mode, and to hold the SSDAn and SSCLn pins at fixed levels.

IICSTAREQ bit (Start Condition Generation)

When a start condition is to be generated, set both IICSDAS[1:0] and IICSCLS[1:0] to 01b in addition to setting the IICSTAREQ bit to 1.

[Setting condition]

- On writing 1 to the bit.

[Clearing condition]

- On completion of start condition generation.

IICRSTAREQ bit (Restart Condition Generation)

When a restart condition is to be generated, set both IICSDAS[1:0] and IICSCLS[1:0] to 01b in addition to setting the IICRSTAREQ bit to 1.

[Setting condition]

- On writing 1 to the bit.

[Clearing condition]

- On completion of restart condition generation.

IICSTPREQ bit (Stop Condition Generation)

When a stop condition is to be generated, set both IICSDAS[1:0] and IICSCLS[1:0] to 01b in addition to setting the IICSTPREQ bit to 1.

[Setting condition]

- On writing 1 to the bit.

[Clearing condition]

- On completion of stop condition generation.

IICSTIF flag (Issuing of Start, Restart, or Stop Condition Completed Flag)

After generating a condition, the IICSTIF flag indicates that the condition generation is complete. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.

When the IICSTIF flag is 1 while an interrupt request is enabled by setting the SCR.TEIE bit, an STI request is output.

[Setting condition]

- On completion of a start, restart, or stop condition generation.

  If the setting condition conflicts with any of the clearing conditions for the flag, the clearing condition takes precedence.

[Clearing conditions]

- On writing 0 to the bit. After writing 0 to the IICSTIF bit, read the bit to check that it is actually set to 0.
- On writing 0 to the SIMR1.IICM bit when operation is not in simple IIC mode.
- On writing 0 to the SCR.TE bit.

IICSDAS[1:0] bits (SDAn Output Select)

The IICSDAS[1:0] bits control output from the SDAn pin. Set IICSDAS[1:0] and IICSCLS[1:0]to the same value during normal operations.

IICSCLS[1:0] bits (SCLn Output Select)

The IICSCLS[1:0] bits control output from the SCLn pin. Set IICSDAS[1:0] and IICSCLS[1:0] to the same value during normal operations.

24.2.24 SISR : IIC Status Register

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x0C

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

Bit	Symbol	Function	R/W
0	IICACKR	ACK Reception Data Flag 0: ACK received 1: NACK received	R
1	—	This bit is read as 0.	R
2	—	The read value is undefined.	R
3	—	This bit is read as 0.	R
5:4	—	The read value is undefined.	R
7:6	—	These bits are read as 0.	R

SISR monitors the state in simple IIC mode.

IICACKR flag (ACK Reception Data Flag)

Received ACK and NACK bits can be read from the IICACKR flag. The IICACKR flag is updated on the rising edge of the SCLn clock for the received ACK/NACK bit.

24.2.25 SPMR : SPI Mode Register

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x0D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CKPH	CKPOL	—	MFF	—	MSS	CTSE	SSE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SSE	SSn Pin Function Enable 0: Disable SSn pin function 1: Enable SSn pin function	R/W*1
1	CTSE	CTS Enable 0: Disable CTS function (enable RTS output function) 1: Enable CTS function	R/W*1
2	MSS	Master Slave Select 0: Transmit through TXDn pin and receive through RXDn pin (master mode) 1: Receive through TXDn pin and transmit through RXDn pin (slave mode)	R/W*1
3	—	This bit is read as 0. The write value should be 0.	R/W
4	MFF	Mode Fault Flag 0: No mode fault error 1: Mode fault error	R/W*2
5	—	This bit is read as 0. The write value should be 0.	R/W
6	CKPOL	Clock Polarity Select 0: Do not invert clock polarity 1: Invert clock polarity	R/W*1
7	CKPH	Clock Phase Select 0: Do not delay clock 1: Delay clock	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (both serial transmission and reception are disabled).
Note 2. Only 0 can be written to this bit, to clear the flag.

SPMR selects the extension settings in asynchronous and clock synchronous modes.

SSE bit (SSn Pin Function Enable)

Set the SSE bit to 1 to use the SSn pin to control transmission and reception in simple SPI mode. Set this bit to 0 in all other modes. In simple SPI mode, when master mode is selected (SCR.CKE[1:0] = 00b and SPMR.MSS = 0) and there is a single master, the SSn pin on the master side is not required to control reception and transmission. In such a case, set the SSE bit to 0. Do not set both the SSE and CTSE bits to 1. If this setting is made, operation is the same as that when these bits are set to 0.

CTSE bit (CTS Enable)

Set the CTSE bit to 1 if the SSn pin is to be used for inputting the CTS control signal to control transmission and reception. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, and simple IIC mode. Do not set both the CTSE and SSE bits to 1. If this setting is made, operation is the same as that when these bits are set to 0.

MSS bit (Master Slave Select)

The MSS bit selects master or slave operation in simple SPI mode. The functions of the TXDn and RXDn pins are reversed when this bit is set to 1, so that data is received through the TXDn pin and transmitted through the RXDn pin.

Set this bit to 0 in modes other than simple SPI mode.

MFF flag (Mode Fault Flag)

The MFF flag indicates mode fault errors. In a multi-master configuration, determine the mode fault error occurrence by reading this flag.

[Setting condition]

- When input on the SSn pin is low during master operation in simple SPI mode (SSE bit = 1 and MSS bit = 0).

[Clearing condition]

- On writing 0 to the bit after it is read as 1.

CKPOL bit (Clock Polarity Select)

The CKPOL bit selects the polarity of the clock signal output through the SCKn pin. See Figure 24.69 for details. Set the CKPOL bit to 0 in all modes other than simple SPI mode and clock synchronous mode.

CKPH bit (Clock Phase Select)

The CKPH bit selects the phase of the clock signal output through the SCKn pin. See Figure 24.69 for details. Set the CKPH bit to 0 in all modes other than simple SPI mode and clock synchronous mode.

24.2.26 FCR : FIFO Control Register

Base address: SCI0 = 0x4007_0000

Offset address: 0x14

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RSTRG[3:0]				RTRG[3:0]				TTRG[3:0]				DRES	TFRST	RFRST	FM
Value after reset:	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FM	FIFO Mode Select Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode. 0: Non-FIFO mode. Selects TDR/RDR or TDRHL/RDRHL for communication. 1: FIFO mode. Selects FTDRHL/FRDRHL for communication.	R/W*1
1	RFRST	Receive FIFO Data Register Reset Valid only when FCR.FM = 1. 0: Do not reset FRDRHL 1: Reset FRDRHL	R/W
2	TFRST	Transmit FIFO Data Register Reset Valid only when FCR.FM = 1. 0: Do not reset FTDRHL 1: Reset FTDRHL	R/W
3	DRES	Receive Data Ready Error Select Selects the interrupt requested when detecting receive data ready. 0: Receive data full interrupt (SCIn_RXI) 1: Receive error interrupt (SCIn_ERI)	R/W
7:4	TTRG[3:0]	Transmit FIFO Data Trigger Number Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode. The trigger number is specified in the TTRG[3:0] bits.	R/W
11:8	RTRG[3:0]	Receive FIFO Data Trigger Number Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode. The trigger number is specified in the RTRG[3:0] bits.	R/W

Bit	Symbol	Function	R/W
15:12	RSTRG[3:0]	RTS Output Active Trigger Number Select Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode, when FCR.FM = 1, SPMR.CTSE = 0, and SPMR.SSE = 0. The trigger number is specified in the RSTRG[3:0] bits.	R/W

Note 1. Writable only when TE = 0 and RE = 0.

FCR selects FIFO mode, resets FTDRHL and FRDRHL, selects the FIFO data trigger number for transmission or reception, and selects the RTS output active trigger number.

FM bit (FIFO Mode Select)

When the FM bit is set to 1, FTDRHL and FRDRHL are selected for communication. When the FM bit is set to 0, TDR and RDR, or TDRHL and RDRHL are selected for communication.

RFRST bit (Receive FIFO Data Register Reset)

When the RFRST bit is set to 1, the FRDRHL register is reset and the received data count resets to 0. When 1 is written to the RFRST bit, it clears to 0 after 1 PCLK.

TFRST bit (Transmit FIFO Data Register Reset)

When the TFRST bit is set to 1, the FTDRHL register is reset and the transmit data count resets to 0. When 1 is written to the TFRST bit, it clears to 0 after 1 PCLK.

DRES bit (Receive Data Ready Error Select)

When detecting a receive data ready error, the selection can be made from an SCIn_RXI interrupt request or an SCIn_ERI interrupt request. When starting DTC and reading from the FRDRH and FRDRL registers, set the DRES bit to 1.

TTRG[3:0] bits (Transmit FIFO Data Trigger Number)

The TDFE flag is set to 1 when the amount of transmit data in FTDRHL is equal to or less than the transmit triggering number specified in the TTRG[3:0] bits, and software can write data to FTDRHL. If SCR.TIE = 1, an SCIn_TXI interrupt request occurs.

RTRG[3:0] bits (Receive FIFO Data Trigger Number)

The RDF flag is set to 1 when the amount of receive data in FRDRHL is equal to or greater than the receive triggering number specified in the RTRG[3:0] bits, and software can read data from FRDRHL. If SCR.RIE = 1, an SCIn_RXI interrupt request occurs.

When RTRG[3:0] is 0, the RDF flag is not set even when the amount of data in the receive FIFO is equal to 0, and an SCIn_RXI interrupt does not occur.

RSTRG[3:0] bits (RTS Output Active Trigger Number Select)

When the amount of receive data stored in FRDRHL is equal to or greater than the receive triggering number specified in the RSTRG[3:0] bits, the RTS signal goes high.

When RSTRG[3:0] is 0, the RTS signal does not go high even when the amount of data in FRDRHL is equal to 0.

24.2.27 FDR : FIFO Data Count Register

Base address: SCI0 = 0x4007_0000

Offset address: 0x16

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

Bit	Symbol	Function	R/W
4:0	R[4:0]	Receive FIFO Data Count Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode, when FCR.FM = 1. Indicates the amount of receive data stored in FRDRHL.	R
7:5	—	These bits are read as 0.	R
12:8	T[4:0]	Transmit FIFO Data Count Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode, when FCR.FM = 1. Indicates the amount of non-transmitted data stored in FTDRHL.	R
15:13	—	These bits are read as 0.	R

The FDR register indicates the amount of data stored in FRDRHL and FTDRHL.

R[4:0] bits (Receive FIFO Data Count)

The R[4:0] bits indicate the amount of receive data stored in FRDRHL. 0x00 means no receive data, and 0x10 means that the maximum received data is stored in FRDRHL.

T[4:0] bits (Transmit FIFO Data Count)

The T[4:0] bits indicate the amount of non-transmitted data stored in FTDRHL. 0x00 means no transmit data, and 0x10 means that all (maximum amount) of the data to be transmitted is stored in FTDRHL.

24.2.28 LSR : Line Status Register

Base address: SCI0 = 0x4007_0000

Offset address: 0x18

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

Bit	Symbol	Function	R/W
0	ORER	Overrun Error Flag Valid only in asynchronous mode, including multi-processor mode, or clock synchronous mode, and when FIFO is selected. 0: No overrun error occurred 1: Overrun error occurred	R*1
1	—	This bit is read as 0.	R
6:2	FNUM[4:0]	Framing Error Count Indicates the amount of data with a framing error in the receive data stored in FRDRHL.	R
7	—	This bit is read as 0.	R
12:8	PNUM[4:0]	Parity Error Count Indicates the amount of data with a parity error in the receive data stored in FRDRHL.	R
15:13	—	These bits are read as 0.	R

Note 1. Write 0 to SSR_FIFO.ORER to clear the flag.

The LSR register indicates the receive error status.

ORER flag (Overrun Error Flag)

The ORER flag reflects the value in SSR_FIFO.ORER.

FNUM[4:0] bits (Framing Error Count)

The FNUM[4:0] value indicates the amount of data with a framing error stored in the FRDRHL register.

PNUM[4:0] bits (Parity Error Count)

The PNUM[4:0] value indicates the amount of data with a parity error stored in the FRDRHL register.

24.2.29 CDR : Compare Match Data Register

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x1A

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

Bit	Symbol	Function	R/W
8:0	CMPD[8:0]	Compare Match Data Holds compare data pattern for address match wakeup function.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

The CDR register sets the compare data for the address match function.

CMPD[8:0] bits (Compare Match Data)

The CMPD[8:0] bits set the data to be compared to receive data for the address match function, when the address match function is enabled (DCCR.DCME = 1).

Three bit lengths are available:

- CMPD[6:0] with 7-bit length
- CMPD[7:0] with 8-bit length
- CMPD[8:0] with 9-bit length

24.2.30 DCCR : Data Compare Match Control Register

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x13

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DCME	IDSEL	—	DFER	DPER	—	—	DCMF
Value after reset:	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DCMF	Data Compare Match Flag 0: Not matched 1: Matched	R/(W)*1
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	DPER	Data Compare Match Parity Error Flag 0: No parity error occurred 1: Parity error occurred	R/(W)*1
4	DFER	Data Compare Match Framing Error Flag 0: No framing error occurred 1: Framing error occurred	R/(W)*1
5	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6	IDSEL	ID Frame Select Valid only in asynchronous mode, including multi-processor mode. 0: Always compare data regardless of the MPB bit value 1: Only compare data when MPB bit = 1 (ID frame)	R/W
7	DCME	Data Compare Match Enable Valid only in asynchronous mode, including multi-processor mode. 0: Disable address match function 1: Enable address match function	R/W

Note 1. Only 0 can be written, to clear the flag after reading 1.

The DCCR register controls the address match function.

DCMF flag (Data Compare Match Flag)

The DCMF flag indicates that the SCI detected a receive data match with the comparison data (CDR.CMPD).

[Setting condition]

- On match of the comparison data (CDR.CMPD) with the receive data when DCCR.DCME = 1.

[Clearing condition]

- When 0 is written after 1 is read from DCMF.

Clearing the SCR.RE bit to 0 does not affect the DCMF flag, which retains its previous value.

DPER flag (Data Compare Match Parity Error Flag)

The DPER flag indicates that a parity error occurred on address match detection (receive data match detection).

[Setting condition]

- When a parity error is detected in a frame in which an address match is detected.

[Clearing conditions]

- When 0 is written after 1 is read from DPER.

When the SCR.RE bit is set to 0 (serial reception is disabled), the DPER flag is not affected and retains its previous value.

DFER flag (Data Compare Match Framing Error Flag)

The DFER flag indicates that a framing error occurred on address match detection (receive data match detection).

[Setting conditions]

- When a stop bit of a frame in which an address match is detected is 0.

  When in 2-stop-bit mode, only the first bit of the stop bits is checked for a value of 1 (the second stop bit is not checked).

[Clearing conditions]

- When 0 is written after 1 is read from DFER.

When the SCR.RE bit is set to 0 (serial reception is disabled), the DFER flag is not affected and retains its previous value.

IDSEL bit (ID Frame Select)

The IDSEL bit selects whether to compare data regardless of the MPB bit value or to compare data only when MPB = 1 (ID frame), when the address match function is enabled.

DCME bit (Data Compare Match Enable)

The DCME bit enables or disables the address match function (data compare match function).

If the SCI detects a match to the comparison data (CDR.CMPD) with the receive data, the DCME bit clears automatically, after which SCI operation mode is in normal receive mode. See section 24.3.6. Address Match (Receive Data Match Detection) Function.

The write value must be 0 for all modes other than asynchronous mode.

24.2.31 SPTR : Serial Port Register

Base address: SCI0 = 0x4007_0000
SCIm = 0x4007_0000 + 0x0020 × m (m = 1 to 2, 9)

Offset address: 0x1C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SPB2IO	SPB2DT	RXDMON
Value after reset:	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	RXDMON	Serial Input Data Monitor Indicates the state of the RXDn pin. 0: RXDn terminal is the low level. 1: RXDn terminal is the High level.	R
1	SPB2DT	Serial Port Break Data Select Selects the output level of the TXDn pin when SCR.TE = 0. 0: Low level is output in TXDn terminal. 1: High level is output in TXDn terminal.	R/W
2	SPB2IO	Serial Port Break I/O Selects whether the value of SPB2DT is output to TXDn pin. 0: Do not output value of SPB2DT bit on TXDn pin 1: Output value of SPB2DT bit on TXDn pin	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The SPTR register provides confirmation of the serial reception pin (RXDn pin) status and sets the transmission pin status.

This register can only be used in asynchronous mode.

The TXDn pin status is determined by the combination of SCR.TE, SPTR.SPB2IO, and SPTR.SPB2DT settings, as shown in Table 24.23.

Table 24.23 TXDn pin status

Value of SCR.TE	Value of SPTR.SPB2IO	Value of SPTR.SPB2DT	TXDn pin status
0	0	—	Hi-Z (initial value)
0	1	0	Low level output
0	1	1	High level output
1	—	—	Serial transmit data is output

Note: —: Do not care.
Note: Use the SPTR register in asynchronous mode only. Using this register in any other mode is not guaranteed.

24.3 Operation in Asynchronous Mode

Figure 24.2 shows the general format for asynchronous serial communications. One frame consists of a start bit (low level), transmit or receive data, a parity bit, and stop bits (high level). In asynchronous serial communications, the communications line is usually held in the mark state (high level).

The SCI monitors the communications line. When the SCI detects a low, it regards that as a start bit and starts serial communication.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and receiver have a double-buffered structure in addition to FIFO mode, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

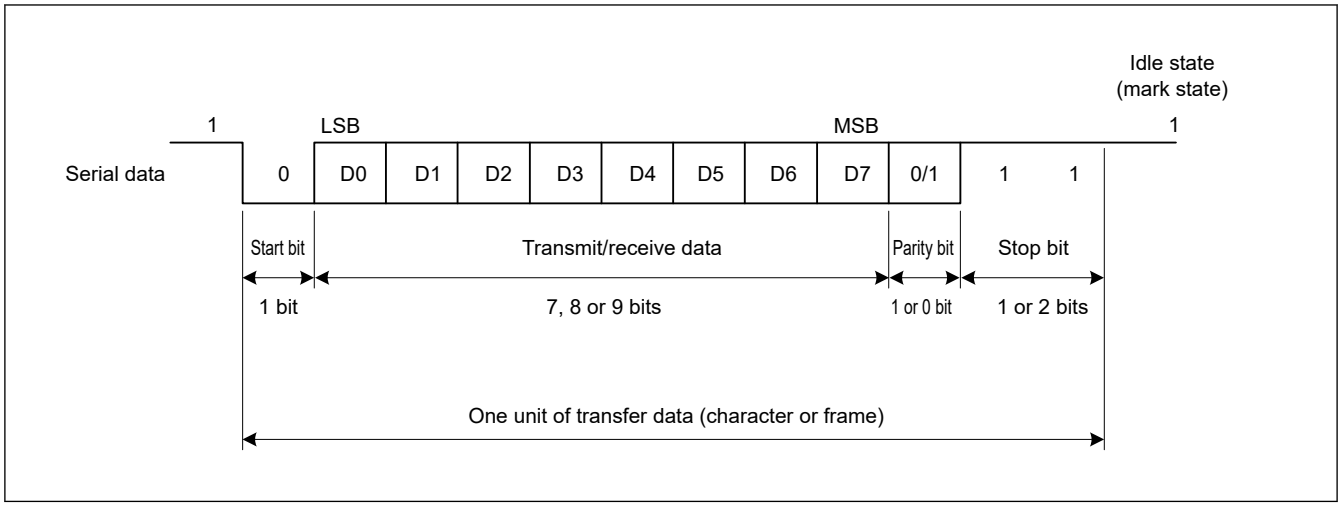


Figure 24.2 Data format in asynchronous serial communications with 8-bit data, parity bit, and 2 stop bits

24.3.1 Serial Data Transfer Format

Table 24.24 lists the serial data transfer formats that can be used in asynchronous mode. Any of 18 transfer formats can be selected with the SMR and SCMR settings. For details on the multi-processor function, see section 24.4. Multi-Processor Communication Function.

Table 24.24 Serial transfer formats in asynchronous mode (1 of 2)

SCMR setting	SMR setting				Serial transfer format and frame length
CHR1	CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	ST 9-bit data SP
0	0	0	0	1	ST 9-bit data SP SP
0	0	1	0	0	ST 9-bit data P SP
0	0	1	0	1	ST 9-bit data P SP SP
1	0	0	0	0	ST 8-bit data SP
1	0	0	0	1	ST 8-bit data SP SP
1	0	1	0	0	ST 8-bit data P SP
1	0	1	0	1	ST 8-bit data P SP SP
1	1	0	0	0	ST 7-bit data SP
1	1	0	0	1	ST 7-bit data SP SP
1	1	1	0	0	ST 7-bit data P SP
1	1	1	0	1	ST 7-bit data P SP SP

Table 24.24 Serial transfer formats in asynchronous mode (2 of 2)

SCMR setting	SMR setting				Serial transfer format and frame length
CHR1	CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	—	1	0	ST | 9-bit data | MPB | SP
0	0	—	1	1	ST | 9-bit data | MPB | SP | SP
1	0	—	1	0	ST | 8-bit data | MPB | SP
1	0	—	1	1	ST | 8-bit data | MPB | SP | SP
1	1	—	1	0	ST | 7-bit data | MPB | SP
1	1	—	1	1	ST | 7-bit data | MPB | SP | SP

ST: Start bit
SP: Stop bit
P: Parity bit
MPB: Multi-processor bit

24.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a base clock with a frequency of 16 times[*1] the bit rate.

In reception, the SCI samples the falling edge of the start bit using the base clock, and performs internal synchronization.

Because receive data is sampled on the rising edge of the 8th pulse[*1] of the base clock, data is latched at the middle of each bit, as shown in Figure 24.3 The reception margin in asynchronous mode is determined by the following formula (1):

$$M = \left|\left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F)\right| \times 100\ [\%] \quad \ldots \quad \text{Formula (1)}$$

Note: M: Reception margin
N: Ratio of bit rate to clock
(N = 16 when SEMR.ABCSE = 0 and SEMR.ABCS = 0,
N = 8 when SEMR.ABCS = 1,
N = 6 when SEMR.ABCSE = 1)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 13)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined using the following formula:
$M = \{0.5 - 1/(2 \times 16)\} \times 100\ (\%) = 46.875\ \%$

This represents the computed value. Renesas recommends a margin of 20% to 30% in system design.

Note 1. In this example, the SEMR.ABCS bit is 0 and the SEMR.ABCSE is 0. When the ABCS bit is 1 and the ABCSE bit is 0, a frequency of 8 times the bit rate is used as a base clock, and receive data is sampled on the rising edge of the 4th pulse of the base clock.

When the ABCSE bit is 1, a sextuple frequency of a bit rate is used as a base clock, and receive data is sampled on the rising edge of the 3rd pulse of the base clock.

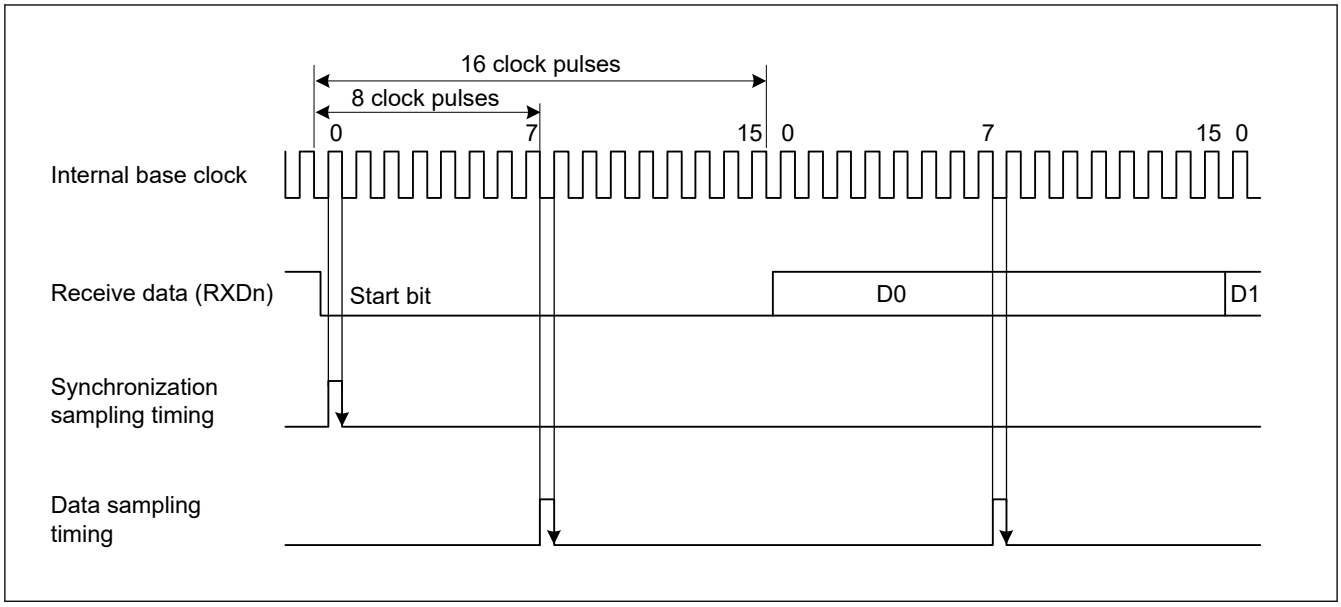


Figure 24.3 Receive data sampling timing in asynchronous mode

24.3.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input to the SCKn pin can be selected as the transfer clock of the SCI, based on the SMR.CM and SCR.CKE[1:0] settings.

When an external clock is input to the SCKn pin, the clock frequency must be 16 times the bit rate (when SEMR.ABCS = 0) or 8 times the bit rate (when SEMR.ABCS = 1).

When the SCI uses its internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is configured so that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 24.4.

When clock output is enabled, the clock is output after setting the SCR.TE or SCR.RE bit to 1.

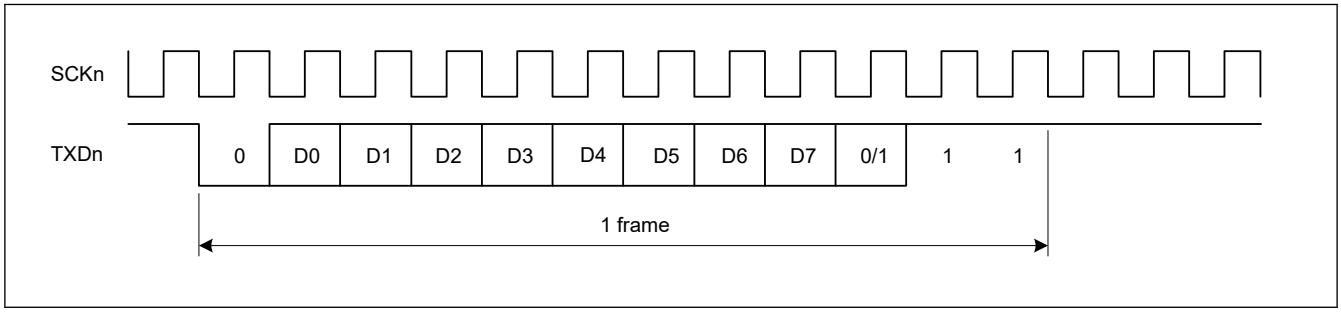


Figure 24.4 Phase relationship between output clock and transmit data in asynchronous mode when SMR.CHR = 0, PE = 1, MP = 0, and STOP = 1

24.3.4 Double-Speed Operation and Frequency of 6 Times the Bit Rate

When the SEMR.ABCS bit is set to 1 and eight pulses of the base clock for a 1-bit period is selected, the SCI operates on the bit rate twice that of when ABCS is set to 0. When the SEMR.BGDM bit is set to 1, the cycle of the base clock is half and the bit rate is double that of when BGDM is set to 0. When the SCR.CKE[1] bit is set to 0 and the on-chip baud rate generator is selected, setting the ABCS and BGDM bits to 1 allows the SCI to operate at a bit rate four times that when the ABCS and BGDM bits are set to 0.

When the SEMR.ABCSE bit is set to 1, the number of base clock pulses is 6 during a period of 1 bit, and the SCI operates at a bit rate 16/3 times that when SEMR.ABCS = 0, SEMR.BGDM = 0, and SEMR.ABCSE = 0.

As shown by Formula (1) in section 24.3.2. Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, the reception margin decreases when the SEMR.ABCS or SEMR.ABCSE bit is set to 1. Therefore, if the target bit rate can be obtained with ABCS or ABCSE set to 0, it is recommended that you use the SCI with ABCS and ABCSE set to 0.

24.3.5 CTS and RTS Functions

The CTS function uses input on the CTSn_RTSn pin in transmission control. Setting the SPMR.CTSE bit to 1 enables the CTS function.

When the CTS function is enabled, placing a low level on the CTSn_RTSn pin causes transmission to start.

Driving the CTSn_RTSn pin high while transmission is in progress does not affect transmission of the current frame.

In the RTS function, which uses output on the CTSn_RTSn pin, a low level is output when reception becomes possible. Conditions for output of the low and high levels are shown in this section.

[Conditions for low level output]

Satisfaction of all conditions are listed in this section.

Non-FIFO selected

- The value of the SCR.RE bit is 1
- Reception is not in progress
- There is no received data yet to be read
- The ORER, FER, and PER flags in the SSR register are all 0

FIFO selected

- The value of the SCR.RE bit is 1
- The amount of receive data written in FRDRHL is equal to or less than the setting value of FCRH.RSTRG[3:0]
- The ORER flag in the SSR_FIFO register (ORER in FRDRH) is 0

[Condition for high level output]

- The conditions for low-level output are not satisfied

24.3.6 Address Match (Receive Data Match Detection) Function

The address match function can be used only in asynchronous mode.

If the DCCR.DCME bit is set to 1[*2], when one frame of data is received, the SCI compares that received data with the data set in CDR.CMPD. If the SCI detects a match to the comparison data (CDR.CMPD[*1]) with the received data, the SCI can issue the SCIn_RXI interrupt request.

If the SMR.MP bit is set to 0, comparison occurs only for valid data in receive format. In multi-processor mode (SMR.MP bit = 1), if the DCCR.IDSEL bit is set to 1, receive data where the MPB bit is 1 is subject to comparison for address match and receive data where the MPB bit is 0 is always treated as a non-match.

If the DCCR.IDSEL bit is set to 0, SCI performs address match detection regardless of the MPB bit value of the received data.

Until SCI detects a match to the comparison data (CDR.CMPD[*1]) with receive data, received data is skipped (discarded), and the SCI cannot detect a parity error or framing error.

When SCI detects a match, the DCCR.DCME bit is automatically cleared, and the DCCR.DCMF flag is set to 1. If the DCCR.IDSEL bit is set to 1, the SCR.MPIE bit is automatically cleared. If DCCR.IDSEL is set to 0, the value of the SCR.MPIE bit is retained. If the SCR.RIE bit is set to 1, the SCI issues an SCIn_RXI interrupt request.

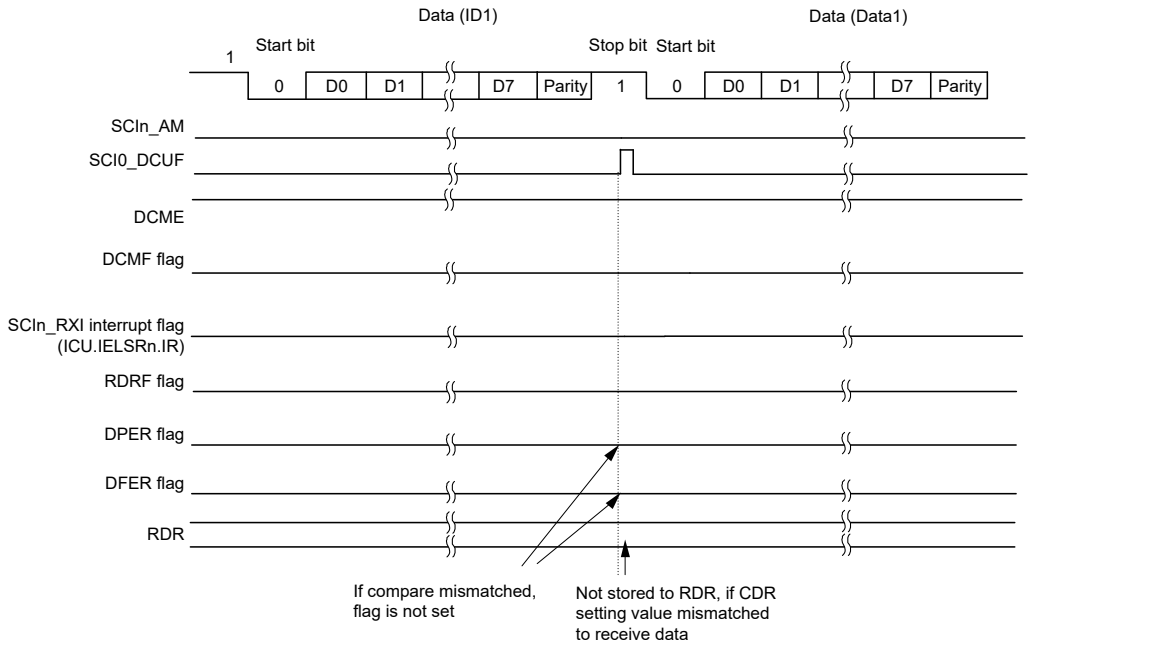
If the SCI detects a framing error in the receive data for which a match is detected, the DCCR.DFER flag is set to 1, and if the SCI detects a parity error in that frame, the DCCR.DPER flag is set to 1. The compared receive data is not stored in the RDR register, and SSR.RDRF remains 0. When FCR.FM = 1, the RDR register indicates the FRDRHL register, and the SSR.RDRF flag indicates the SSR_FIFO.RDF flag.

After the SCI detects a match, and DCCR.DCME is automatically cleared, the SCI receives the next data continuously based on the current register setting.

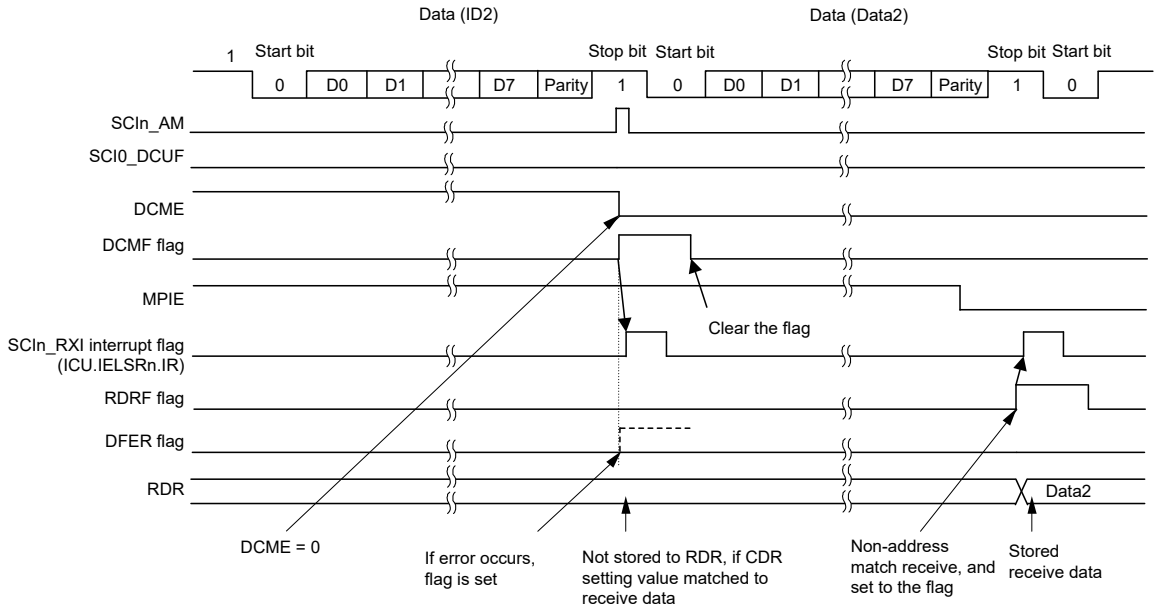
When the DCCR.DFER or DCCR.DPER flag is set, the address match is not performed. Before enabling the address match function, set the DCCR.DFER and DCCR.DPER flags to 0.

Examples of the address match function are shown in Figure 24.5 and Figure 24.6.

Note 1. This comparative target can select one length of 3 types: CMPD[6:0] with 7-bit length, CMPD[7:0] with 8-bit length, and CMPD[8:0] with 9-bit length.
Note 2. Set the DCCR.DCME bit to 1 before receiving the start bit of the received frame that performs address matching.

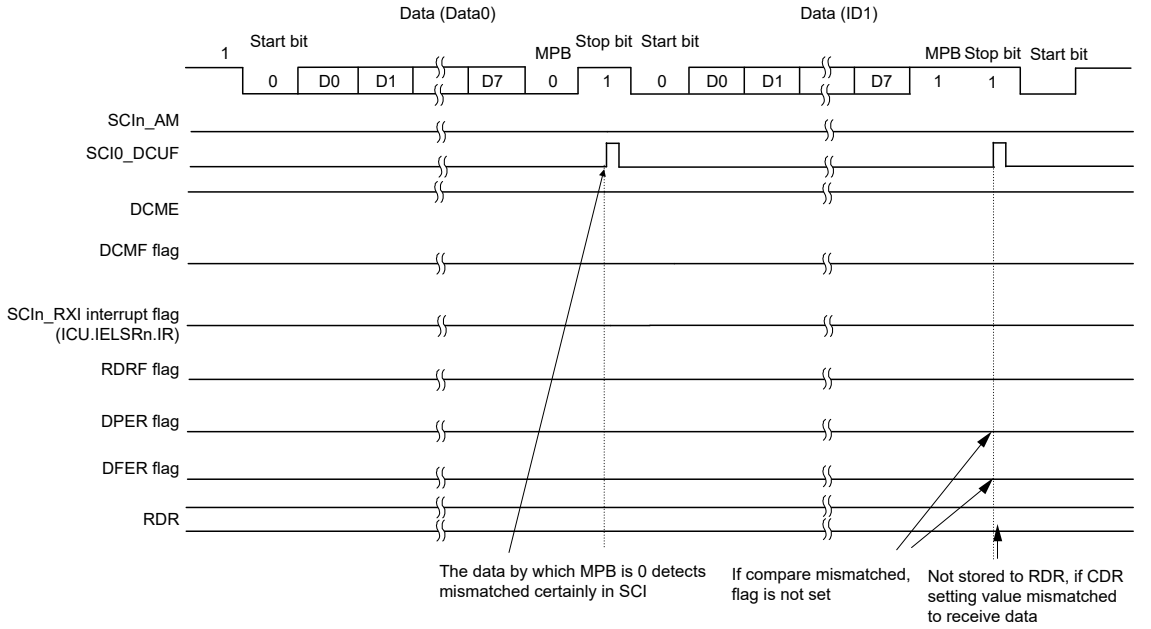


(a) Example of compare mismatched between receive data and CDR (8-bit length/parity/non multi-processor mode)

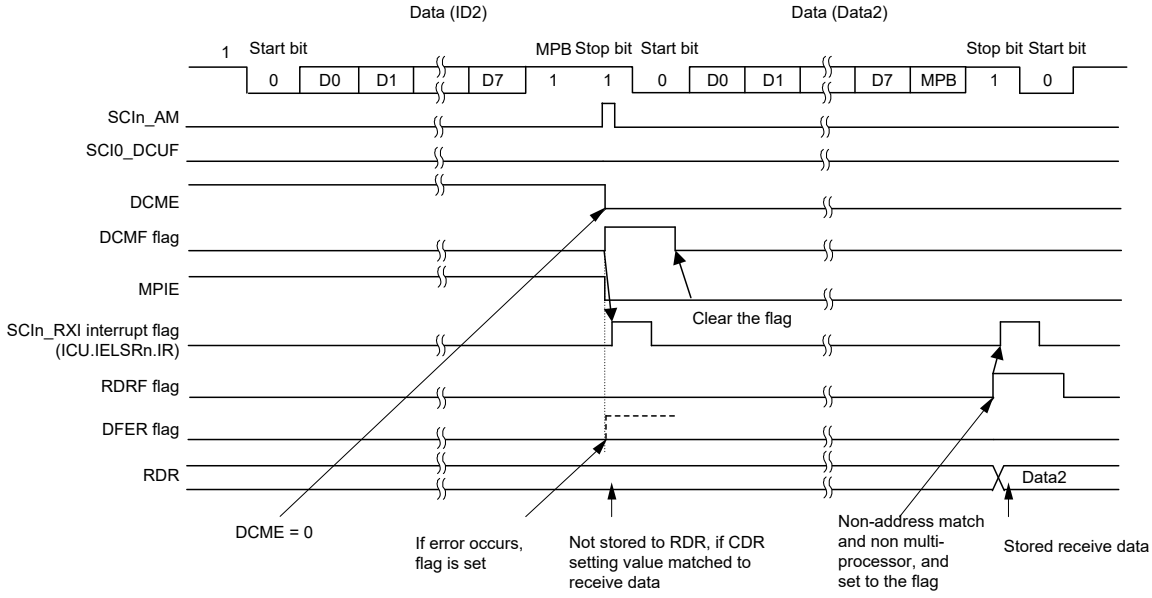


(b) Example of compare matched between receive data and CDR (8-bit length/parity/non multi-processor mode)

Figure 24.5 Example of address match (1) normal mode



(a) Example of compare mismatched between receive data and CDR (8-bit length/IDSEL = 1/multi-processor mode)



(b) Example of compare matched between receive data and CDR (8-bit length/IDSEL = 1/multi-processor mode)

Figure 24.6 Example of address match (2) multi-processor mode

24.3.7 SCI Initialization in Asynchronous Mode

Before transmitting and receiving data, start by writing the initial value 0x00 to the SCR register, then continue through the SCI initialization procedure (select non-FIFO or FIFO) shown in Table 24.25 and Table 24.26. Whenever the operating mode or transfer format is to be changed, the SCR register must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied during initialization.

Note: Setting the SCR.RE bit to 0 initializes neither the ORER, FER, RDRF, RDF, PER, and DR flags in SSR/SSR_FIFO nor RDR and RDRHL. When the TE bit is set to 0, the TEND flag for the selected FIFO buffer is not initialized.

Note: In non-FIFO mode, switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of an SCIn_TXI interrupt request.

Table 24.25 Example flow of SCI initialization in asynchronous mode with non-FIFO selected

No.	Step Name	Description
1	Start initialization	
2	Set the SCR.TIE, RIE, TE, RE, and TEIE bits to 0	
3	Set the FCR.FM bit to 0	Set the FCR.FM bit to 0.
4	Set the SCR.CKE[1:0] bits	Set the clock selection in SCR. When the clock output is selected in asynchronous mode, the clock is output immediately after SCR settings are made.
5	Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits to 0.	Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits to 0. Step 5 can be skipped if the values have not been changed from the initial values.
6	Set the data transmission/ reception format in SMR, SCMR, and SEMR	Set data transmission/reception format in SMR, SCMR, and SEMR.
7	Set a value in BRR	Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.
8	Set a value in MDDR	Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is set to 0 or an external clock is used.
9	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
10	Set the SCR.TE or RE bit to 1, and set the SCR.TIE and RIE bits	Set the SCR.TE or RE bit to 1. Also set the SCR.TIE and RIE bits. Setting the TE and RE bits allows TXDn and RXDn pins to be used.
11	Initialization completion	

Table 24.26 Example flow of SCI initialization in asynchronous mode with FIFO selected (1 of 2)

No.	Step Name	Description
1	Start initialization	
2	Set the SCR.TIE, RIE, TE, RE, and TEIE bits to 0	
3	Set the FCR.FM, TFRST, and RFRST bits to 1. Set the FCR.TTRG[3:0], RTRG[3:0], and RSTRG[3:0] bits.	Set the FCR.FM, TFRST, and RFRST bits to 1 (FIFO mode enabled, transmit/receive FIFOs empty). Set the FCR.TTRG[3:0], RTRG[3:0], and RSTRG[3:0] bits.
4	Set the SCR.CKE[1:0] bits	Set the clock selection in SCR. When the clock output is selected in asynchronous mode, the clock is output immediately after SCR settings are made.
5	Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits to 0.	Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits to 0. Step 5 can be skipped if the values have not been changed from the initial values.
6	Set the data transmission/ reception format in SMR, SCMR, and SEMR	Set data transmission/reception format in SMR, SCMR, and SEMR.
7	Set a value in BRR	Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.
8	Set a value in MDDR	Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is set to 0 or an external clock is used.

Table 24.26 Example flow of SCI initialization in asynchronous mode with FIFO selected (2 of 2)

No.	Step Name	Description
9	Set the FCR.TFRST and RFRST bits to 0	Set the FCR.TFRST and RFRST bits to 0.
10	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
11	Set the SCR.TE or RE bit to 1, and set the SCR.TIE and RIE bits	Set the SCR.TE or RE bit to 1. Also set the SCR.TIE and RIE bits. Setting the TE and RE bits allows TXDn and RXDn to be used.
12	Initialization completion	

24.3.8 Serial Data Transmission in Asynchronous Mode

(1) Non-FIFO selected

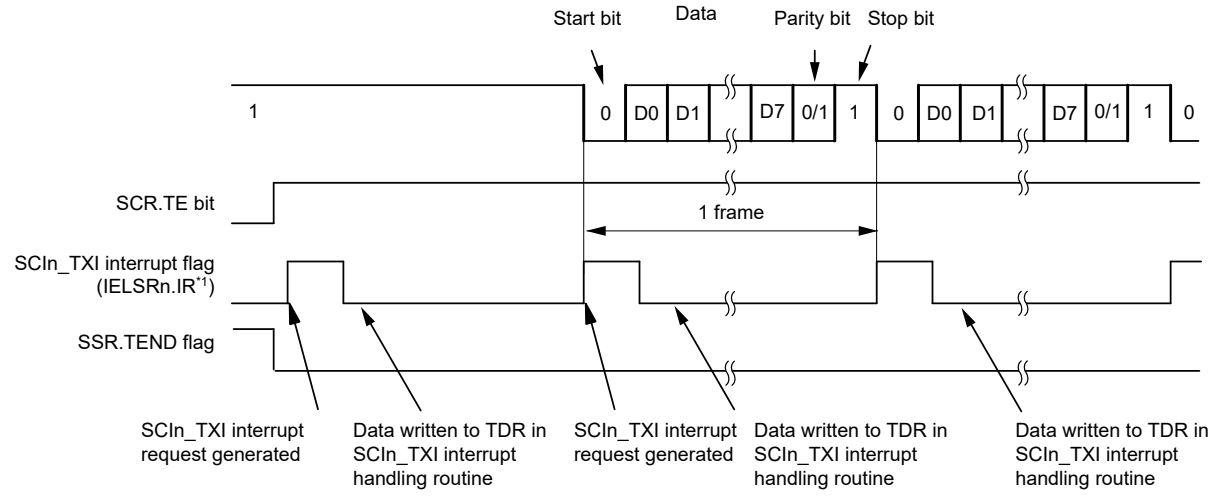
Figure 24.7, Figure 24.8, and Figure 24.9 show examples of serial transmission in asynchronous mode.

In serial transmission, the SCI operates as described in this section. When the SCR.TE bit is set to 1, the high level is output to TXDn for one frame.

1. The SCI transfers data from the TDR[*1] register to the TSR register when data is written to TDR[*1] in the SCIn_TXI interrupt handling routine.
   The SCIn_TXI interrupt request at the beginning of transmission is generated when the SCR.TE and SCR.TIE bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the SPMR.CTSE bit is set to 0 (CTS function is disabled) or a low level on the CTSn_RTSn pin causes data transfer from the TDR[*1] register to the TSR register. If the SCR.TIE bit is 1, an SCIn_TXI interrupt request is generated. Continuous transmission is possible by writing the next transmit data to the TDR[*1] register in the SCIn_TXI interrupt handling routine before transmission of the current transmit data is complete. When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 (SCIn_TXI interrupt requests are disabled) and the SCR.TEIE bit to 1 (an SCIn_TEI interrupt request is enabled) after the last of the data to be transmitted is written to the TDR[*1] register from the handling routine for SCIn_TXI requests.
3. Data is sent from the TXDn pin in the following order:
   - Start bit
   - Transmit data
   - Parity bit or multi-processor bit (can be omitted depending on the format)
   - Stop bit
4. The SCI checks for update of the TDR register on output of the stop bit.
5. When the TDR register is updated, setting the SPMR.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn_RTSn pin causes transfer of the next transmit data from the TDR[*1] register to the TSR register and transmission of the stop bit, after which serial transmission of the next frame starts.
6. If the TDR register is not updated, the SSR.TEND flag is set to 1, the stop bit is sent, and the mark state is entered, in which 1 is output. If the SCR.TEIE bit is 1, the SSR.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated.

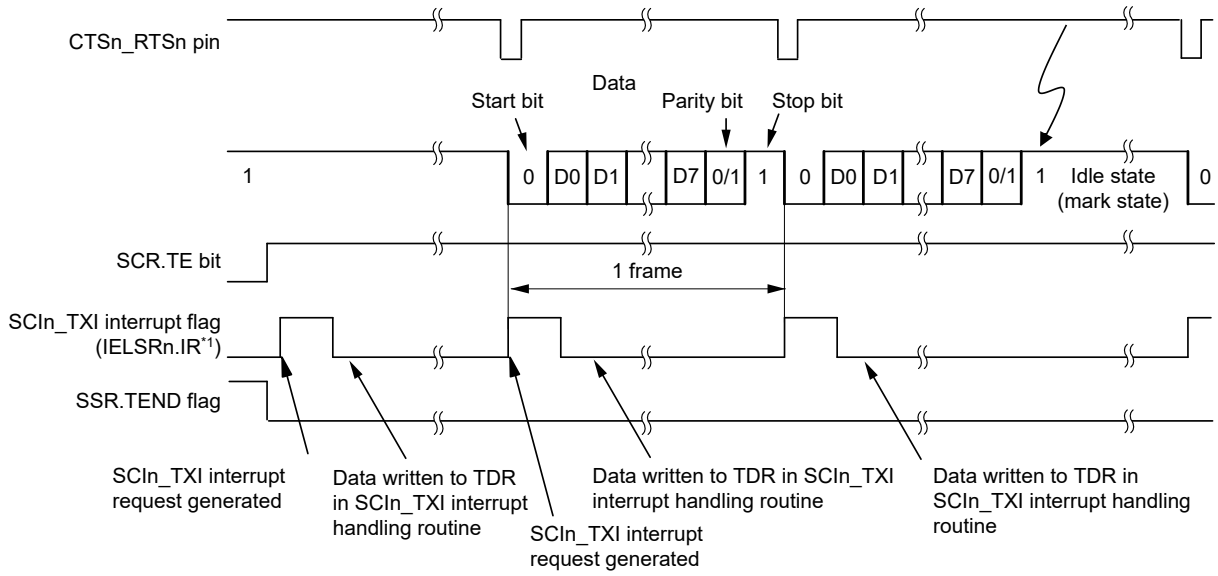
Note 1. The TDRHL register when 9-bit data length is selected.

Figure 24.7, Figure 24.8, and Figure 24.9 show examples of serial transmission in asynchronous mode.



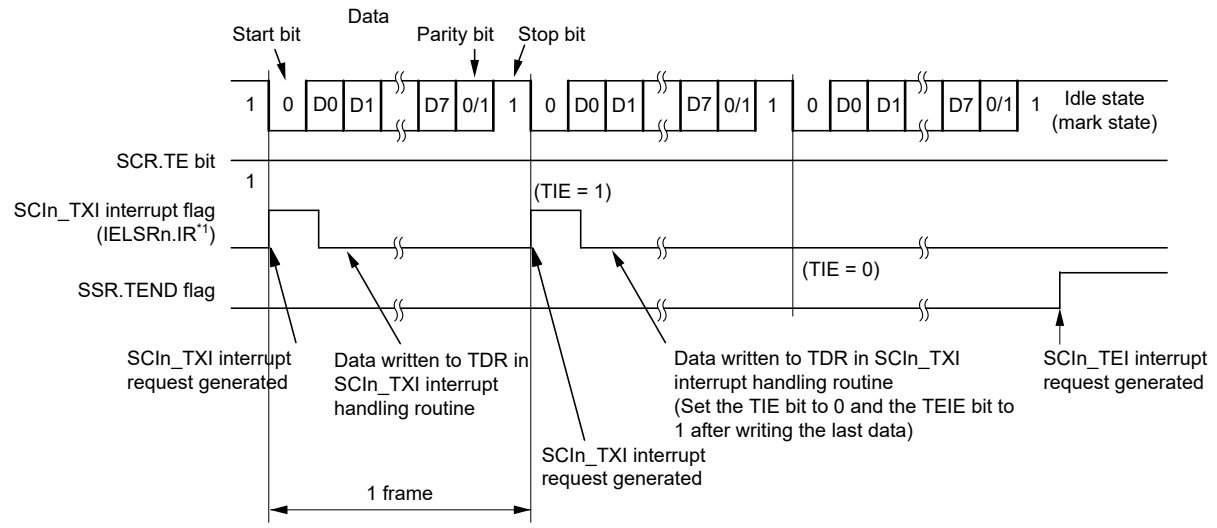
Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 24.7 Example operation for serial transmission in asynchronous mode (1) with 8-bit data, parity bit, 1 stop bit, CTS function not used, and at the beginning of transmission



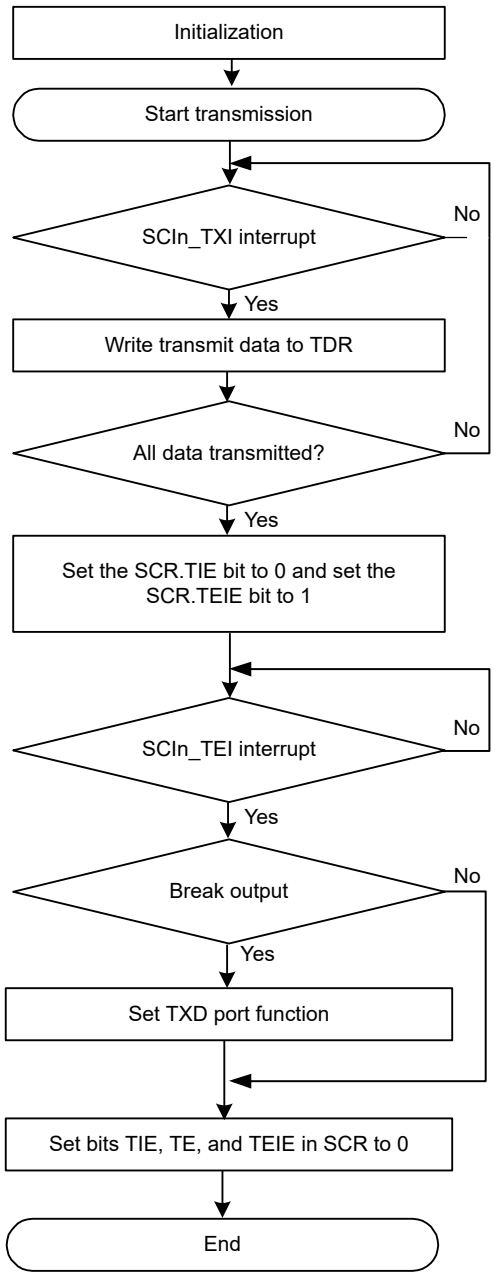
Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 24.8 Example operation for serial transmission in asynchronous mode (2) with 8-bit data, parity bit, one stop bit, CTS function used, and at the beginning of transmission



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 24.9 Example operation for serial transmission in asynchronous mode (3) with 8-bit data, parity bit, one stop bit, CTS function not used, and from the middle of transmission until transmission completion



[1] [1] SCI Initialization:
Set data transmission.
After the SCR.TE bit is set to 1, 1 is output for a frame, and transmission is enabled.

[2] [2] Transmit data write to TDR by an SCIn_TXI interrupt request. When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (SCIn_TXI) request is generated. Write transmit data to TDR once in the SCIn_TXI interrupt processing routine.

[3] [3] Serial transmission continuation procedure:
To continue serial transmission write transmit data to TDR once using an SCIn_TXI interrupt request. Transmit data can also be written to TDR by activating the DTC.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to TDR.

[4] [4] Break output at the end of serial transmission:
To output a break in serial transmission, after setting the output state (LOW level output) of TXD pin by SPTR.SPB2IO and SPTR.SPB2DT bits, set the SCR.TE bit to 0.

Figure 24.10 Example flow of serial transmission in asynchronous mode with non-FIFO selected

(2) FIFO selected

Figure 24.11 shows an example of a data format that is written to FTDRH and FTDRL register in asynchronous mode.

Data corresponding to the data length is set to FTDRH and FTDRL. Write 0 for unused bits. Write in order from FTDRH to FTDRL.

Data Length	Register Setting		Transmit data in FTDRH, FTDRL															
			FTDRHL															
			FTDRH								FTDRL							
	SCMR.CHR1	SMR.CHR	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	0	–	–	–	–	–	–	–	–	–	7-bit transmit data						
8 bits	1	1	–	–	–	–	–	–	–	–	8-bit transmit data							
9 bits	0	Don't care	–	–	–	–	–	–	–	9-bit transmit data								

Note: —: Invalid. The write value should be 0.

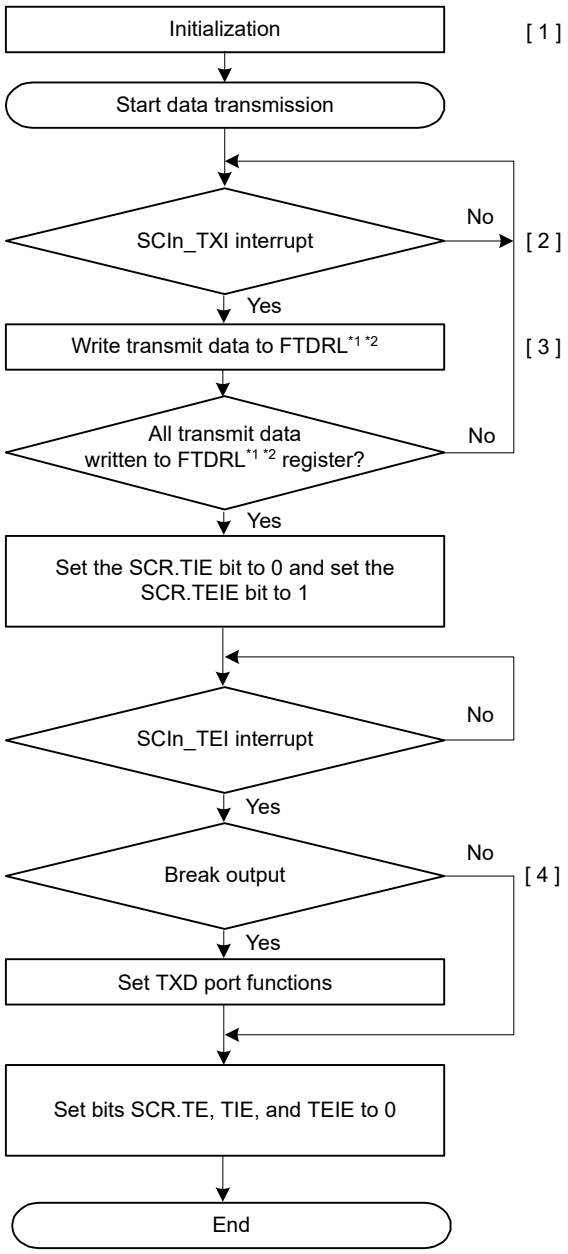
Figure 24.11 Data format written to FTDRH and FTDRL with FIFO selected

In serial transmission, the SCI operates as described in this section. When the TE bit is set to 1, the high level is output to TXDn for one frame (preamble).

1. The SCI transfers data from the FTDRL[*1] register to the TSR register when data is written to FTDRL[*1] in the SCIn_TXI interrupt handling routine. The amount of data that can be written to FTDRL is 16 minus FDR.T[4:0] bytes. The SCIn_TXI interrupt request at the beginning of transmission is generated when the SCR.TE and SCR.TIE bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the SPMR.CTSE bit is set to 0 (CTS function is disabled) or a low level on the CTSn_RTSn pin causes data transfer from the FTDRL[*1] register to the TSR register. When the amount of transmit data written in FTDRL is equal to or less than the specified transmit triggering number, SSR_FIFO.TDFE is set to 1. If the SCR.TIE bit is 1, an SCIn_TXI interrupt request is generated. Continuous transmission is possible by writing the next transmit data to FTDRL[*1] in the SCIn_TXI interrupt handling routine before transmission of the current transmit data is complete. When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 (SCIn_TXI interrupt requests are disabled) and the SCR.TEIE bit to 1 (an SCIn_TEI interrupt request is enabled) after the last of the data to be transmitted is written to the FTDRL[*1*2] register from the handling routine for SCIn_TXI requests.
3. Data is sent from the TXDn pin in the following order:
   - Start bit
   - Transmit data
   - Parity bit or multi-processor bit (can be omitted depending on the format)
   - Stop bit
4. On output of the stop bit, the SCI checks whether non-transmitted data remains in the FTDRL[*3] register.
5. When data is set to FTDRL[*3], setting the SPMR.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn_RTSn pin causes transfer of the next transmit data from FTDRL[*1] to TSR and transmission of the stop bit, after which serial transmission of the next frame starts.
6. If data is not set in FTDRL[*3], the TEND flag in SSR_FIFO is set to 1, the stop bit is sent, and the mark state is entered in which 1 is output. If the SCR.TEIE bit is 1, the SSR_FIFO.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated.

Note 1. Write data not to FTDRL but to the FTDRH and FTDRL registers.
Note 2. Write data in order from FTDRH to FTDRL when 9-bit data length is selected.
Note 3. The SCI only checks for update to the FTDRL register and not the FTDRH register when 9-bit data length is selected.

Figure 24.12 shows an example flow of serial transmission in asynchronous mode with FIFO selected.



[1] SCI initialization:
Set data transmission.
After the SCR.TE bit is set to 1, 1 is output for a frame, and transmission is enabled.

[2] Transmit data write to FTDRL*1 by an SCIn_TXI interrupt request:
When transmit data is transferred from FTDRL to TSR, when the quantity of transmit data written in FTDRL is equal to or less than the specified transmit triggering number, a transmit data FIFO empty interrupt (SCIn_TXI) request is generated. Write transmit data to FTDRL*1 *2 once in the SCIn_TXI interrupt handling routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data to FTDRL using an SCIn_TXI interrupt request and clear the SSR_FIFO.TDFE flag to 0. The number of transmission data it is possible to write in is 16 - (the number of stored transmit FIFO data).
Transmit data can also be written to FTDRL by activating the DTC. When writing the data to FTDRL by DTC, the TDFE flag is cleared automatically, so do not write to the TDFE flag.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to FTDRL.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, after setting the output state (LOW level output) of TXD pin by SPTR.SPB2IO and SPTR.SPB2DT bits, set the SCR.TE bit to 0.

Note 1. When data length is 9 bits, this refers to FTDRH and FTDRL registers.
Note 2. When data length is 9 bits, write in order from FTDRH to FTDRL.

Figure 24.12 Example flow of serial transmission in asynchronous mode with FIFO selected

24.3.9 Serial Data Reception in Asynchronous Mode

(1) Non-FIFO selected

Figure 24.13 and Figure 24.14 show an example of the operation for serial data reception in asynchronous mode.

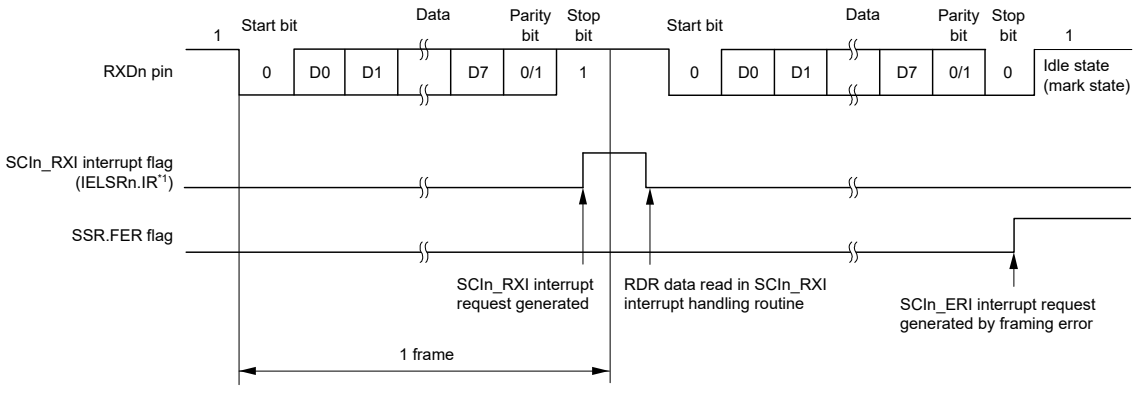
In serial data reception, the SCI operates as follows:

1. When the value of the SCR.RE bit becomes 1, the output signal on the CTSn_RTSn pin goes low.
2. The SCI monitors the communications line and when it detects a start bit, the SCI performs internal synchronization, stores receive data in RSR.
3. If the multi-processor communication function is enabled (SMR.MP = 1), see section 24.4.2. Multi-Processor Serial Data Reception. If the address match function (data compare match function) is enabled (DCCR.DCME = 1), the SCI

cannot detect a parity or framing error as receive data are skipped (discarded) until the SCI detects a match between the receive data and comparison data (CDR.CMPD[*1]).

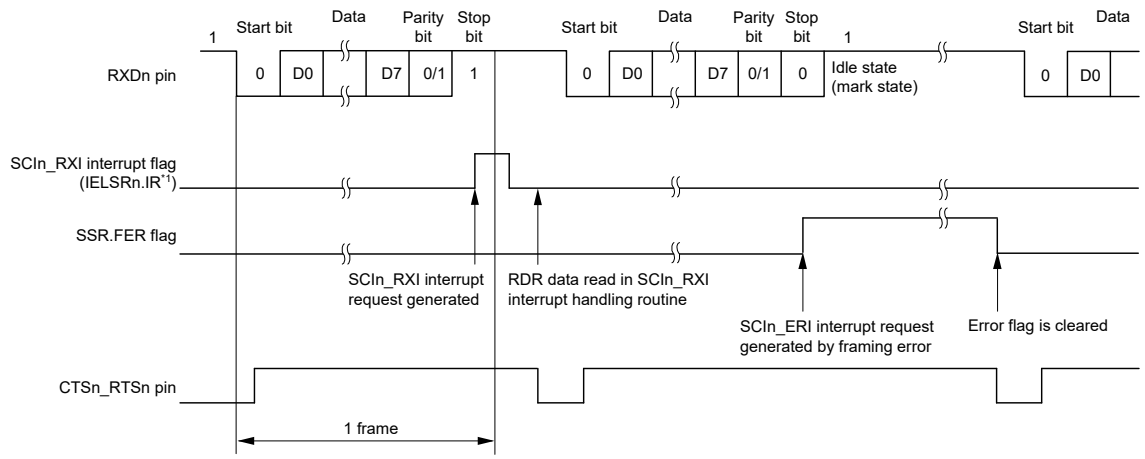
4. If the SCI detects an address match, the DCCR.DCME bit is automatically cleared, the DCCR.DCMF flag becomes 1, and an SCIn_AM interrupt[*2] request is generated. To enable the generation of an SCIn_RXI interrupt request, set the SCR.RIE bit to 1. The compared receive data are not stored in the RDR register[*3]. The SSR.RDRF flag remains 0.
5. If the SCI detects a framing error in the receive data for which an address match is detected, the DCCR.DFER flag is set to 1, and if the SCI detects a parity error in that frame, the DCCR.DPER flag becomes 1. To enable the generation of an SCIn_ERI interrupt request, set the SCR.RIE bit to 1.
6. If a framing or a parity error is detected (the DCCR.DFER flag or DCCR.DPER flag is 1) in the SCIn_AM interrupt handling routine, set the DCCR.DFER and DCCR.DPER flags to 0 and set the DCCR.DCME bit to 1 to enable the address match function again. If neither a framing nor a parity error has been detected (the DCCR.DFER and DCCR.DPER flags are both 0), set the DCCR.DCMF flag to 0. See Figure 24.5.
7. If an overrun error occurs, the SSR.ORER flag is set to 1. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the RDR[*3] register.
8. If a parity error is detected, the SSR.PER flag is set to 1 and receive data is transferred to the RDR[*3] register. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated.
9. If a framing error is detected, the SSR.FER flag is set to 1 and receive data is transferred to the RDR[*3] register. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated.
10. When reception finishes successfully, receive data is transferred to the RDR[*3] register. If the SCR.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in the SCIn_RXI interrupt handling routine before reception of the next receive data is complete. Reading the received data that was transferred to the RDR register causes the CTSn_RTSn pin to output low.

Note 1. This scope of comparison is selectable as one of three lengths: CMPD[6:0] is for 7-bit length, CMPD[7:0] is for 8-bit length, and CMPD[8:0] is for 9-bit length.
Note 2. As no interrupt enable bit is assigned to the SCIn_AM interrupt, an interrupt request is generated by setting the DCCR.DCMF to 1.
Note 3. Only read data in the RDRHL register when 9-bit data length is selected.



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 24.13 Example of SCI operation for serial reception in asynchronous mode (1) when the RTS function is not used, and with 8-bit data, parity bit, and 1 stop bit



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 24.14 Example of SCI operation for serial reception in asynchronous mode (2) when RTS function is used, and with 8-bit data, parity bit, and 1 stop bit

Table 24.27 lists the states of the flags in the SSR register and receive data handling when a receive error is detected.

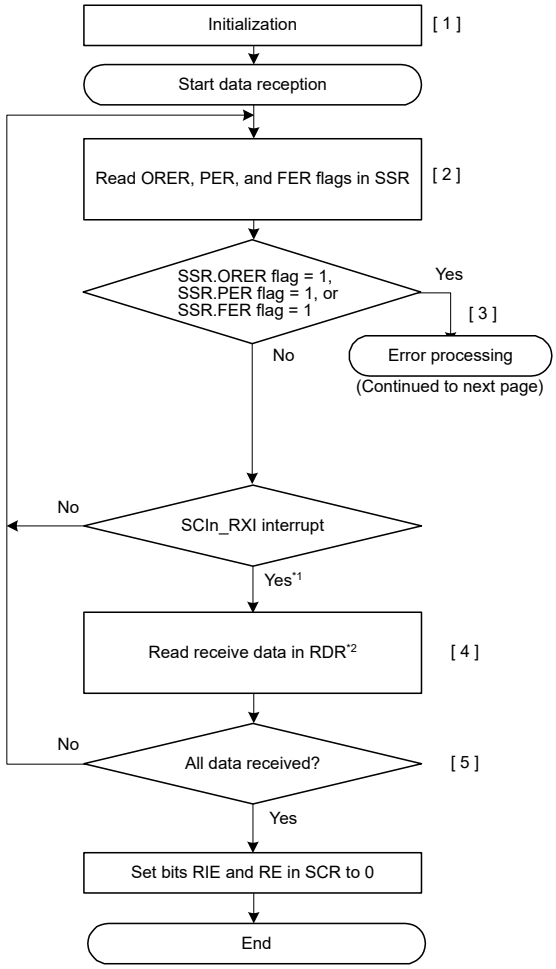
If a receive error is detected, an SCIn_ERI interrupt request is generated but an SCIn_RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER bits to 0 before resuming reception. In addition, be sure to read the RDR or RDRHL register during overrun error processing. When a reception is forced to terminate by setting the SCR.RE bit to 0 during operation, read the RDR or RDRHL register because received data that is not yet read might be left in the RDR or RDRHL.

Figure 24.15 and Figure 24.16 show example flows of serial data reception.

Table 24.27 Flags in SSR Status Register and receive data handling

Flags in the SSR Status Register			Receive data	Receive error type
ORER	FER	PER		
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR*1	Framing error
0	0	1	Transferred to RDR*1	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR*1	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error

Note 1. Only read data in the RDRHL register when 9-bit data length is selected.



[1]SCI initialization:
Set data reception.

[2][3]Receive error processing and break detection:
If a receive error occurs, an SCIn_ERI interrupt is generated. The error type is identified by reading the ORER, PER, and FER flags in SSR. After performing the appropriate error processing, be sure to set the ORER, PER, and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. For a framing error, a break can be detected by reading the value of the input port associated with the RXDn pin.

[4]Read the receive data in RDR once in the SCIn_RXI interrupt handling routine.

[5]To continue serial reception, before the stop bit of the current frame is received, read data from RDR in the SCIn_RXI interrupt processing routine. The RDR data can also be read by activating the DTC.

Note 1. Do not set RE to 0 before reading RDR.
Note 2. The RDR register becomes the RDRHL register when 9-bit data length is selected.

Figure 24.15 Example flow of serial reception in asynchronous mode with non-FIFO selected and Address Matching Disabled (1)

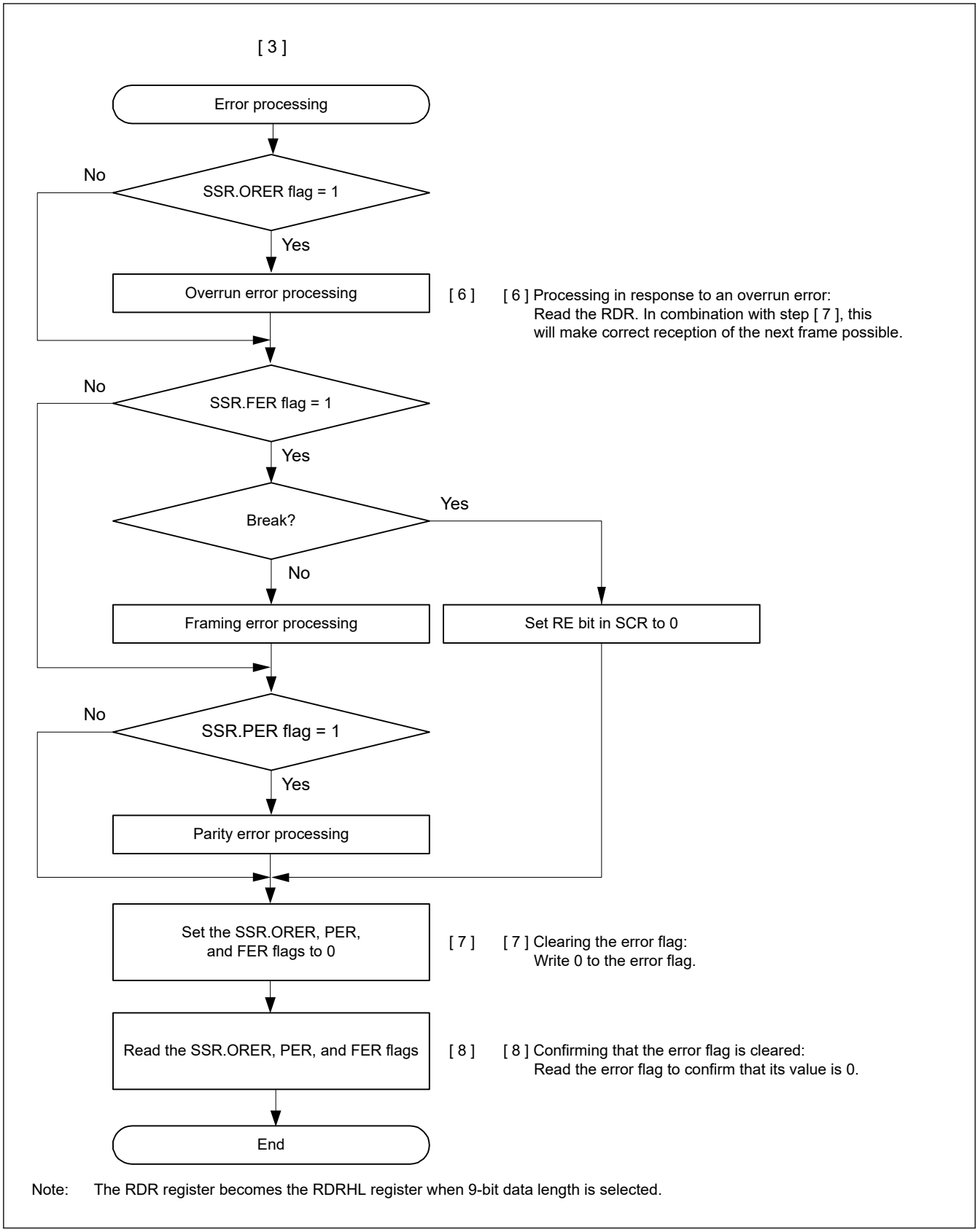
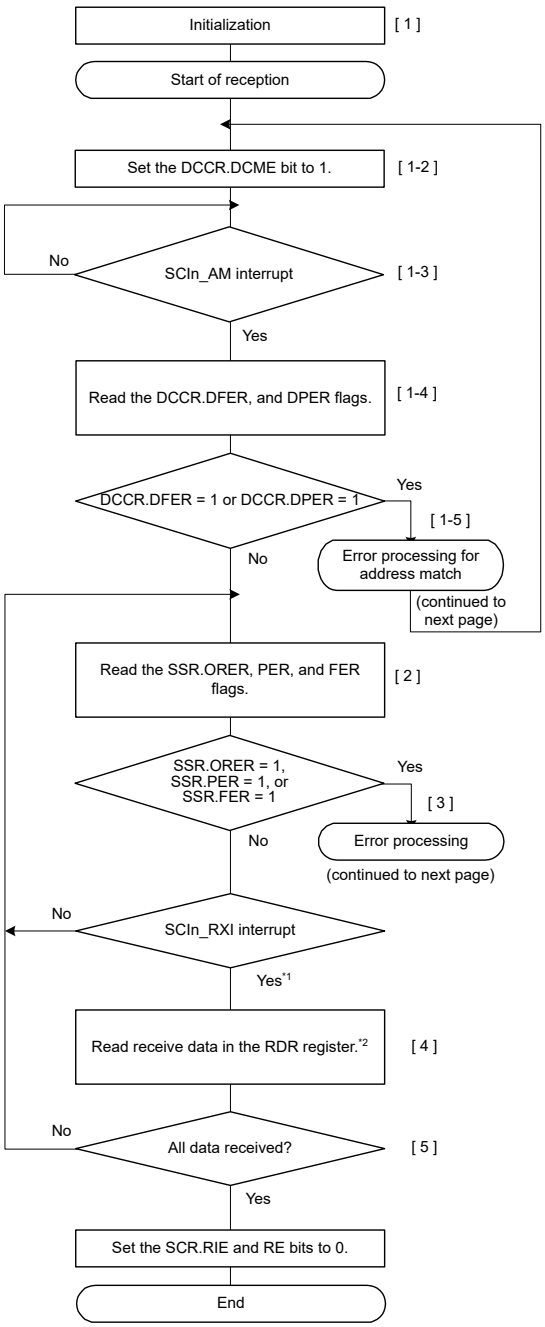


Figure 24.16 Example flow of serial reception in asynchronous mode with non-FIFO selected and Address Matching Disabled (2)



[1] SCI initialization:
Set data reception.

[1-2] Address match cycle:
Set the DCCR.DCME bit to 1 to enable address matching.

[1-3] Comparing receive data and data for comparison (CDR.CMPD):
Compare receive data and data for comparison (CDR.CMPD).
The SCI cannot detect a parity or framing error as receive data are skipped (discarded) until the SCI detects a match between the receive data and data for comparison.
If the SCI detects an address match, the DCCR.DCME bit is cleared, the DCCR.DCMF flag becomes 1, and an SCIn_AM interrupt request is issued.

[1-4] [1-5] Reception error processing:
An error is identified by reading the DFER or DPER flag in DCCR in the SCIn_AM interrupt handling routine.
After appropriate handling of the error, set both the DFER and DPER flags to 0.
If either of these flags is set to 1, address matching cannot be resumed.

[2] [3] Reception error processing and break detection:
When a reception error occurs, the SCIn_ERI interrupt is generated.
The error is identified by reading the ORER, PER, and FER flags in the SSR.
After appropriate handling of the error, set all of the ORER, PER, and FER flags to 0. If any of these flags is set to 1, reception cannot be resumed.
If a framing error occurs, a break can be detected by reading the value of the input port bit corresponding to the RXDn pin.

[4] Read the receive data in the RDR register from within the SCIn_RXI interrupt handling routine.

[5] Procedure for continuing serial reception:
To continue serial reception, read the value of the RDR register in the SCIn_RXI interrupt handling routine before the stop bit is received.
The value of the RDR register can also be read by activating the DTC.

Note 1. Be sure to set the RE bit to 0 before reading the RDR register.
Note 2. The RDRHL register is used instead of the RDR register when the 9-bit data length is selected.

Figure 24.17 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO not Selected and Address Matching Enabled) (1)

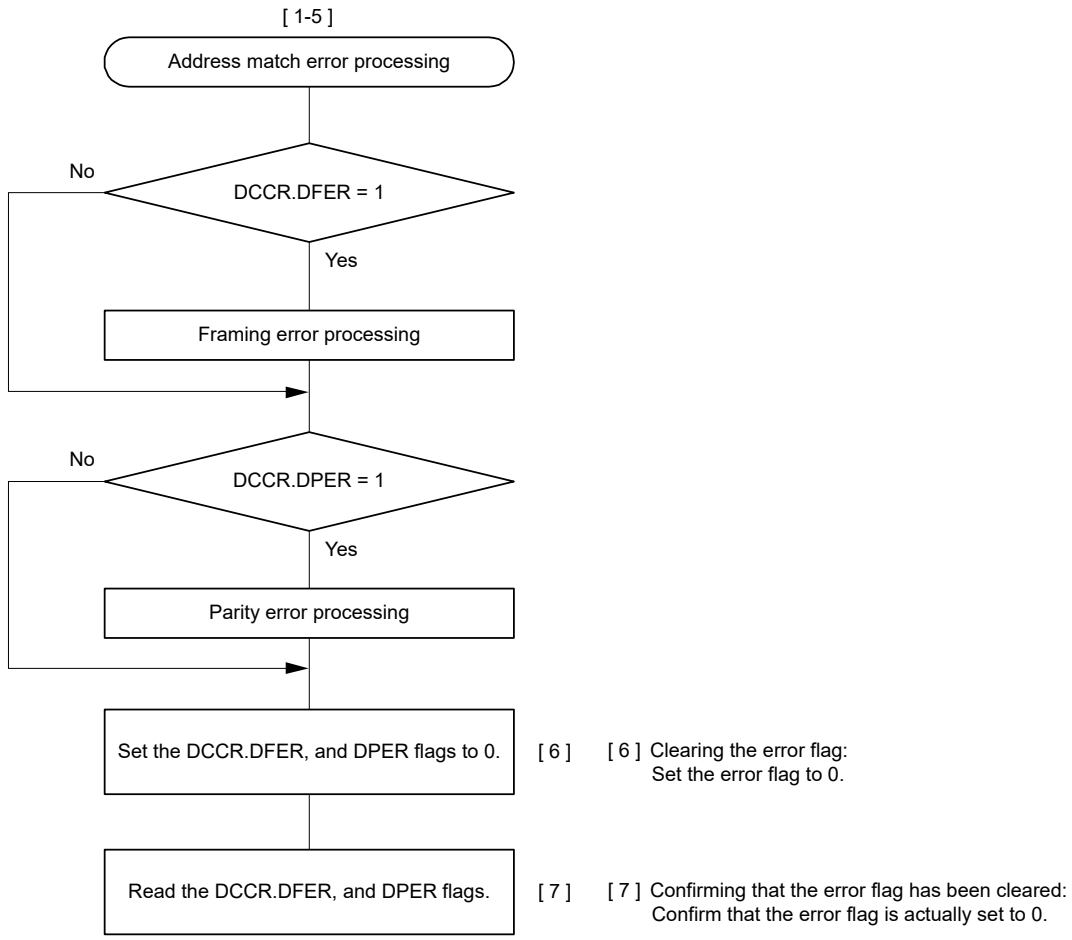
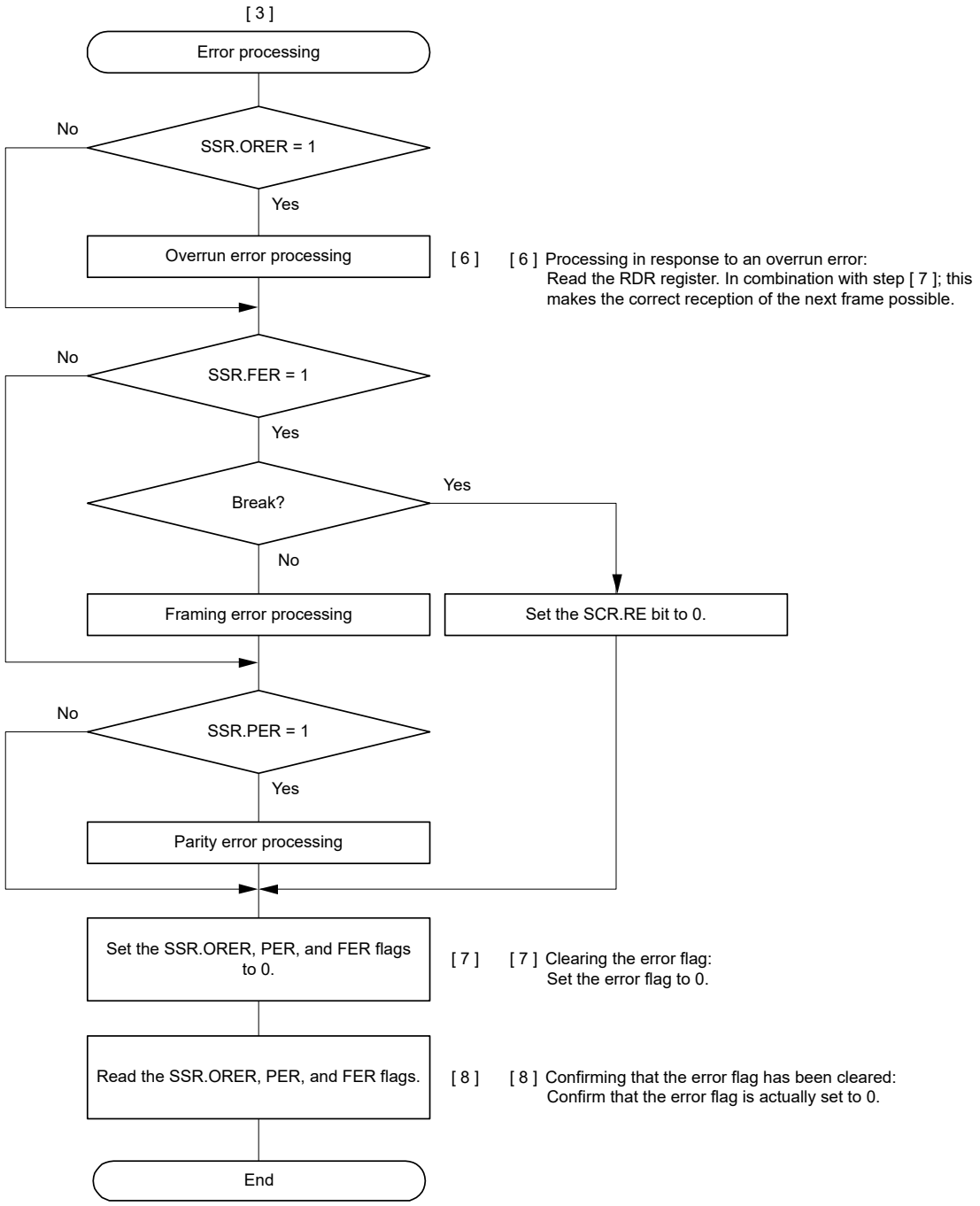


Figure 24.18 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO not Selected and Address Matching Enabled) (2)



Note: The RDRHL register is used instead of the RDR register when the 9-bit data length is selected.

Figure 24.19 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO not Selected and Address Matching Enabled) (3)

(2) FIFO selected

Figure 24.20 shows an example of a data format that is written to FRDRH register and FRDRL register in asynchronous mode.

In asynchronous mode, 0 is written to the MPB bit in the FRDRH register. Data that corresponds to the data length is written to FRDRH and FRDRL. Unused bits are written as 0. Read in order from FRDRH to FRDRL. If software reads FRDRL, the

SCI updates FER, PER, and receive data (RDAT[8:0]) in the FRDRL register with the next data. The flags RDF, ORER, and DR in the FRDRH register always reflect the associated flags in the SSR_FIFO register.

Data Length	Register Setting		Receive data in FRDRH, FRDRL															
			FRDRHL															
			FRDRH								FRDRL							
	SCMR.CHR1	SMR.CHR	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	0	-	RDF	ORER	FER	PER	DR	0	0	0	7-bit receive data						
8 bits	1	1	-	RDF	ORER	FER	PER	DR	0	0	8-bit receive data							
9 bits	0	Don't care	-	RDF	ORER	FER	PER	DR	0	9-bit receive data								

Note: 0 is always read for MPB bit (FRDRH[1])
When data length is 7 bits, 0 is always read for FRDRH[0] and FRDRL[7]
When data length is 8 bits, 0 is always read for FRDRH[0]
FRDRH[7] bit is read as an indefinite value

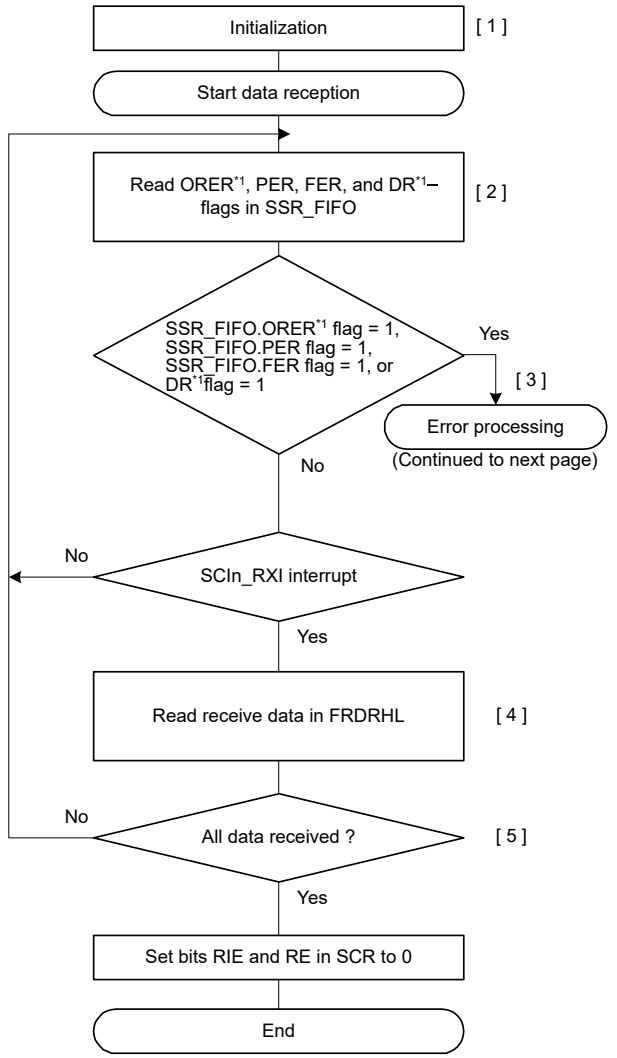
Figure 24.20 Data format stored in FRDRH and FRDRL with FIFO selected

In serial data reception, the SCI operates as follows:

1. When the value of the SCR.RE bit becomes 1, the output signal on the CTSn_RTSn pin goes low.
2. The SCI monitors the communications line and, when it detects a start bit, the SCI performs internal synchronization, stores receive data in the RSR register.
3. If the multi-processor communications function is enabled (SMR.MP = 1), see section 24.4.2. Multi-Processor Serial Data Reception. If the address match function (data compare match function) is enabled (DCCR.DCME = 1), the SCI cannot detect a parity or framing error as receive data are skipped (discarded) until the SCI detects a match between the receive data and the data for comparison (CDR.CMPD[*1]).
4. If the SCI detects an address match, the DCCR.DCME bit is automatically cleared, the DCCR.DCMF flag becomes 1, and an SCIn_AM interrupt[*2] request is generated. To enable the generation of an SCIn_RXI interrupt request, set the SCR.RIE bit to 1. The compared receive data are not stored in the RDR register[*3]. The SSR.RDRF flag remains 0.
5. If the SCI detects a framing error in the receive data for which an address match was detected, the DCCR.DFER flag is set to 1, and if the SCI detects a parity error in that frame, the DCCR.DPER flag becomes 1. To enable the generation of an SCIn_ERI interrupt request, set the SCR.RIE bit to 1.
6. If a framing or a parity error is detected (the DCCR.DFER flag or DCCR.DPER flag is 1) in the SCIn_AM interrupt handling routine, set the DCCR.DFER and DCCR.DPER flags to 0 and set the DCCR.DCME bit to 1 to enable the address match function again. If neither a framing nor a parity error has been detected (the DCCR.DFER and DCCR.DPER flags are 0), set the DCCR.DCMF flag to 0. See Figure 24.5.
7. If an overrun error occurs during normal communications, the SSR_FIFO.ORER flag is set to 1. If the SCR.RIE bit in SCR is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the FRDRL[*3] register.
8. If a parity error is detected, the PER flag and receive data are transferred to the FRDRL[*3] register. If the SCR.RIE bit is set to 1, an SCIn_ERI interrupt request is generated.
9. If a framing error is detected, the FER flag and receive data are transferred to the FRDRL[*3] register. If the SCR.RIE bit is set to 1, an SCIn_ERI interrupt request is generated.
10. After a framing error is detected and when SCI detects that the continuous receive data is zero for one frame, reception stops.
11. When the amount of data stored in the FRDRL register falls below the specified receive triggering number, and the next data is not received after 15 etus from the last stop bit in asynchronous mode, the SSR_FIFO.DR flag is set to 1. When the SCR.RIE bit is 1 and the FCR.DRES bit is 0, the SCI generates an SCIn_RXI interrupt request. When the FCR.DRES bit is 1, SCI generates an SCIn_ERI interrupt request.

12. When reception finishes successfully, receive data is transferred to the FRDRL[*3] register. The RDF bit is set to 1 when the amount of receive data written to FRDRHL is equal to or greater than the specified receive triggering number. If the SCR.RIE bit in SCR is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the FRDRL[*4] register in the SCIn_RXI interrupt handling routine, before an overrun error occurs. If the received data that is transferred to FRDRL[*5] is less than the RTS trigger number, the CTSn_RTSn pin outputs low.

Note 1. One of three lengths is selected for the target for comparison: CMPD[6:0] is for 7-bit length, CMPD[7:0] is for 8-bit length, and CMPD[8:0] is for 9-bit length.
Note 2. As no interrupt enable bit is assigned to the SCIn_AM interrupt, an interrupt request is generated by setting the DCCR.DCMF to 1.
Note 3. Only read data in the FRDRH and FRDRL registers when 9-bit data length is selected.
Note 4. Read data in order from FRDRH to FRDRL when 9-bit data length is selected.
Note 5. The SCI only checks for update to the FRDRL register and not to the FRDRH register when 9-bit data length is selected.



[1] SCI initialization:
Set data reception.

[2] [3] Receive error processing and break detection:
If a receive error occurs, an SCIn_ERI interrupt is generated. A break can be detected by reading the SPTR.RXDMON flag. An error is identified by reading the ORER*1, PER, DR*1, and FER flags in SSR_FIFO. After performing the appropriate error processing, be sure to set the ORER*1 flag to 0. Reception cannot be resumed if ORER*1 flag is set to 1. The reception operation is continuous, even if the FER = 1 or PER = 1 or DR*1 = 1.

[4] Read the receive data in FRDRHL in the SCIn_RXI interrupt handling routine. The receive data stored in the FRDRHL register is read until the number of stored data is below the FCR.RTRG value. Confirm the number of the reception data in the FIFO by reading FDR.R.

[5] Serial reception continuation procedure:
To continue serial reception, before an overrun error occurs, read data from FRDRHL in the SCIn_RXI interrupt handling routine and clear RDF flag and DR flag to 0.
The FRDRHL data can also be read by activating the DTC. The RDF flag is cleared automatically in this case, so do not write to the RDF flag.

Note 1. Can be read in the FRDRHL.ORER and DR flags.

Figure 24.21 Example flow of serial reception in asynchronous mode with FIFO selected and Address Matching Enabled (1)

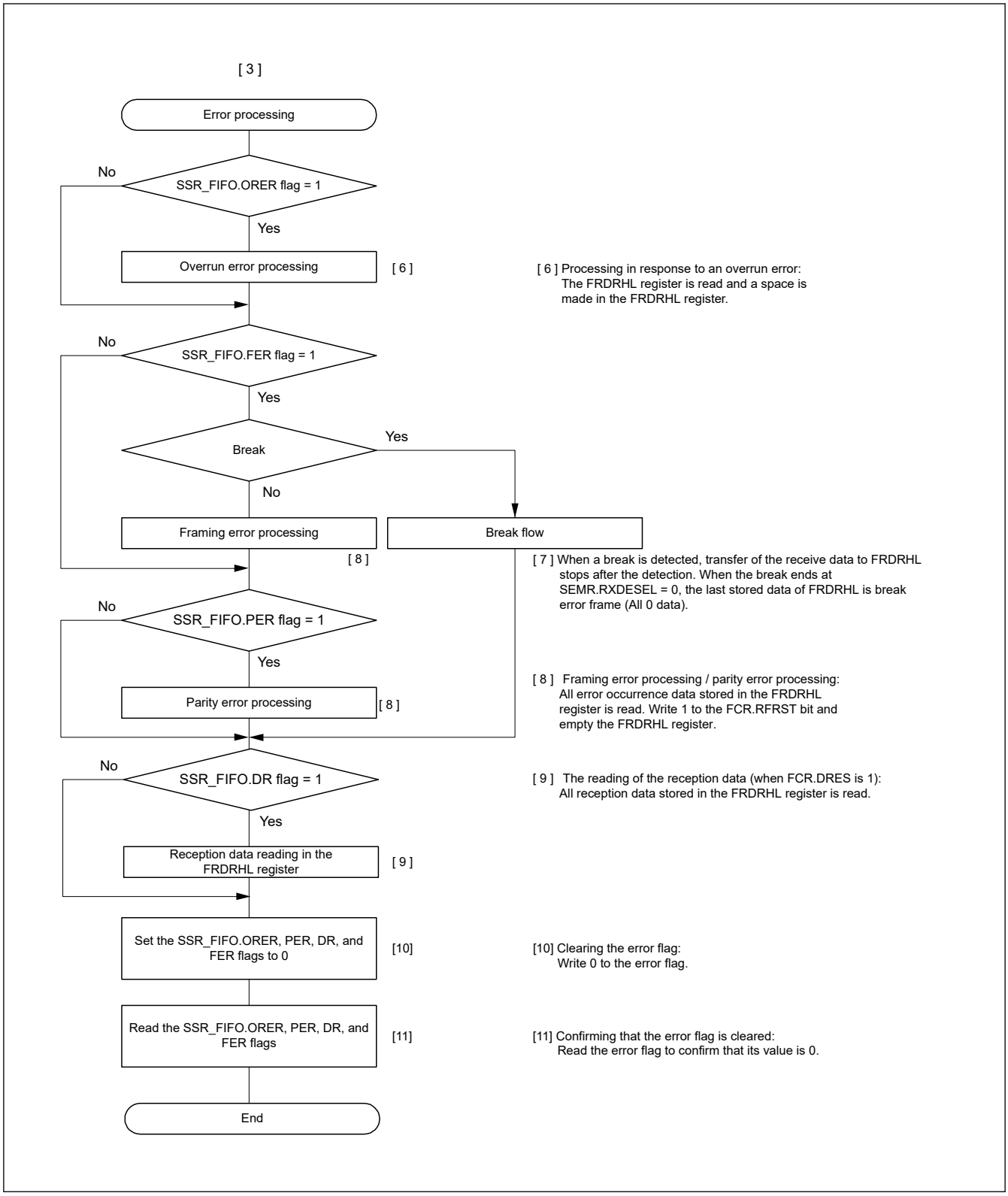
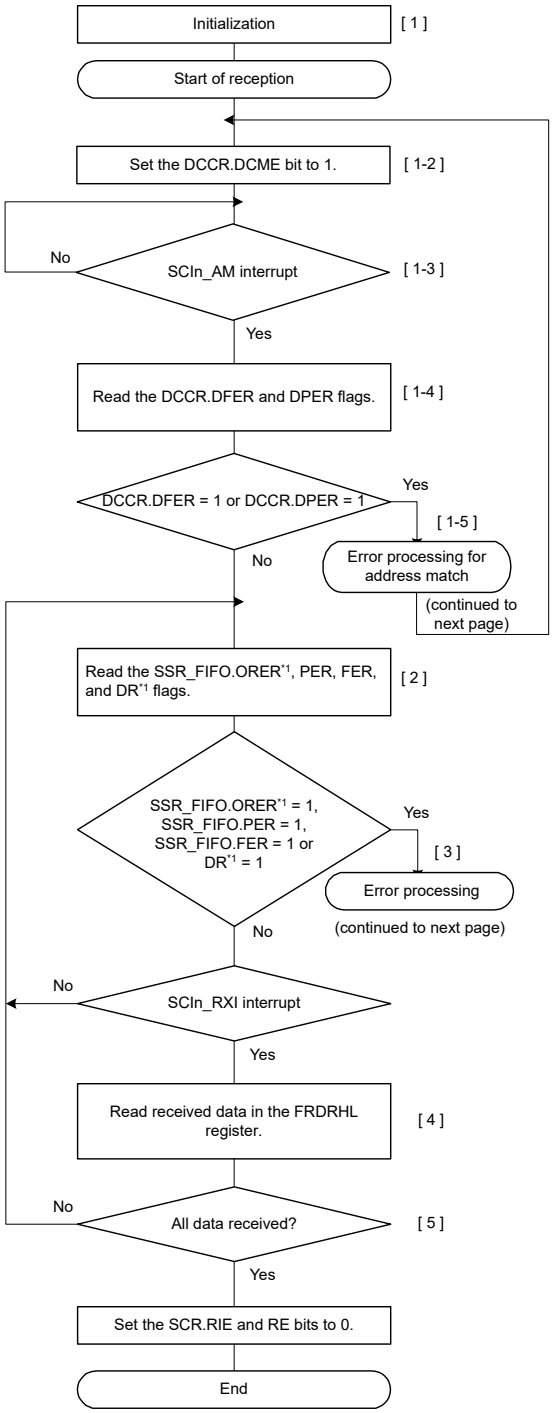


Figure 24.22 Example flow of serial reception in asynchronous mode with FIFO selected Address Matching Disabled (2)



[1] SCI initialization:
Set data reception.

[1-2] Address match cycle:
Set the DCCR.DCME bit to 1 to enable address matching.

[1-3] Comparing received data and data for comparison (CDR.CMPD):
Compare receive data and comparison data.
The SCI cannot detect a parity or framing error as received data are skipped (discarded) until the SCI detects a match between the received data and data for comparison.
If the SCI detects an address match, the DCCR.DCME bit is cleared, the DCCR.DCMF flag becomes 1, and an SCIn_AM interrupt request is issued.

[1-4] [1-5] Reception error processing:
An error is identified by reading the DFER or DPER flag in DCCR from within the SCIn_AM interrupt handling routine.
After appropriate handling of the error, set both the DFER and DPER flags to 0.
If either of these flags is set to 1, address matching cannot be resumed.

[2] [3] Reception error processing and break detection:
When a reception error occurs, an SCIn_ERI interrupt is generated.
A break is identified by reading SPTR.RXDMON.
The error is identified by reading each among the SSR_FIFO.ORER*1, PER, FER, and DR*1 flags.
After appropriate handling of the error, set the ORER*1 flag to 0.
If the ORER*1 flag remains 1, reception cannot be resumed.
The reception operation is continuous, even if the FER = 1, PER = 1 or DR*1 = 1.

[4] Reading the received data in the FRDRHL register from within the SCIn_RXI interrupt handling routine:
The received data stored in the FRDRHL register are read until the amount of stored data is less than the value of the FCR.RTRG[3:0] bits.
Confirm the number of received data in the FIFO with the value of the FDR.R[4:0] bits.

[5] Serial reception continuation procedure:
To continue with serial reception, read data from the FRDRHL register from within the SCIn_RXI interrupt handling routine before an overrun error occurs, and set the RDF and DR flags to 0. The data in the FRDRHL register can also be read by activating the DTC. In this case, do not write to the RDF flag because the flag is automatically cleared.

Note 1. The FRDRHL.ORER and DR flags can also be read.

Figure 24.23 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO Selected and Address Matching Enabled) (1)

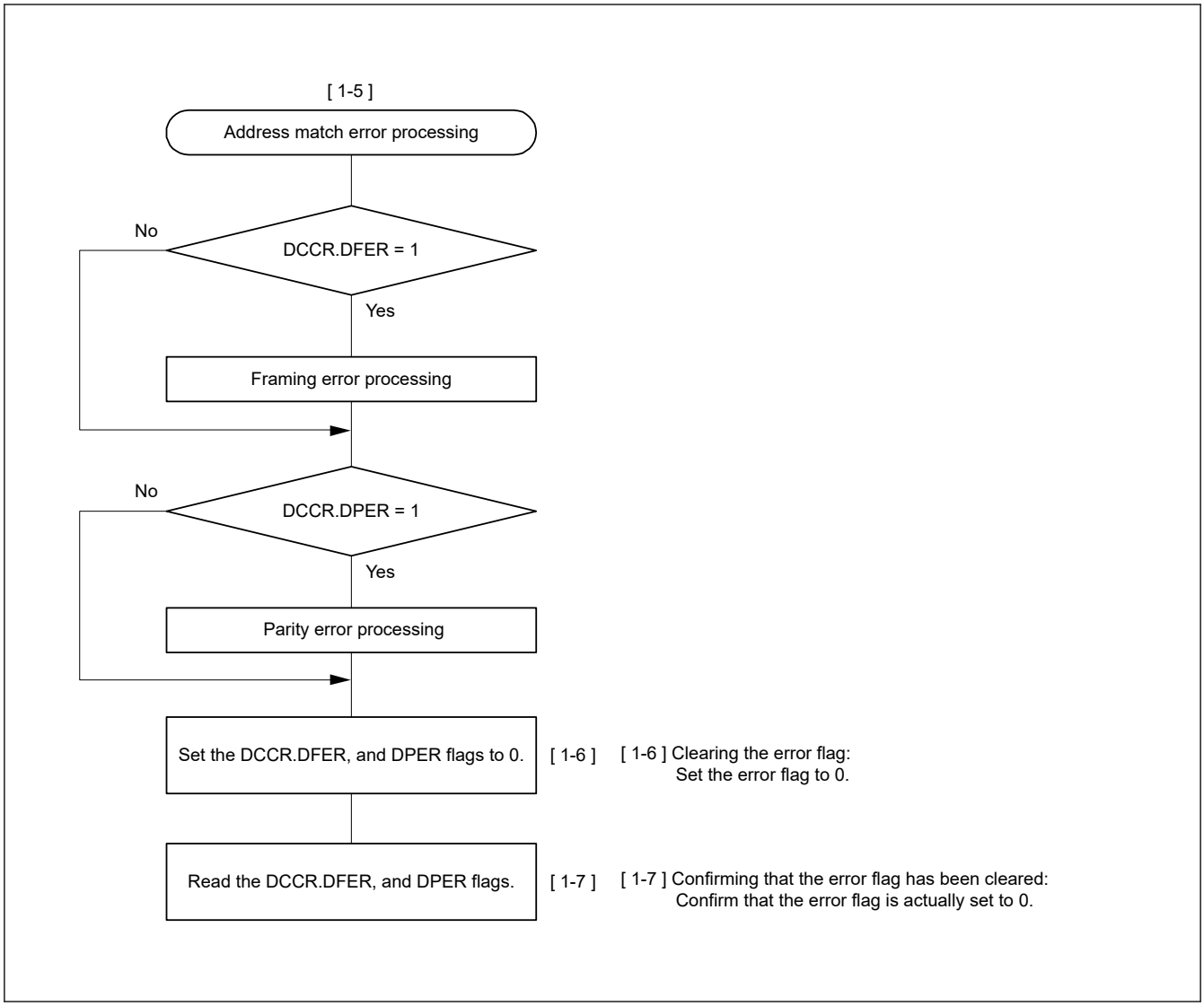


Figure 24.24 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO Selected and Address Matching Enabled) (2)

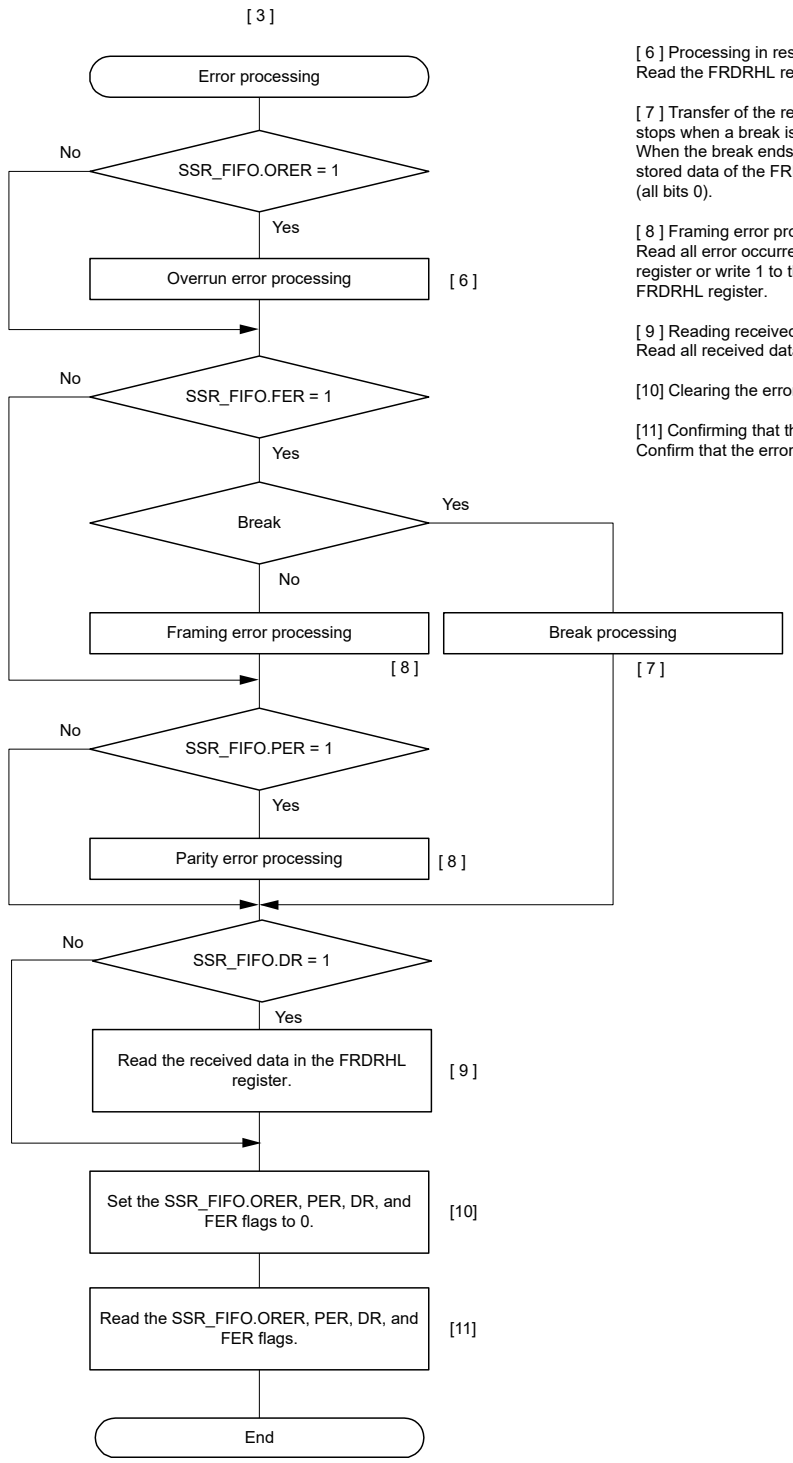


Figure 24.25 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO Selected and Address Matching Enabled) (3)

24.4 Multi-Processor Communication Function

The multi-processor communication function enables the SCI to transmit and receive data between multiple processors by sharing an asynchronous serial communication line that has an added multi-processor bit. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID

transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station.

The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle:

- When the multi-processor bit is set to 1, the transmission cycle is the ID transmission cycle
- When the multi-processor bit is set to 0, the transmission cycle is the data transmission cycle

Figure 24.26 shows an example of communication between processors using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits communication data in which the multi-processor bit set to 0 is added to the transmit data. After receiving communication data with the multi-processor bit set to 1, the receiving station compares the received ID with the ID of the receiving station itself. If the two match, the receiving station receives communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until it receives data in which the multi-processor bit is set to 1.

(1) Non-FIFO selected

To support this function, the SCI provides the SCR.MPIE bit. When the MPIE bit is set to 1, the following operations are disabled until the reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from the RSR register to the RDR register (the RDRHL register when 9-bit data length is selected)
- Detection of a receive error
- Setting of the respective RDRF, ORER, and FER status flags in the SSR register

When the SCI receives a character in which the multi-processor bit is set to 1, the SSR.MPBT bit is set to 1 and the SCR.MPIE bit is automatically cleared, returning the SCI to normal reception operation. If the SCR.RIE bit is set to 1, an SCIn_RXI interrupt is generated.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in normal asynchronous mode. The clock used for the multi-processor communication is the same as the clock used in normal asynchronous mode.

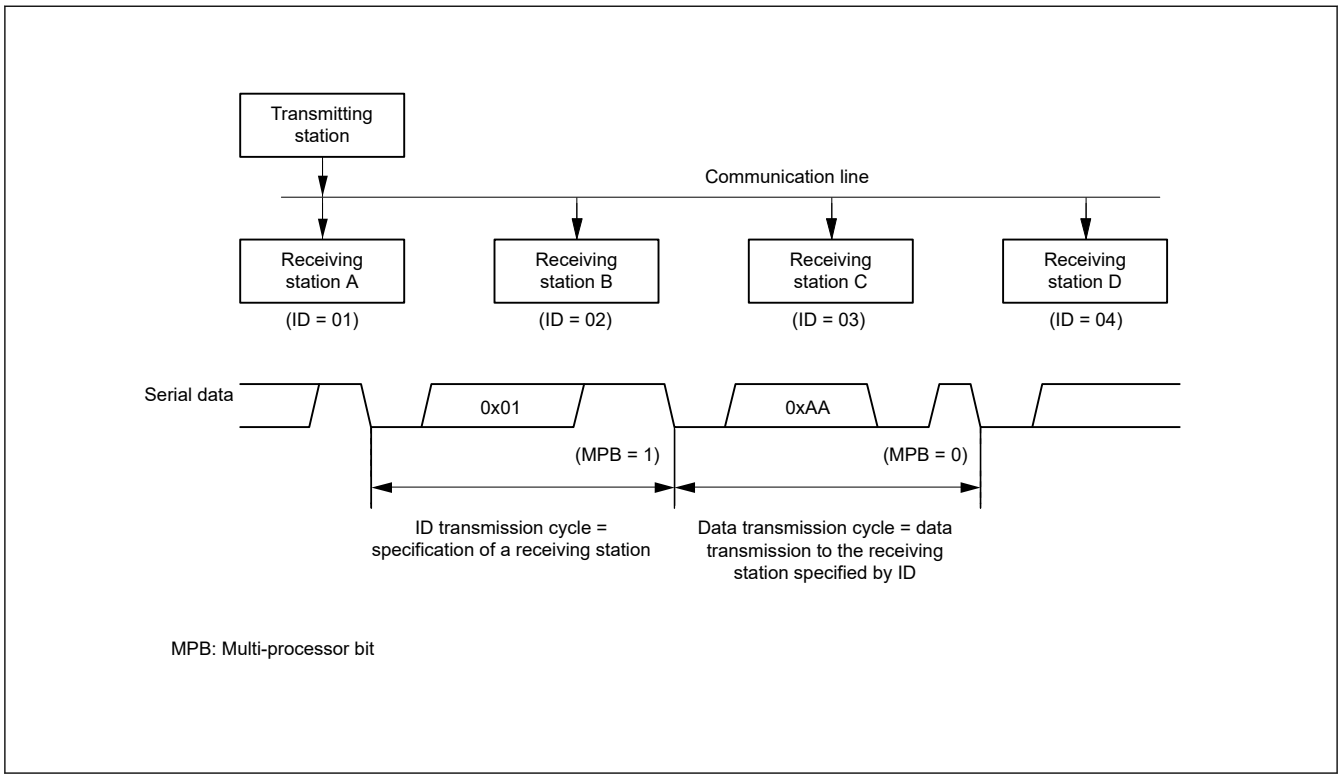


Figure 24.26 Example of communication using multi-processor format with transmission of data 0xAA to receiving station A

(2) FIFO selected

For data transmission, software must write data to FTDRHL.MPBT that corresponds to transmit data in FTDRHL.TDAT. For data reception, the multi-processor bit that is part of the receive data is written to FTDRHL.MPB and receive data is written to FRDRL.

When the MPIE bit is set to 1, the following operations are disabled until reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from the RSR register to the FRDRHL register
- Detection of a receive error
- Break
- Setting of the respective RDF, ORER, and FER status flags in the SSR_FIFO register

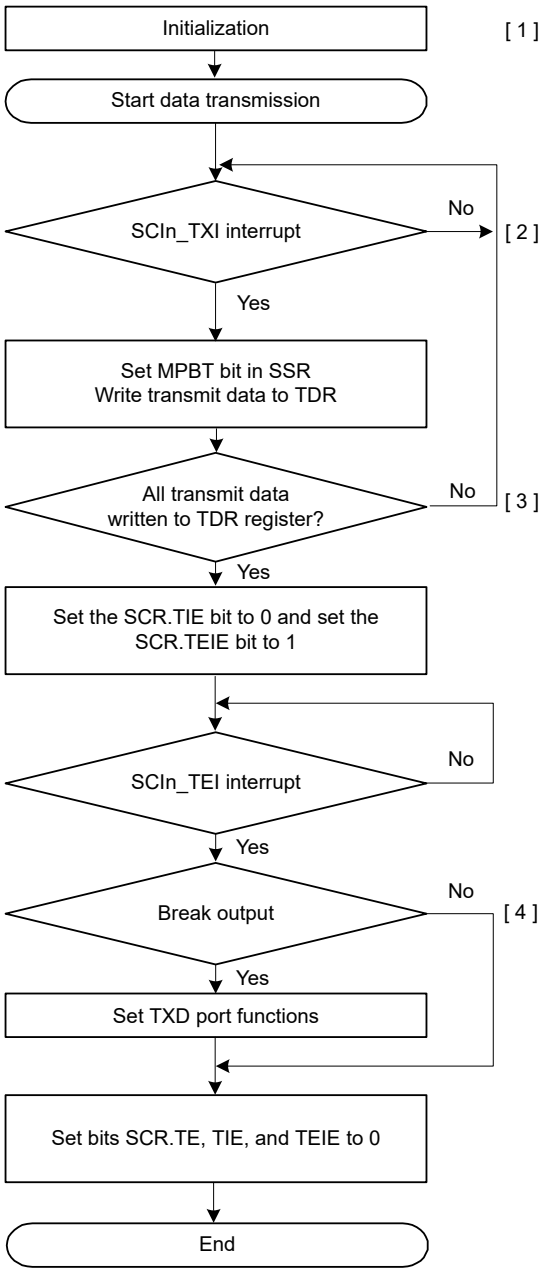
When the SCI receives an 8-bit character in which the multi-processor bit is set to 1, the FTDRHL.MPB bit is set to 1 and receive data is written to FRDRHL.RDAT. The SCR.MPIE bit is automatically cleared, returning the SCI to normal reception operation. If the SCR.RIE bit is set to 1, an SCIn_RXI interrupt is generated.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in normal asynchronous mode with FIFO selected.

24.4.1 Multi-Processor Serial Data Transmission

(1) Non-FIFO selected

Figure 24.27 shows an example flow of multi-processor data transmission. In the ID transmission cycle, the ID must be transmitted with the SSR.MPBT bit set to 1. In the data transmission cycle, the data must be transmitted with the MPBT bit set to 0. The rest of the operations are the same as operations in asynchronous mode. Write the values in the order of the FTDRH register then the FTDRL register.



[1] SCI initialization:
Set data transmission.
After the SCR.TE bit is set to 1, 1 is output for a frame, and transmission is enabled.

[2] SCIn_TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (SCIn_TXI) request is generated.
Set the MPBT bit in SSR to 0 or 1, and write transmit data to TDR once in the SCIn_TXI interrupt processing routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR once using an SCIn_TXI interrupt request.
Transmit data can also be written to TDR by activating the DTC.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, after setting the output state (LOW level output) of TXD pin by SPTR.SPB2IO and SPTR.SPB2DT bits, set the SCR.TE bit to 0.

Figure 24.27 Example flow of multi-processor serial transmission with non-FIFO selected

(2) FIFO selected

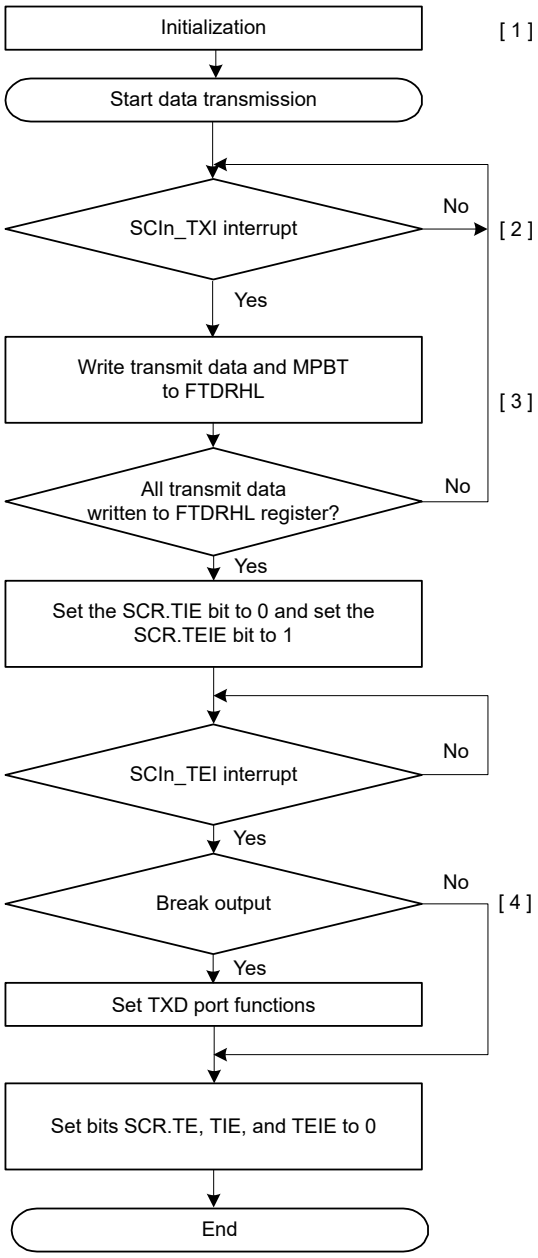
Figure 24.28 shows an example of data format that is written to FTDRH and FTDRL in multi-processor mode. The FTDRH.MPBT bit is set to 1. Data is set to FTDRH and FTDRL with the correct data length. Write 0 for unused bits. Write in order from FTDRH to FTDRL.

Data Length	Register Setting		Transmit data in FTDRH, FTDRL															
			FTDRHL															
			FTDRH								FTDRL							
	SCMR.CHR1	SMR.CHR	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	0	–	–	–	–	–	–	MPBT	–	–	7-bit transmit data						
8 bits	1	1	–	–	–	–	–	–	MPBT	–	8-bit transmit data							
9 bits	0	Don't care	–	–	–	–	–	–	MPBT	9-bit transmit data								

Note: —: Invalid. The write value should be 0.

Figure 24.28 Data format written to FTDRH and FTDRL in multi-processor mode with FIFO selected

Figure 24.29 shows an example flow of multi-processor serial transmission with FIFO selected. In the ID transmission cycle, the ID must be transmitted with the FTDRH.MPBT bit set to 1. In the data transmission cycle, the data must be transmitted with the MPBT bit set to 0. The rest of the operations are the same as operations in asynchronous mode with FIFO selected.



[1] SCI initialization:
Set data transmission.
After the SCR.TE bit is set to 1, 1 is output for a frame, and transmission is enabled.

[2] Transmit data write to FTDRHL by an SCIn_TXI interrupt request:
When transmit data is transferred from FTDRHL to TSR, when the quantity of receive data written in FTDRHL is equal to or less than the specified transmit triggering number, a transmit data FIFO empty interrupt (SCIn_TXI) request is generated.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data and MPBT to FTDRHL once using an SCIn_TXI interrupt request.
Transmit data can also be written to FTDRHL by activating the DTC.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the FTDRHL.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, after setting the output state (LOW level output) of TXD pin by SPTR.SPB2IO and SPTR.SPB2DT bits, set the SCR.TE bit to 0.

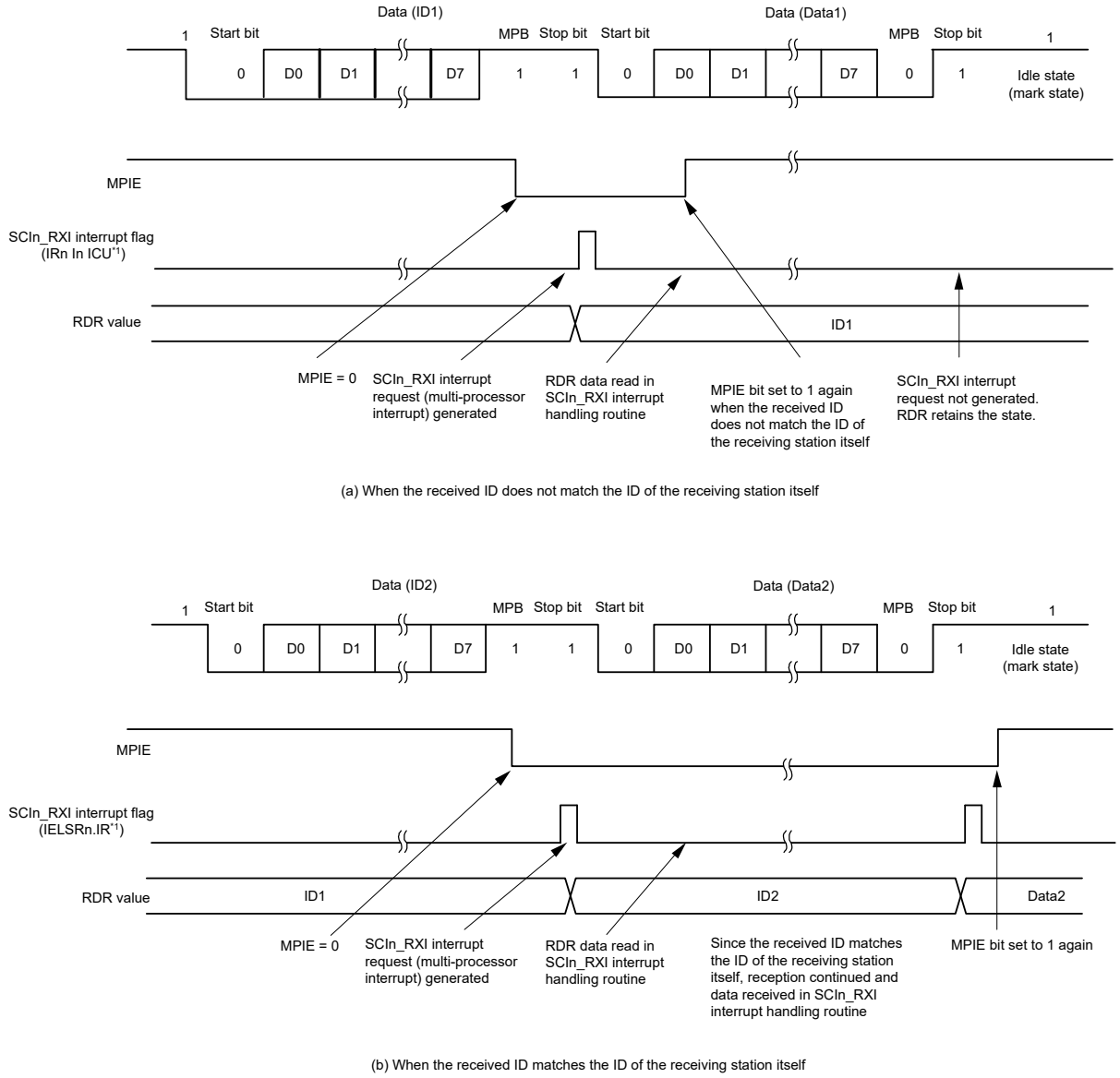
Figure 24.29 Example flow of serial transmission in multi-processor mode with FIFO selected

24.4.2 Multi-Processor Serial Data Reception

(1) Non-FIFO selected

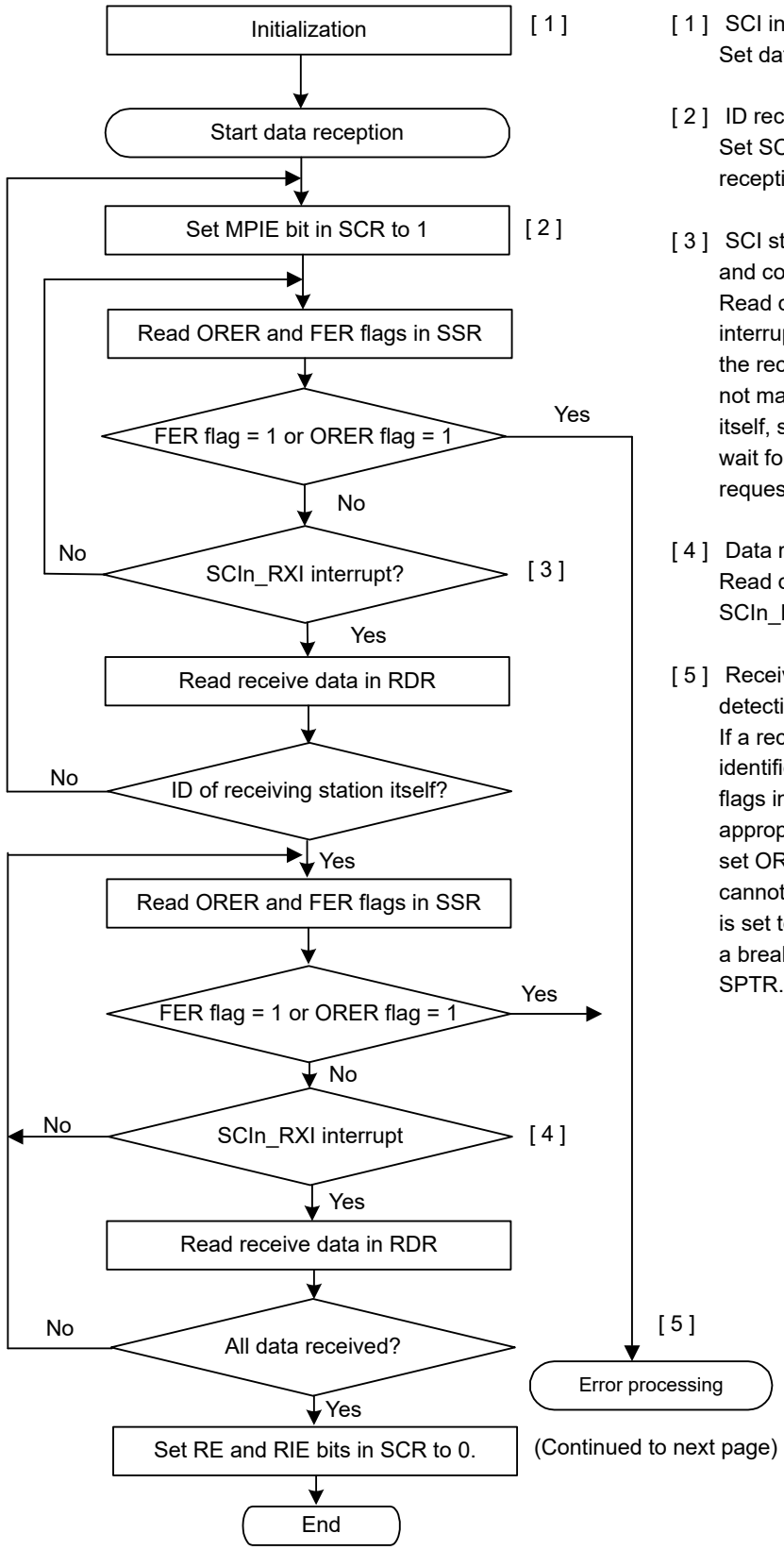
Figure 24.31 and Figure 24.32 are example flows of multi-processor serial reception. When the SCR.MPIE bit is set to 1, reading communication data is skipped until reception of communication data in which the multi-processor bit is set to 1. When communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to the RDR register (the RDRHL register when 9-bit data length is selected), and the SCIn_RXI interrupt request is generated. The rest of the operations are the same as operations in asynchronous mode. Read the order from FRDRH to FRDRL.

Figure 24.30 shows an example operation for data reception.



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 24.30 Example of SCI reception with 8-bit data, multi-processor bit, and 1 stop bit



[1] SCI initialization:
Set data reception.

[2] ID reception cycle:
Set SCR.MPIE bit to 1 and wait for ID reception.

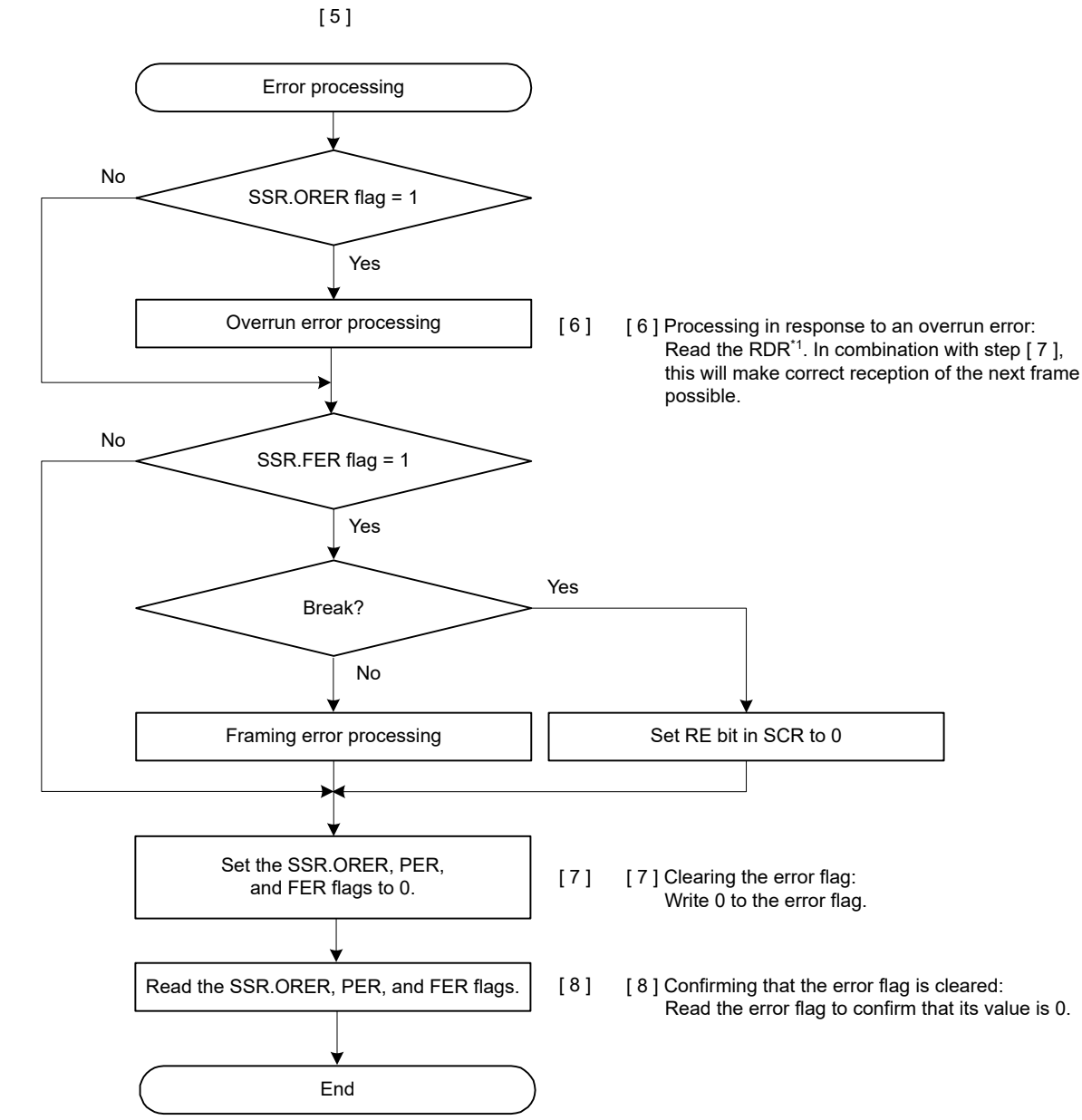
[3] SCI status confirmation and reception and comparison of ID:
Read data in RDR*1 at the first SCIn_RXI interrupt and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again, and wait for another SCIn_RXI interrupt request.

[4] Data reception at an SCIn_RXI interrupt:
Read data in RDR*1 once in the SCIn_RXI interrupt routine.

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and FER flags in SSR. After performing the appropriate error processing, be sure to set ORER and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading SPTR.RXDMON flag.

Note 1. The RDR register becomes the RDRHL register when 9-bit data length is selected.

Figure 24.31 Example flow of multi-processor serial reception with non-FIFO selected (1)



Note 1. The RDR register becomes the RDRHL register when 9-bit data length is selected.

Figure 24.32 Example flow of multi-processor serial reception with non-FIFO selected (2)

(2) FIFO selected

Figure 24.33 shows an example of a data format that is written to FRDRH and FRDRL in multi-processor mode.

In multi-processor mode, the MPB value that is a part of the receive data is written to the FRDRH.MPB bit. A value of 0 is written to the FRDRH.PER flag. Data is written to FRDRH and FRDRL with the correct data length. Unused bits are written with 0. Read in order from FRDRH to FRDRL. When software reads the FRDRL register, the SCI updates FER, MPB, and receive data (RDAT[8:0]) in FRDRL with the next data. The RDF, ORER and DR flags in the FRDRH register always reflect the associated flags in the SSR_FIFO register.

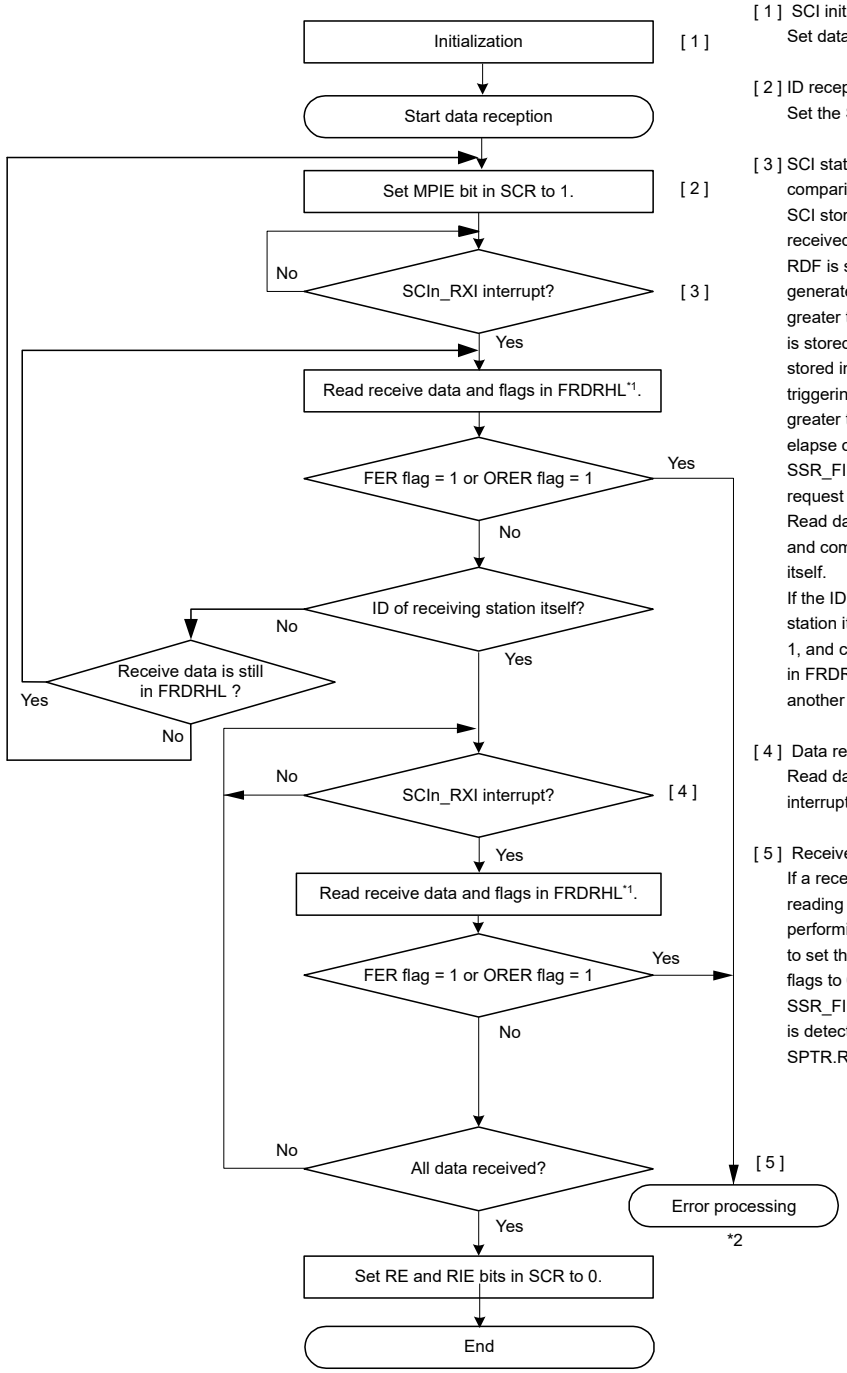
Data Length	Register Setting		Receive data in FRDRH, FRDRL															
			FRDRHL															
			FRDRH								FRDRL							
	SCMR.CHR1	SMR.CHR	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	0	–	RDF	ORER	FER	0	DR	MPB	0	0	7-bit receive data						
8 bits	1	1	–	RDF	ORER	FER	0	DR	MPB	0	8-bit receive data							
9 bits	0	Don't care	–	RDF	ORER	FER	0	DR	MPB	9-bit receive data								

Note: When data length is 7 bits, 0 is always read for FRDRH[0] and FRDRL[7]
When data length is 8 bits, 0 is always read for FRDRH[0]
FRDRHL[15] bit is read as an indefinite value

Figure 24.33 Data format stored in FRDRH and FRDRL in multi-processor mode with FIFO selected

Figure 24.34 shows an example flow of multi-processor data reception with FIFO selected. When the SCR.MPIE is set to 1, reading communication data is skipped until reception of communication data in which the multi-processor bit is set to 1. When communication data in which the multi-processor bit is set to 1 is received, the received data, MPB and associated errors are transferred to the FRDRHL register. The SCR.MPIE bit is automatically cleared and normal reception continues.

If a framing error occurs and the SSR_FIFO.FER flag is set to 1, the SCI continues data reception. The rest of the operations are the same as operations in asynchronous mode with FIFO selected.



[1] SCI initialization:
Set data reception.

[2] ID reception cycle:
Set the SCR.MPIE bit to 1 and wait for ID reception.

[3] SCI status confirmation and reception and comparison of ID:
SCI stores first data (MPB = 1) and subsequent received data in FRDRHL.
RDF is set to 1 and an SCIn_RXI interrupt request is generated when a quantity of receive data equal to or greater than the specified receive triggering number is stored in FRDRHL. When the quantity of data stored in FRDRHL falls the specified receive triggering number and received data is equal to or greater than 1, and no next data is received after the elapse of 15 etus from the last stop bit, SSR_FIFO.DR is set to 1. An SCIn_RXI interrupt request is generated when FCR.DRES bit is 0.
Read data in FRDRHL at the first SCIn_RXI interrupt, and compare it with the ID of the receiving station itself.
If the ID does not match the ID of the receiving station itself, the SCI reads until the data with MPB = 1, and compares next ID. If it is no data with MPB = 1 in FRDRHL, set the MPIE bit to 1 again and wait for another SCIn_RXI interrupt request.

[4] Data reception at an SCIn_RXI interrupt:
Read data in FRDRHL once in the SCIn_RXI interrupt routine.

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading the ORER and FER flags in SSR_FIFO. After performing the appropriate error processing, be sure to set the SSR_FIFO.ORER and SSR_FIFO.FER flags to 0. Reception cannot be resumed if the SSR_FIFO.ORER flag is set to 1. When framing error is detected, a break can be detected by reading the SPTR.RXDMON flag.

Note 1. If FRDRH and FRDRL are used instead of FRDRHL, read in order from FRDRH to FRDRL.
Note 2. Same as Figure 24.32

Figure 24.34 Example flow of serial reception in multi-processor mode with FIFO selected

24.5 Operation in Clock Synchronous Mode

Figure 24.35 shows the data format for clock synchronous serial data communications.

In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. For single-character data transfer, data consists of 8-bit. In clock synchronous mode, no parity bit can be added.

In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next falling edge. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit

data is output, the transmission line holds the last bit as output state. When the SPMR.CKPH bit is 1 in slave mode, the transmission line holds the first bit output state.

Within the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by using a shared clock. Both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.

However, it is not possible to perform continuous transfer in the fastest bit rate setting (BRR[7:0] = 0x00 and SMR.CKS[1:0] = 00b). Therefore, when the FIFO is selected, this setting (BRR[7:0] = 0x00 and SMR.CKS[1:0] = 00b) is not available.

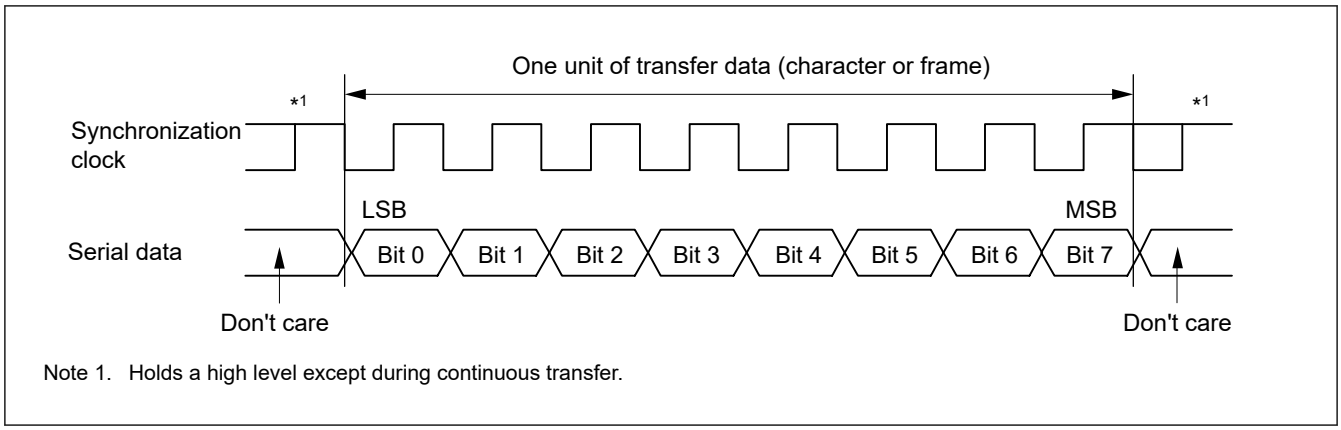


Figure 24.35 Data format in clock synchronous serial communications with LSB-first order

24.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCKn pin can be selected based on the SCR.CKE[1:0] setting.

When the SCI operates on an internal clock, the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output in the transfer of one character. When no transfer is performed, the clock is held high. However, when only data reception is performed while the CTS function is disabled, the synchronization clock output starts when the SCR.RE bit set to 1. The synchronization clock stops when it goes high[*1] and an overrun error occurs or the SCR.RE bit is set to 0.

When only data reception is performed and the CTS function is enabled, the clock output does not start when the SCR.RE bit set to 1 and the CTSn_RTSn pin input is high. The synchronization clock output starts when the SCR.RE bit is set to 1 and the CTSn_RTSn pin input is low. Following that, when the CTSn_RTSn pin input is high on completion of the frame reception, the synchronization clock output stops when it goes high. If the CTSn_RTSn pin input continues to be low, the synchronization clock stops when it goes high[*1] and an overrun error occurs or the SCR.RE bit is set to 0.

Note 1. The signal is held high while (SPMR.CKPH = 0 and SPMR.CKPOL = 1) or (SPMR.CKPH = 1 and SPMR.CKPOL = 1). It is held low while (SPMR.CKPH = 0 and SPMR.CKPOL = 1) or (SPMR.CKPH = 1 and SPMR.CKPOL = 0).

24.5.2 CTS and RTS Functions

In the CTS function, the CTSn_RTSn pin input controls the start of data reception or transmission when the clock source is the internal clock. Setting the SPMR.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, setting the CTSn_RTSn pin low causes data reception or transmission to start.

Setting the CTSn_RTSn pin high while the data transmission or reception is in progress does not affect transmission or reception of the current frame.

In the RTS function, the CTSn_RTSn pin output is used to request the start of data reception or transmission when the clock source is an external synchronizing clock. The CTSn_RTSn output goes low when serial communication is enabled. Conditions for output of the CTSn_RTSn low and high are shown as follows:

[Conditions for low output]

Satisfaction of all the following conditions:

Non-FIFO selected when all of the following conditions are satisfied

- The value of the SCR.RE bit or the SCR.TE bit is 1
- Neither transmission nor reception is in progress
- There is no received data available to be read when the SCR.RE bit is 1
- Transmit data is written when the SCR.TE bit is 1 and SCR.CKE[1] bit is 0
- Data is available for transmission in the TSR register when SCR.TE bit is 1 and SCR.CKE[1] bit is 1
- The SSR.ORER flag is 0

FIFO selected when all of the following conditions are satisfied

- The value of the SCR.RE bit or the SCR.TE bit is 1
- Neither transmission nor reception is in progress
- The amount of receive data written in FRDRHL is less than the setting value of FCRH.RSTRG[3:0] when SCR.RE = 1
- Data that has not been transmitted is available in FTDRHL when SCR.TE bit is 1 and SCR.CKE[1] bit is 0
- Data is available for transmission in the TSR register when SCR.TE bit is 1 and SCR.CKE[1] bit is 1
- The SSR_FIFO.ORER flag is 0

[Condition for high output]

- The conditions for low output are not satisfied

24.5.3 SCI Initialization in Clock Synchronous Mode

Before transmitting and receiving data, start by writing the initial value 0x00 to the SCR register, then continue through the SCI initialization procedure given in the sections describing non-FIFO and FIFO selection in section 24.5.2. CTS and RTS Functions. Anytime the operating mode or transfer format is to be changed, the SCR register must be initialized before the change can be made.

Note: Setting the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in SSR/SSR_FIFO nor the RDR register. When the TE bit is set to 0, the TEND flag for the selected FIFO buffer is not initialized.

Note: In non-FIFO mode, switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 when the SCR.TIE bit is 1 generates an SCIn_TXI interrupt request.

Table 24.28 Example flow of SCI initialization in clock synchronous mode with non-FIFO selected (1 of 2)

No.	Step Name	Description
1	Start initialization	
2	Set the SCR.TIE, RIE, TE, RE, and TEIE bits to 0	
3	Set the FCR.FM bit to 0	Set the FCR.FM bit to 0.
4	Set the SCR.CKE[1:0] bits	Set the clock selection in SCR.
5	Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits.	Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits. Step 5 can be skipped if the values have not been changed from the initial values.
6	Set the data transmission/ reception format in SMR, SCMR, and SEMR	Set data transmission/reception format in SMR, SCMR, and SEMR.
7	Set a value in BRR	Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.
8	Set a value in MDDR	Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is set to 0 or an external clock is used.
9	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

Table 24.28 Example flow of SCI initialization in clock synchronous mode with non-FIFO selected (2 of 2)

No.	Step Name	Description
10	Set the SCR.TE or RE bit to 1, and set the SCR.TIE and RIE bits	Set the SCR.TE or RE bit to 1. Also set the SCR.TIE and RIE bits. Setting the TE and RE bits allows TXDn and RXDn pins to be used.
11	Initialization completion	

Note: In simultaneous transmit and receive operations, the TE and RE bits in SCR must both be set to 0 or set to 1 simultaneously

Table 24.29 Example flow of SCI initialization in clock synchronous mode with FIFO selected

No.	Step Name	Description
1	Start initialization	
2	Set the SCR.TIE, RIE, TE, RE, and TEIE bits to 0	
3	Set the FCR.FM, TFRST, and RFRST bits to 1. Set the FCR.TTRG[3:0], RTRG[3:0], and RSTRG[3:0] bits.	Set the FCR.FM, TFRST, and RFRST bits to 1 (FIFO mode enabled, transmit/receive FIFOs empty). Set the FCR.TTRG[3:0], RTRG[3:0], and RSTRG[3:0] bits.
4	Set the SCR.CKE[1:0] bits	Set the clock selection in SCR.
5	Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits.	Set the SIMR1.IICM bit to 0. Set the SPMR.CKPH and CKPOL bits. Step 5 can be skipped if the values have not been changed from the initial values.
6	Set the data transmission/reception format in SMR, SCMR, and SEMR	Set data transmission/reception format in SMR, SCMR, and SEMR.
7	Set a value in BRR	Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.
8	Set a value in MDDR	Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is set to 0 or an external clock is used.
9	Set the FCR.TFRST and RFRST bits to 0	Set the FCR.TFRST and RFRST bits to 0.
10	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
11	Set the SCR.TE or RE bit to 1, and set the SCR.TIE and RIE bits	Set the SCR.TE or RE bit to 1. Also set the SCR.TIE and RIE bits. Setting the TE and RE bits allows TXDn and RXDn pins to be used.
12	Initialization completion	

Note: In simultaneous transmit and receive operations, the TE and RE bits in SCR must both be set to 0 or set to 1 simultaneously.

24.5.4 Serial Data Transmission in Clock Synchronous Mode

(1) Non-FIFO selected

Figure 24.36, Figure 24.37, and Figure 24.38 show examples of serial transmission in clock synchronous mode.

In serial data transmission, the SCI operates as follows:

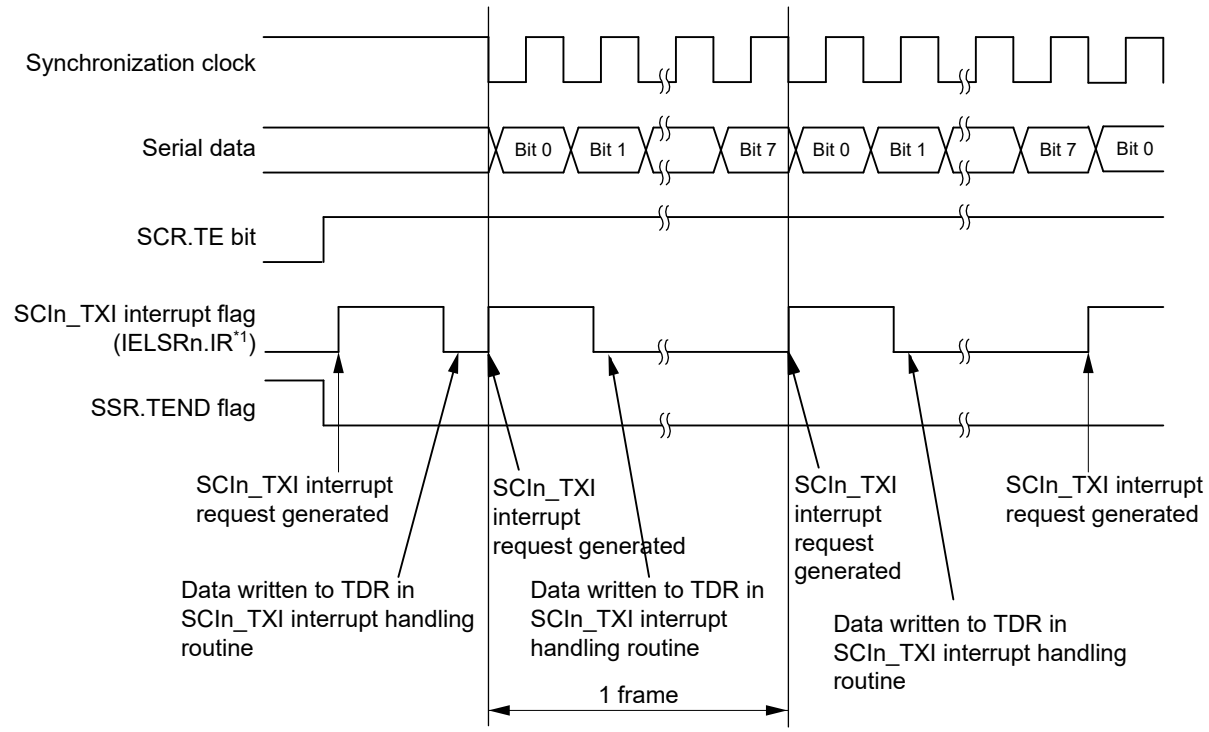
1. The SCI transfers data from the TDR register to the TSR register when data is written to TDR in the SCIn_TXI interrupt handling routine. The SCIn_TXI interrupt request at the beginning of transmission is generated when the TE bit is set to 1 but only after the TIE bit in the SCR is also set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. After transferring data from TDR to TSR, the SCI starts transmission. When the SCR.TIE bit is set to 1, an SCIn_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to TDR in the SCIn_TXI interrupt handling routine before transmission of the current transmit data finishes. When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the TDR register from the handling routine for SCIn_TXI requests.

3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when the clock output mode is specified and in synchronization with the input clock when the use of an external clock is specified. Output of the clock signal is suspended until the input CTS signal is low when the SPMR.CTSE bit is 1.
4. The SCI checks for update to the TDR register on output of the last bit.
5. When the TDR register is updated, the next transmit data is transferred from TDR to TSR, and serial transmission of the next frame starts.
6. If TDR is not updated, the SSR.TEND flag is set to 1. The TXDn pin retains the output state of the last bit. If the SCR.TEIE bit is 1, an SCIn_TEI interrupt request is generated and the SCKn pin is held high.

Figure 24.36, Figure 24.37, and Figure 24.38 show examples of serial data transmission.

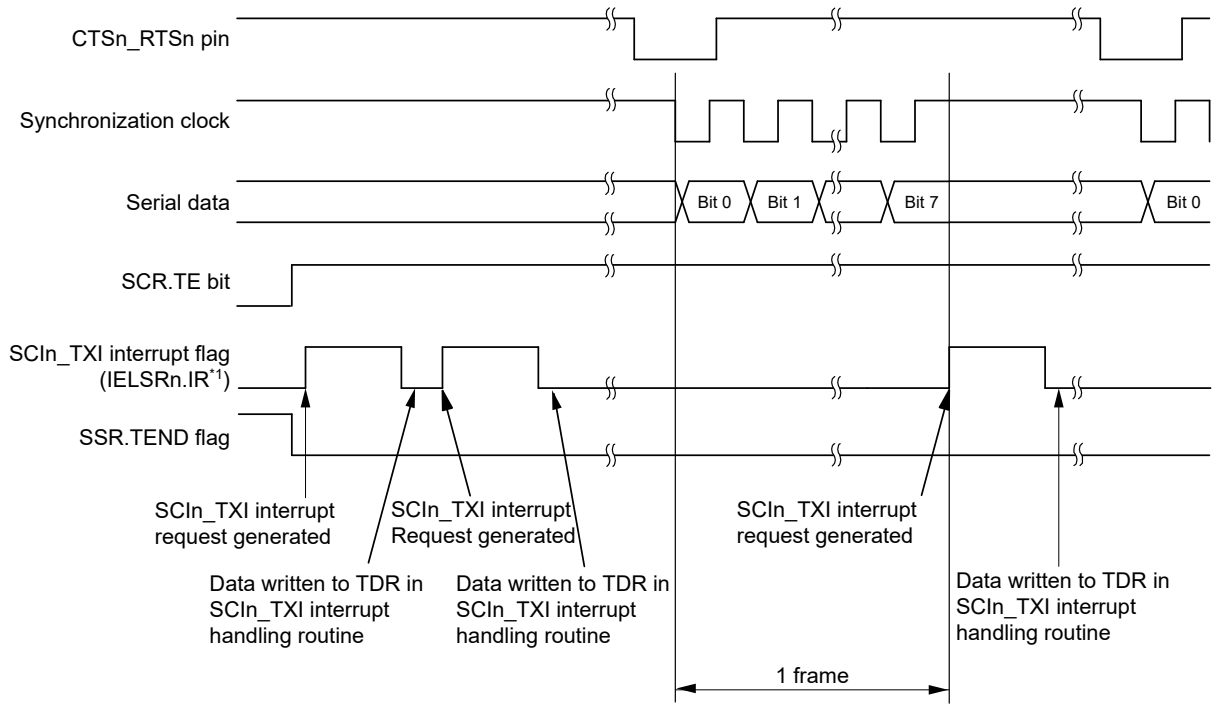
Transmission does not start while a receive error flag (ORER, FER, or PER in SSR) is set to 1. Always set the receive error flags to 0 before starting transmission.

Note: Setting the SCR.RE bit to 0 does not clear the receive error flags.



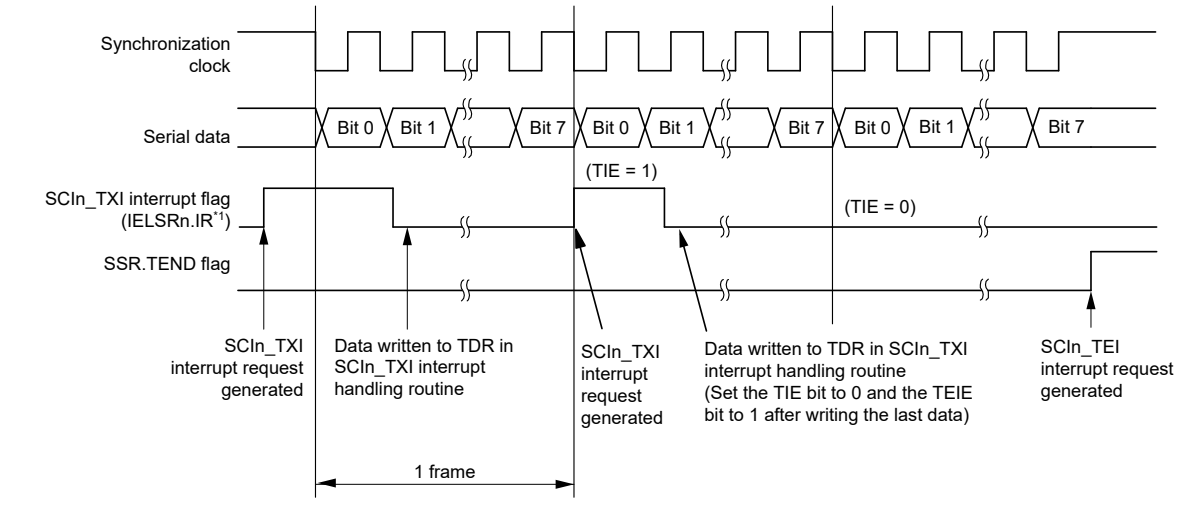
Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 24.36 Example of serial data transmission in clock synchronous mode when the CTS function is not used at the beginning of transmission



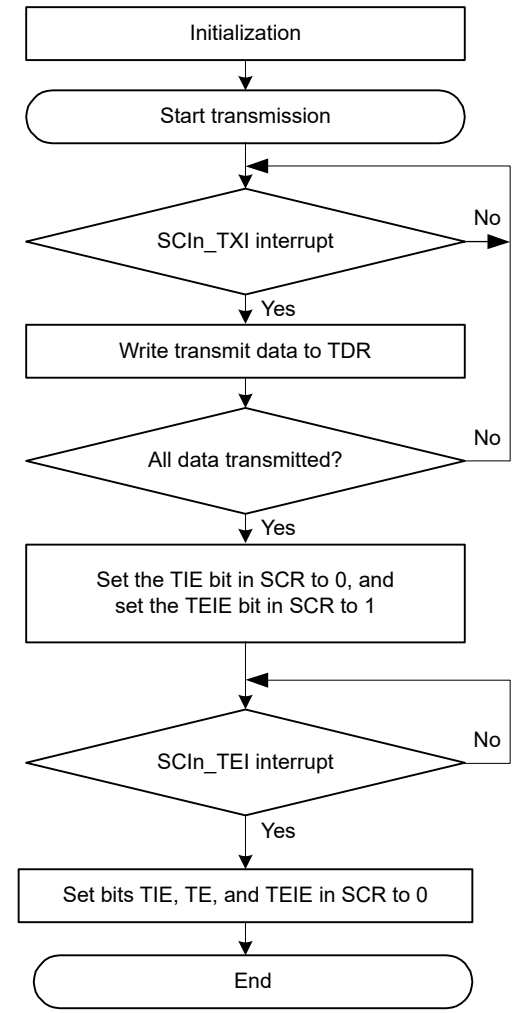
Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 24.37 Example of serial data transmission in clock synchronous mode when the CTS function is used at the beginning of transmission



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 24.38 Example of serial data transmission in clock synchronous mode from the middle of transmission until transmission completion



[1] [1]SCI initialization:
Set data transmission.

[2] [2]Writing transmit data write to TDR by an SCIn_TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (SCIn_TXI) request is generated.
Transmit data is written to TDR once from the handling routine for SCIn_TXI requests.

[3] [3]Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR upon accepting a transmit data empty interrupt (SCIn_TXI) request. Transmit data can also be written to TDR by activating the DTC by theSCIn_TXI interrupt request.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to theTDR

Note: When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCKn pin for the last bit sets the SSR.TEND flag to 1. Setting the SCR.TE bit to 0 immediately after this might lead to insufficient received-data hold time on the receiver side.

Figure 24.39 Example flow of serial transmission in clock synchronous mode with non-FIFO selected

(2) FIFO selected

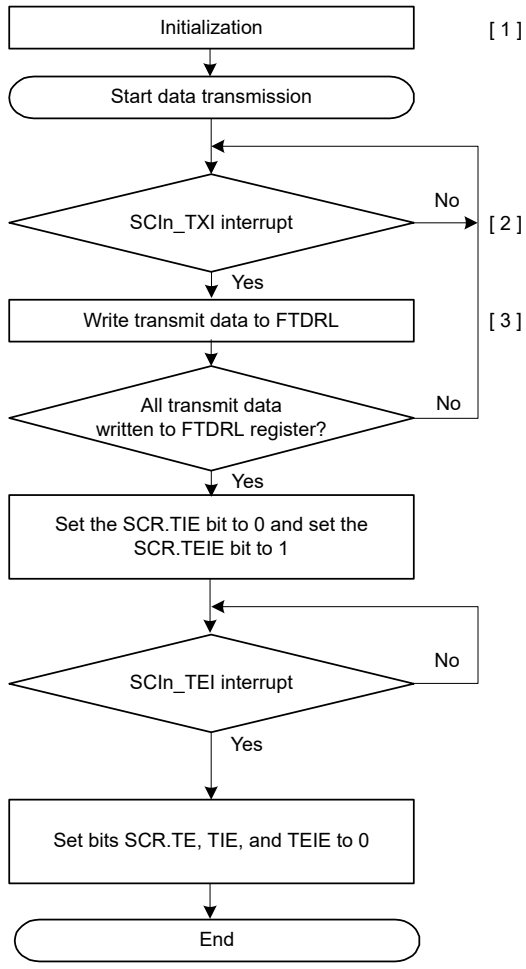
Figure 24.40 shows an example of serial transmission in clock synchronous mode with FIFO selected.

In serial data transmission, the SCI operates as follows:

1. The SCI transfers data from the FTDRL[*1] register to the TSR register when data is written to FTDRL[*1] in the SCIn_TXI interrupt handling routine. The amount of data that can be written to FTDRL is 16 minus FDR.T[4:0] bytes. The SCIn_TXI interrupt request at the beginning of transmission is generated when the SCR.TE bit is set to 1 but only after the SCR.TIE bit is also set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. After transferring data from FTDRL to TSR, the SCI starts transmission. When the amount of transmit data written in FTDRL is equal to or less than the specified transmit triggering number, the SSR_FIFO.TDFE is set to 1. When the SCR.TIE bit is set to 1, an SCIn_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to FTDRL in the SCIn_TXI interrupt handling routine before transmission of the current transmit data has finished. When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the FTDRL from the handling routine for SCIn_TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when the clock output mode is specified and in synchronization with the input clock when the use of an external clock is specified. Output of the clock signal is suspended until the input CTS signal is low when the SPMR.CTSE bit is 1.
4. The SCI checks whether non-transmitted data remains in FTDRL on output of the stop bit.

5. When FTDRL is updated, the next transmit data is transferred from FTDRL to TSR and serial transmission of the next frame starts.
6. If FTDRL is not updated, the SSR_FIFO.TEND flag is set to 1. The TXDn pin retains the output state of the last bit. If the SCR.TEIE bit is 1, an SCIn_TEI interrupt request is generated and the SCKn pin is held high.

Note 1. In clock synchronous mode, FTDRH is not used.



[1]SCI initialization:
Set data transmission.
After the TE bit in SCR is set to 1, 1 is output for a frame, and transmission is enabled.

[2]Transmit data write to FTDRL by an SCIn_TXI interrupt request:
For transmission data from FTDRL to TSR, when the amount of transmit data written in FTDRL is equal to or less than the specified transmit triggering number, a transmit data FIFO empty interrupt (SCIn_TXI) request is generated. Write transmit data to FTDRL once in the SCIn_TXI interrupt handling routine.

[3]Serial transmission continuation procedure:
To continue serial transmission, write the next transmit data to FTDRL in this SCIn_TXI interrupt handling routine and clear the SSR_FIFO.TDFE flag to 0 before transmission of the current transmit data finishes.
Transmit data can also be written to FTDRL by activating the DTC. The TDFE flag is cleared automatically in this case. Do not write to the TDFE flag.
When SCIn_TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted is written to the FTDRL.

Note: When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCKn pin for the last bit sets the SSR_FIFO.TEND flag to 1. Setting the SCR.TE bit to 0 immediately after this might lead to insufficient received-data hold time on the receiver side.

Figure 24.40 Example flow of serial transmission in clock synchronous mode with FIFO selected

24.5.5 Serial Data Reception in Clock Synchronous Mode

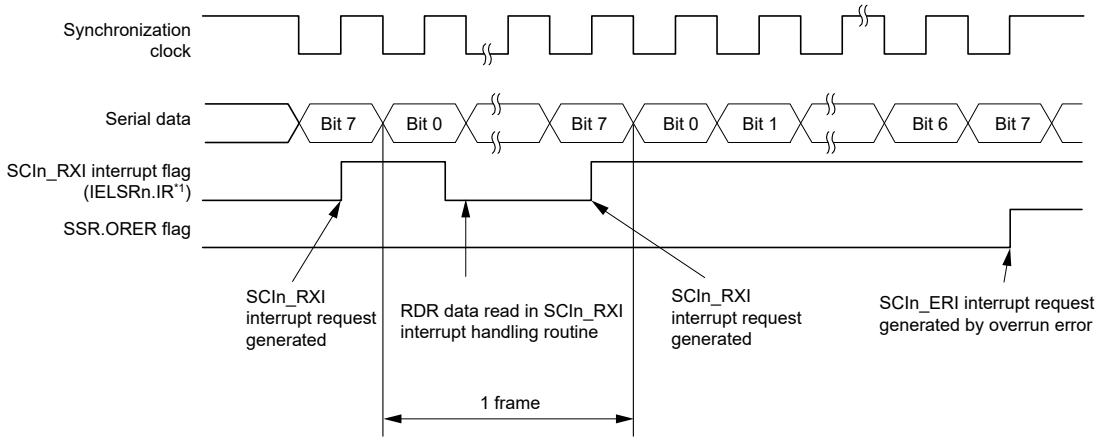
(1) Non-FIFO selected

Figure 24.41 and Figure 24.42 show examples of SCI operation for serial reception in clock synchronous mode.

In serial data reception, the SCI operates as follows:

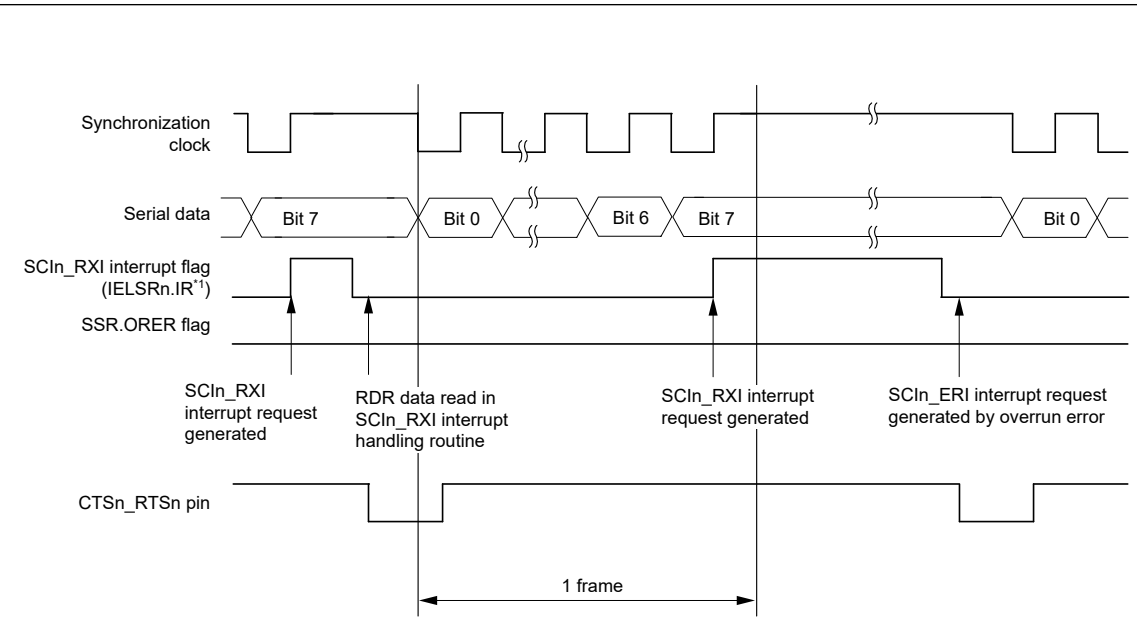
1. When the value of the SCR.RE bit becomes 1, the CTSn_RTSn pin goes low.
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in the RSR register.
3. If an overrun error occurs, the SSR.ORER flag is set to 1. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to the RDR register.

4. When reception completes successfully, receive data is transferred to the RDR register. If the SCR.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the received data transferred to the RDR register in the SCIn_RXI interrupt handling routine before reception of the next receive data completes. Reading the received data that is transferred to RDR causes the CTSn_RTSn pin to output low.



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 24.41 Example operation for serial reception in clock synchronous mode (1) when the RTS function is not used



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 24.42 Example operation for serial reception in clock synchronous mode (2) when RTS function is used

Data transfer cannot resume while the receive error flag is 1. Therefore, clear the ORER, FER, and PER flags in the SSR register to 0 before resuming data reception. Additionally, always read the RDR register during overrun error processing. When a data reception is forced to terminate by a 0 write to the SCR.RE bit during operation, read the RDR register because received data that is not yet read might be left in the RDR register.

Figure 24.43 shows an example flow of serial data reception.

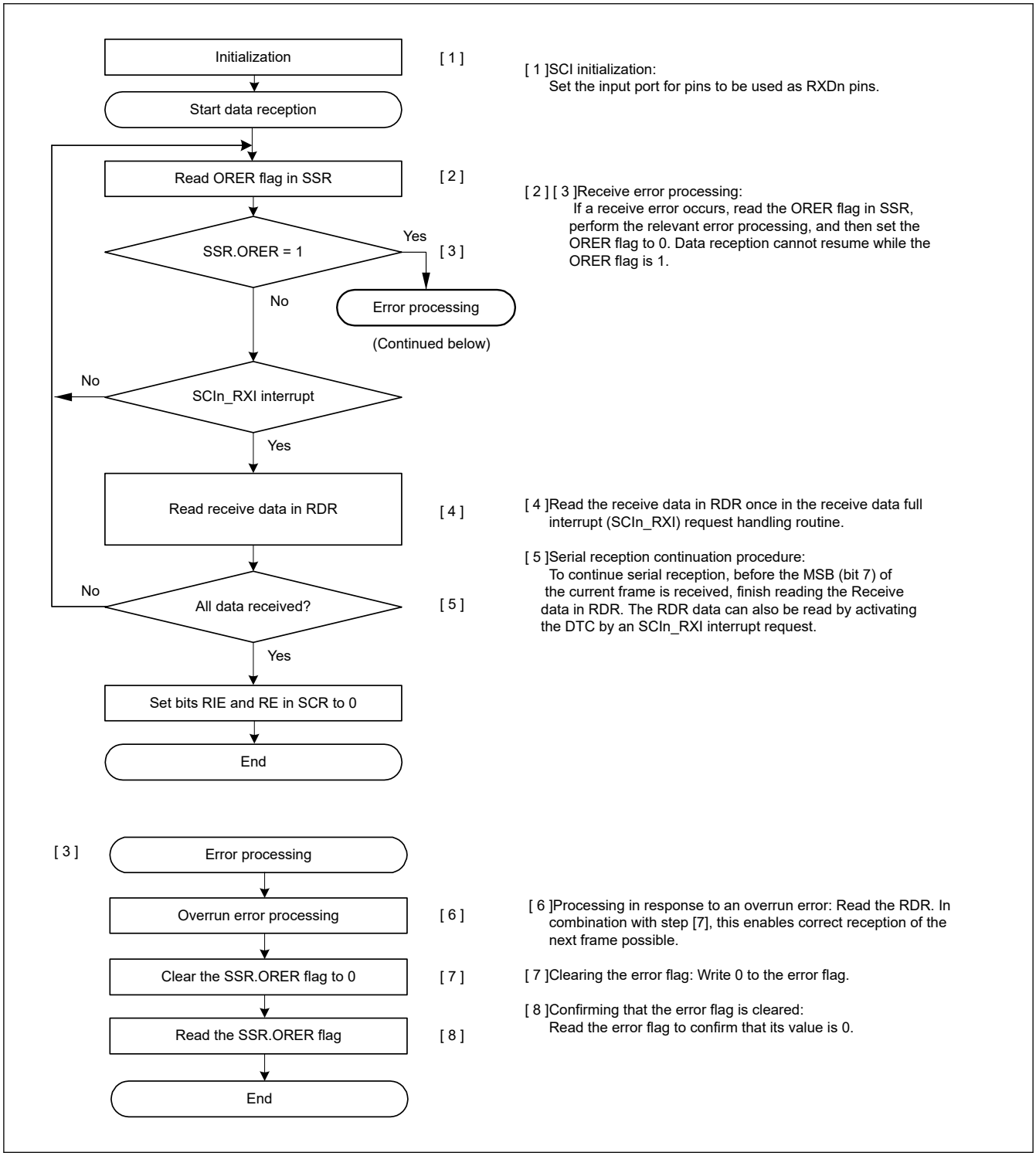


Figure 24.43 Example flow of serial reception in clock synchronous mode with non-FIFO selected

(2) FIFO selected

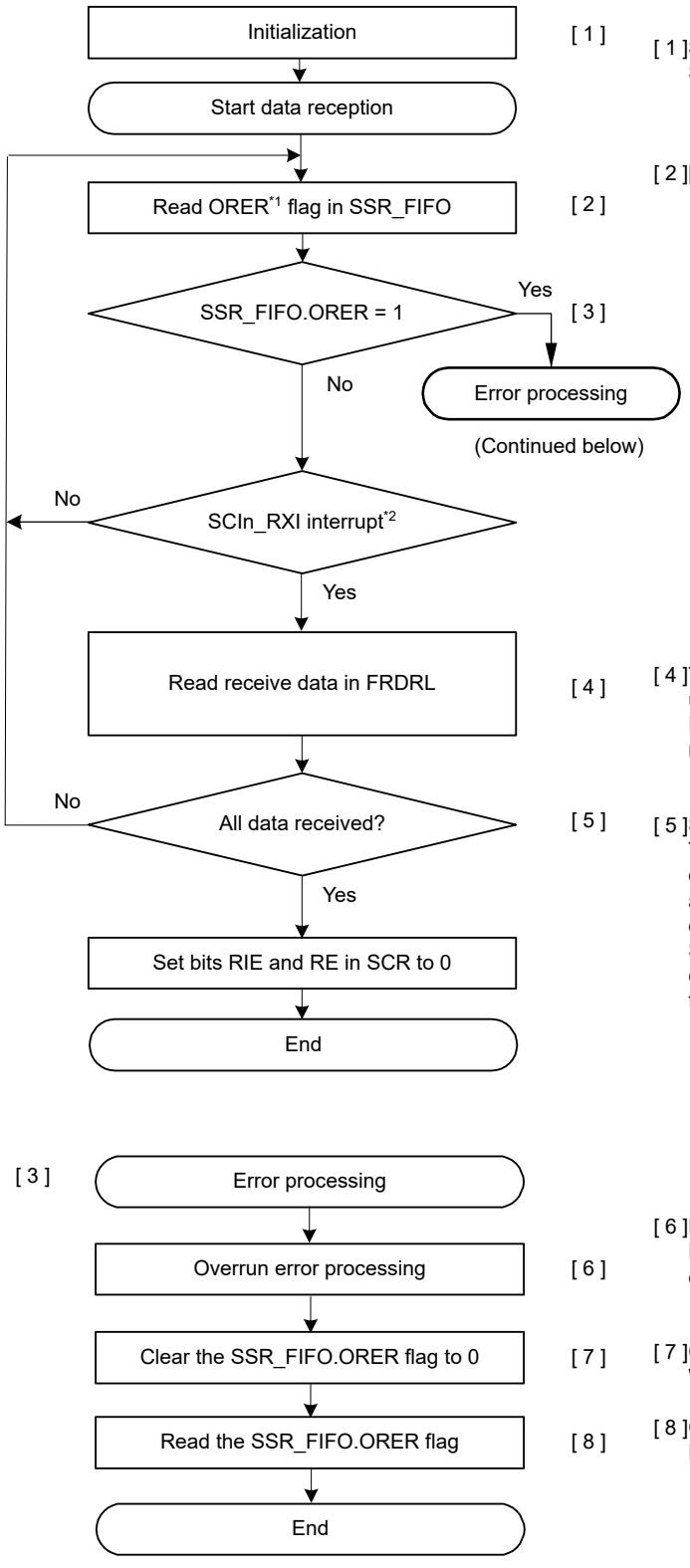
Figure 24.44 shows an example of serial reception in clock synchronous mode with FIFO selected.

In serial data reception, the SCI operates as follows:

1. When the value of the SCR.RE bit becomes 1, the CTSn_RTSn pin goes low.
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in the RSR register.

3. If an overrun error occurs, the SSR_FIFO.ORER flag is set to 1. If the SCR.RIE bit is 1, an SCIn_ERI interrupt request is generated. Received data is not transferred to the FRDRL[*1] register.
4. When data reception completes successfully, the receive data is transferred to the FRDRL[*1] register. The RDF flag is set to 1 when the amount of the receive data stored in FRDRL is equal to or greater than the specified receive triggering number. If the SCR.RIE bit is 1, an SCIn_RXI interrupt request is generated. Continuous data reception is enabled by reading the receive data transferred to FRDRL[*2] in the SCIn_RXI interrupt handling routine before an overrun error occurs. If the amount of received data that is transferred to FRDRL is less than the specified receive triggering number, the CTSn_RTSn pin goes low.

Note 1. In clock synchronous mode, FRDRH is not used.
Note 2. Read data in order from FRDRH to FRDRL when RDF and ORER are read with receive data.



[1]SCI initialization:
Set the input port for pins to be used as RXDn pins.

[2][3]Receive error processing:
If a receive error occurs, read the ORER flag in SSR_FIFO, perform the relevant error processing, then set the SSR_FIFO.ORER flag to 0. Data reception cannot resume while the ORER flag is 1.

[4]The receive data stored in FRDRL register is read until the amount of stored data is less than the FCR.RTRG[3:0] value. Software can check readable data in FDR.R[4:0].

[5]Serial reception continuation procedure:
To continue serial reception, before overrun error occurs, finish reading the receive data in FRDRL and clear the SSR_FIFO.RDF flag to 0.The FRDRL data can also be read by activating the DTC by an SCIn_RXI interrupt request. The RDF flag is cleared automatically in this case. Do not write to the RDF flag.

[6]Processing in response to an overrun error:
Read the FRDRL. In combination with step [7], this enables correct reception of the next frame possible.

[7]Clearing the error flag:
Write 0 to the error flag.

[8]Confirming that the error flag is cleared:
Read the error flag to confirm that its value is 0.

Note 1. It can also be read from FRDRH.ORER. However, to clear the ORER flag, write 0 to the associated bit in the SSR_FIFO register.
Note 2. It should be all receive data and is an integer times the FIFO triggering number.

Figure 24.44 Example flow of serial reception in clock synchronous mode with FIFO selected

24.5.6 Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode

(1) Non-FIFO selected

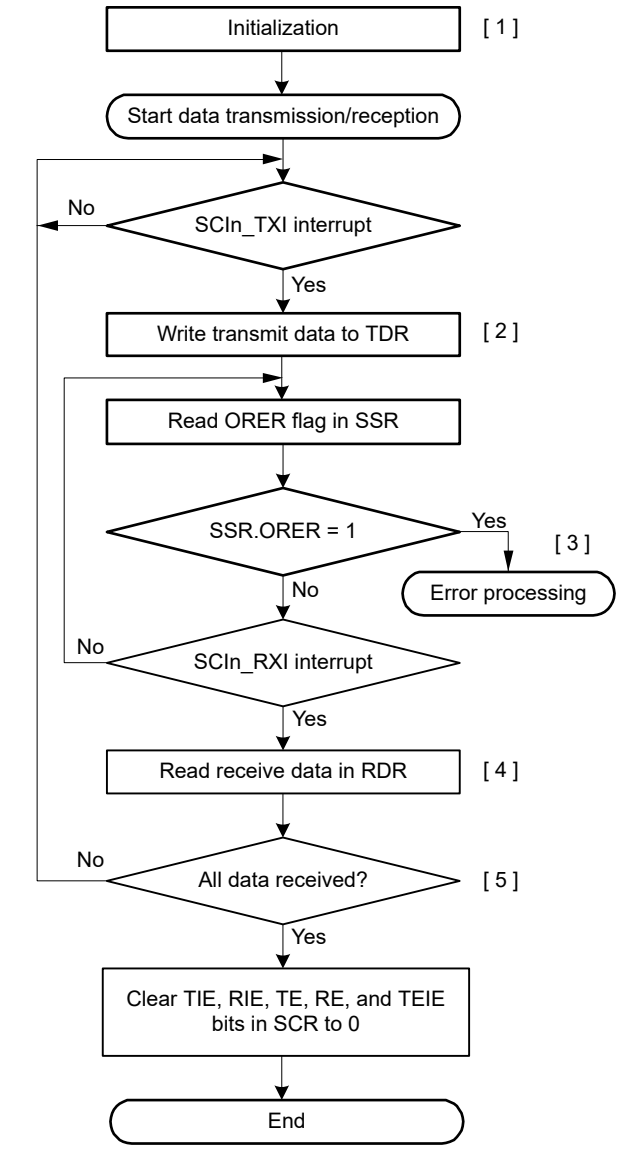
Figure 24.45 shows an example flow of simultaneous serial transmit and receive operations in clock synchronous mode. After initializing the SCI, use the following procedure for simultaneous serial data transmit and receive operations.

To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the data transmission by verifying that the SSR.TEND flag is set to 1.
2. Initialize the SCR register, and then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the data reception.
2. Set the RIE and RE bits to 0, and then check that the receive error flag ORER in the SSR register is 0.
3. Set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.



[1]SCI initialization:
The TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2]Transmit data write:
Write transmit data to TDR once in the SCIn_TXI interrupt request handling routine.

[3]Receive error processing:
If a receive error occurs, read the ORER flag in SSR, perform the relevant error processing, then set the ORER flag to 0. Data reception cannot resume while the ORER flag is 1.

[4]Reading receive data:
Read the receive data in RDR once in the SCIn_RXI interrupt request handling routine.

[5]Serial transmission/reception continuation procedure:
To continue serial transmission and reception, before the MSB bit 7 of the current frame is received, finish reading the receive data in RDR by the SCIn_RXI interrupt. Also, before the MSB bit 7 of the current frame is transmitted, write data to TDR by the SCIn_TXI interrupt.
Transmit data can also be written to TDR by activating the DTC by a transmit data empty interrupt (SCIn_TXI) request. Similarly, the RDR data can also be read by activating the DTC by a receive data full interrupt (SCIn_RXI) request

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first set the TIE, RIE, TE, RE, and TEIE bits in SCR to 0, then set TIE, RIE, TE, and RE bits to 1 simultaneously.

Figure 24.45 Example flow of simultaneous serial transmission and reception in clock synchronous mode with non-FIFO selected

(2) FIFO selected

Figure 24.46 shows an example flow of simultaneous serial transmit and receive operations in clock synchronous mode with FIFO selected.

After initializing the SCI, use the following procedure for simultaneous serial data transmit and receive operations.

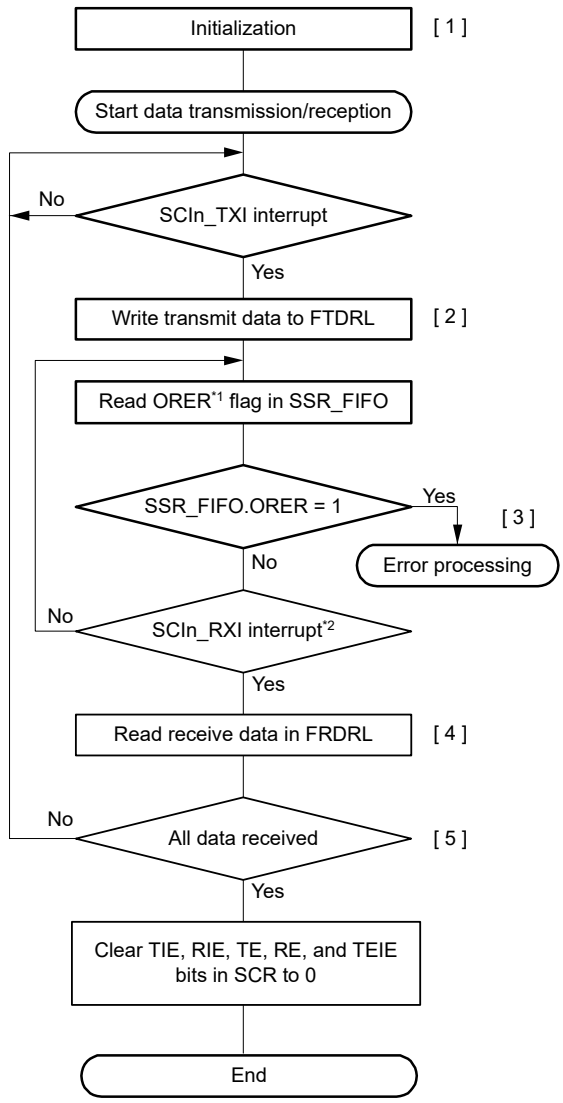
To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the transmission by verifying that the SSR_FIFO.TEND flag is set to 1.
2. Initialize the SCR register, then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the reception.

2. Set the RIE and RE bits to 0.
3. Check that the receive error flags ORER in the SSR_FIFO register are 0, and then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.



[1] SCI initialization:
The TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2] Transmit data write:
Write transmit data to FTDRL in the SCIn_TXI interrupt request handling routine. The number of transmission data that can be written is 16 bits, which is the number of stored transmit data in the FIFO.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR_FIFO, perform the relevant error processing, and then set the ORER flag to 0. Data reception cannot resume while the ORER flag is 1.

[4] Reading receive data:
The reception data stored in FRDRL register is read until the number of stored data is less than the FCR.RTRG value. Software can check readable data in FDR.R[4:0].

[5] Serial transmission/reception continuation procedure:
To continue serial reception, before overrun error occurs, finish reading the receive data in FRDRL and clear the SSR_FIFO.RDF flag to 0. To continue serial transmission, before transmission of the current transmit data is finished, write the next transmit data to FTDRL in the SCIn_TXI interrupt handling routine and clear the SSR_FIFO.TDFE flag to 0.
Transmit data can also be written to FTDRL by activating the DTC by a transmit FIFO data empty interrupt (SCIn_TXI) request. Similarly, the FRDRL data can also be read by activating the DTC by a receive FIFO data full interrupt (SCIn_RXI) request. The RDF and TDFE flags are cleared automatically in this case.
Do not write to the RDF and TDFE flags.

Note 1. It can also be read from FRDRH.ORER. To clear the ORER flag, write 0 to the associated bit in the SSR_FIFO register.
Note 2. It should be all receive data with amount an integer times FIFO triggering number.

Figure 24.46 Example flow of simultaneous serial transmission and reception in clock synchronous mode with FIFO selected

24.6 Operation in Smart Card Interface Mode

The SCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function of the SCI.

Smart card interface mode can be selected using the appropriate register.

24.6.1 Example Connection

Figure 24.47 shows an example connection between a smart card (IC card) and the MCU. As shown in Figure 24.47, because the MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.

Setting the SCR_SMCI.TE and SCR_SMCI.RE bits to 1 with an IC card disconnected enables closed-loop transmission or reception, allowing self-diagnosis. To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card.

An output port of the MCU can be used to output a reset signal.

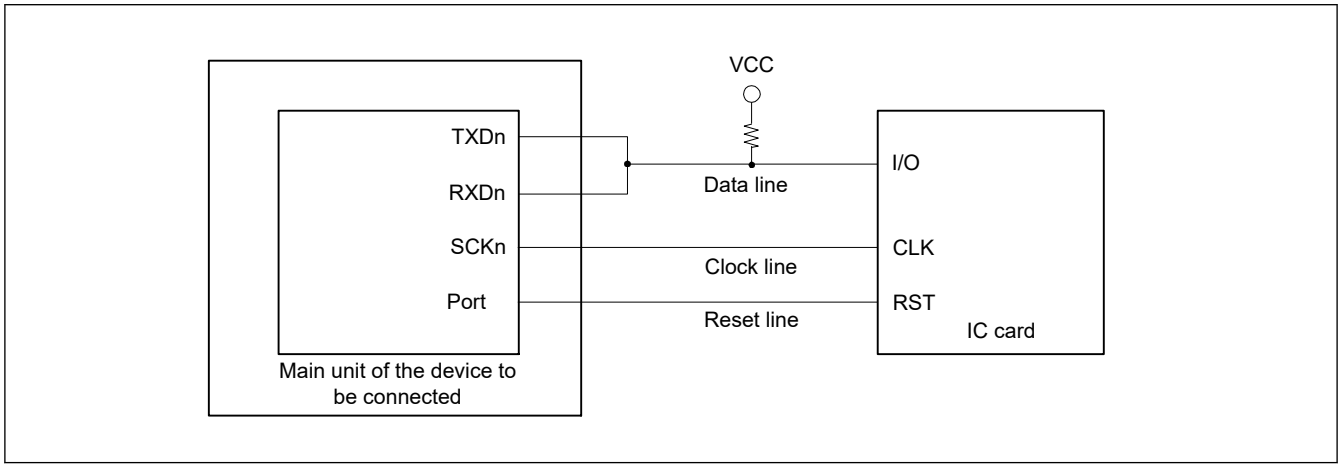


Figure 24.47 Example connection with a smart card (IC card)

24.6.2 Data Format (Except in Block Transfer Mode)

Figure 24.48 shows the data transfer formats in smart card interface mode:

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etus (elementary time unit – the time required for transferring 1 bit) is set as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low error signal is output for 1 etu after 10.5 etus elapse from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etus.

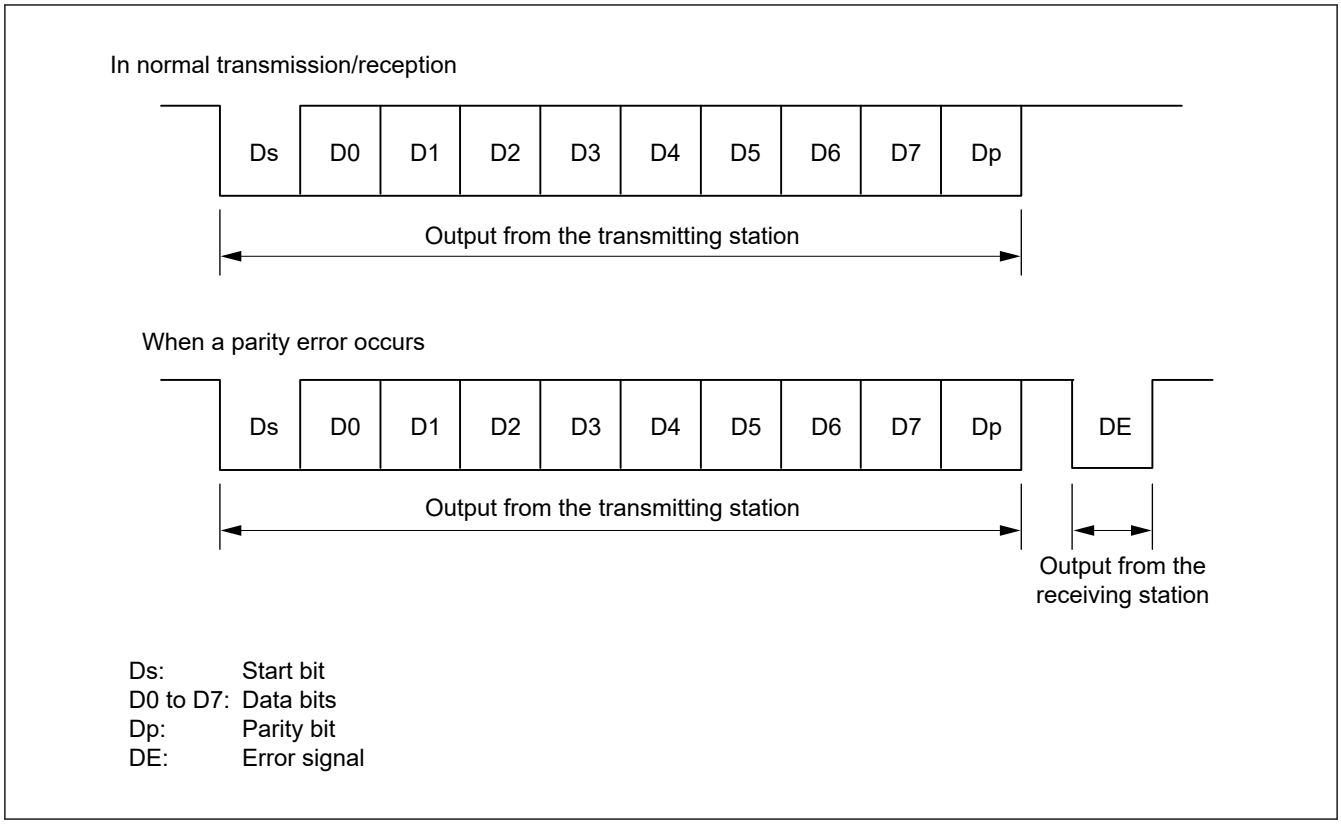


Figure 24.48 Data formats in smart card interface mode

For communications with IC cards of the direct convention type and inverse convention type, follow the procedures in this section.

(1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 indicate the Z and A states, respectively, and data is transferred with LSB-first for the start character, as shown in Figure 24.49. Therefore, data in the start character in the figure is 0x3B.

When using the direct convention type, write 0 to both the SCMR.SDIR and SCMR.SINV bits. Write 0 to the SMR_SMCI.PM bit to use even parity, which is prescribed by the smart card standard.

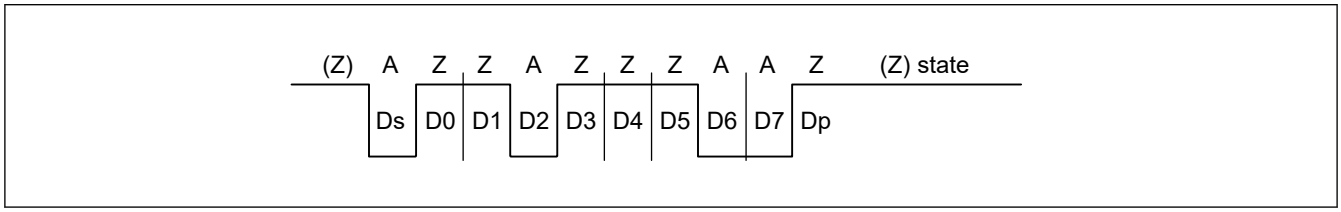


Figure 24.49 Direct convention with SDIR in SCMR = 0, SINV in SCMR = 0, and PM in SMR_SMCI = 0

(2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 indicate the A and Z states, respectively, and data is transferred with MSB-first for the start character, as shown in Figure 24.50. Therefore, data in the start character in the figure is 0x3F.

When using the inverse convention type, write 1 to both the SCMR.SDIR and SCMR.SINV bits. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to the Z state. Because the SINV bit of the MCU only inverts data bits D7 to D0, write 1 to the PM bit in SMR_SMCI to invert the parity bit for both transmission and reception.

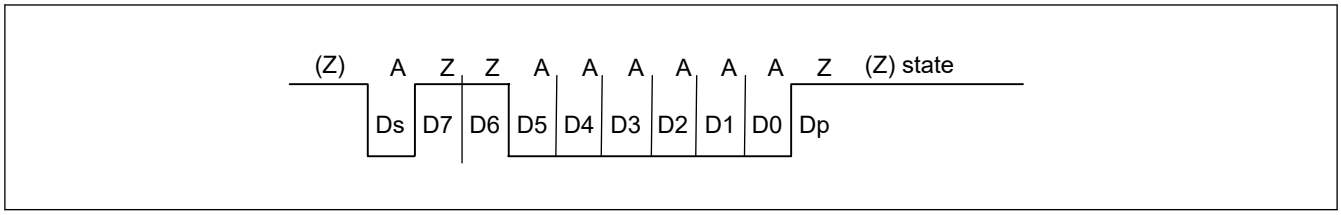


Figure 24.50 Inverse convention with SDIR in SCMR = 1, SINV in SCMR = 1, and PM in SMR_SMCI = 1

24.6.3 Block Transfer Mode

Block transfer mode differs from normal smart card interface mode as follows:

- Even if a parity error is detected during reception, no error signal is output. Because the PER flag in SSR_SMCI is set by error detection, clear the PER flag before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is set as a guard time from the end of the parity bit until the start of the next frame
- Because the same data is not retransmitted, the TEND flag in SSR_SMCI is set to 11.5 etus after transmission starts
- In block transfer mode, the ERS flag in SSR_SMCI indicates the error signal status as in normal smart card interface mode, but the flag is read as 0 because no error signal is transferred

24.6.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the on-chip baud rate generator can be used as a transfer clock in smart card interface mode.

In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate set up in the SCMR.BCP2 and the SMR_SMCI.BCP[1:0] bits. The frequency is always 16 times the bit rate in normal asynchronous mode.

For data reception, the falling edge of the start bit is sampled with the base clock to perform internal synchronization.

Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 24.51. The reception margin is determined by the following formula:

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ [\%]$$

M: Reception margin (%)

N: Ratio of bit rate to clock ($N = 32, 64, 372, 256$)

D: Duty cycle of clock ($D = 0$ to 1.0)

L: Frame length ($L = 10$)

F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the specified formula, the reception margin is determined using the following formula:

$$M = \{0.5 - 1/(2 \times 372)\} \times 100\ [\%] = 49.866\ \%$$

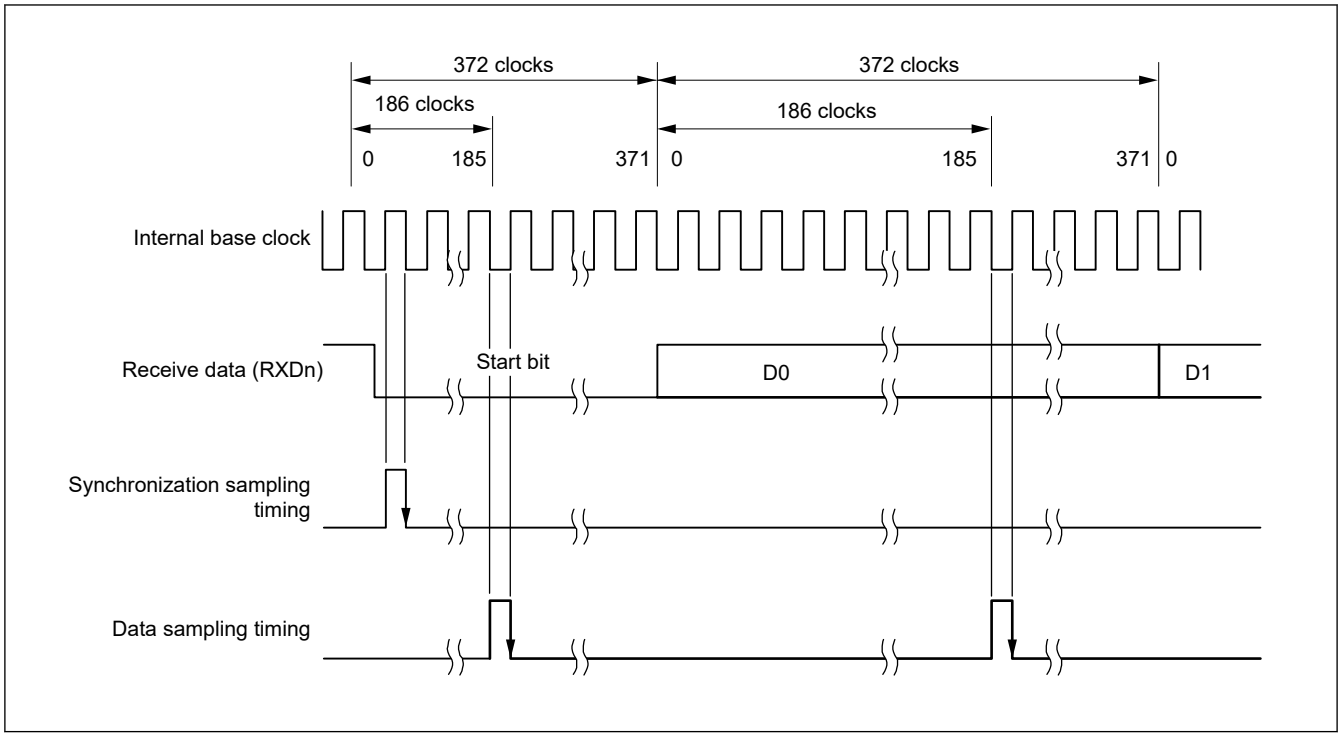


Figure 24.51 Receive data sampling timing in smart card interface mode when the clock frequency is 372 times the bit rate

24.6.5 SCI Initialization (Smart Card Interface Mode)

Before transmitting and receiving data, write the initial value 0x00 in the SCR_SMCI register and initialize the SCI following the example flow shown in Table 24.30.

Always set the initial value in the TIE, RIE, TE, RE, TEIE bits in the SCR_SMCI register before switching from transmission to reception mode or from reception to transmission mode. When SCR_SMCI.RE is set to 0, the RDR register is not initialized.

To change from reception mode to transmission mode, first check that reception has completed, then initialize the SCI. At the end of initialization, set SCR_SMCI.TE = 1 and SCR_SMCI.RE = 0. Reception completion can be verified by reading the SCIn_RXI request, ORER, or PER flag in SSR_SMCI.

To change transmission mode to reception mode, first check that transmission has completed, then initialize the SCI. At the end of initialization, set SCR_SMCI.TE = 0 and SCR_SMCI.RE = 1. Transmission completion can be verified by reading the TEND flag in SSR_SMCI.

Table 24.30 Example flow of SCI initialization in smart card interface mode (1 of 2)

No.	Step Name	Description
1	Start initialization	
2	Set SCR_SMCI.TIE, RIE, TE, RE, TEIE, and CKE[1:0] to 0	
3	Set I/O port functions	Set the I/O ports so that the necessary pin functions can be enabled among the TXDn, RXDn, and SCKn pins.
4	Set SSR_SMCI.ORER, ERS, PER to 0	Set the SSR_SMCI.ORER, ERS, and PER flags in to 0. After reading the SSR_SMCI register, write 0 to the target flags.
5	Set the SIMR1.IICM bit to 0 Set the SPMR.CKPH and CKPOL bits to 0	Set the SIMR1.IICM bit to 0, and set the SPMR.CKPH and CKPOL bits to 0. Skip this step when the initial values are not changed.
6	Set SMR_SMCI.GM, BLK, PM, BCP[1:0], CKS[1:0], and set SMR_SMCI.PE to 1	Set the operation mode and the transmission or reception format in SMR_SMCI.

Table 24.30 Example flow of SCI initialization in smart card interface mode (2 of 2)

No.	Step Name	Description
7	Set SCMR.BCP2, SDIR, SINV Set SCMR.SMIF to 1	Set the transmission or reception format in SCMR.
8	Set SEMR.BRME and SEMR.RXDESEL to 0	Set SEMR.BRME and SEMR.RXDESEL to 0.
9	Set a value in BRR	Write the value for the bit rate in BRR.
10	Set a value in the MDDR	Write the value obtained by correcting a bit rate error in the MDDR register. This step is not required if the bit rate adjustment function is not used.
11	Set a value in SCR_SMCI.CKE[1:0]	Set the SCR_SMCI.CKE[1:0] bits. When the CKE[0] bit is set to 0, the clock is output from the SCKn pin.
12	Set SCR_SMCI.TE or RE to 1, and set SCR_SMCI.TIE, RIE	Set the TE or RE bit in SCR_SMCI to 1, then set the TIE and RIE bits in SCR_SMCI. Do not simultaneously set the TE and RE bits to 1 if self-diagnosis is not used.
13	Initialization completed	

24.6.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode) is different from that in non-smart card interface mode, in that an error signal is sampled and data can be re-transmitted in smart card mode. Figure 24.52 shows the data re-transfer operation during transmission.

1. When an error signal from the receiver end is sampled after 1-frame data is transmitted, the SSR_SMCI.ERS flag is set to 1. If the SCR_SMCI.RIE bit is 1, an SCIn_ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.
2. For a frame in which an error signal is received, the SSR_SMCI.TEND flag is not set. Data is re-transferred from TDR to TSR, allowing automatic data retransmission.
3. If no error signal is returned from the receiver, the ERS flag is not set to 1.
4. In this case, the SCI determines that transmission of 1-frame data, including the re-transfer, is complete, and the TEND flag is set. If the SCR_SMCI.TIE bit is 1, an SCIn_TXI interrupt request is generated. Write transmit data to the TDR to start transmission of the next data.

Figure 24.54 shows an example flow of serial transmission. All the processing steps are automatically performed using an SCIn_TXI interrupt request to activate the DTC.

When the SSR_SMCI.TEND flag is set to 1 in transmission and when the SCR_SMCI.TIE bit is 1, an SCIn_TXI interrupt request is generated.

The DTC is activated by an SCIn_TXI interrupt request if the SCIn_TXI interrupt request is previously specified as a source of DTC activation, allowing the transfer of transmit data. The TEND flag is automatically set to 0 when the DTC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept at 0 and the DTC is not activated. Therefore, the SCI and DTC automatically transmit the specified number of bytes, including retransmission when an error occurs. Because the ERS flag is not automatically cleared, set the RIE bit to 1 before enabling an SCIn_ERI interrupt request to be generated if an error occurs, and clear the ERS flag to 0.

When transmitting or receiving data using the DTC, always enable the DTC before making the SCI settings.

For DTC settings, see section 15, Data Transfer Controller (DTC).

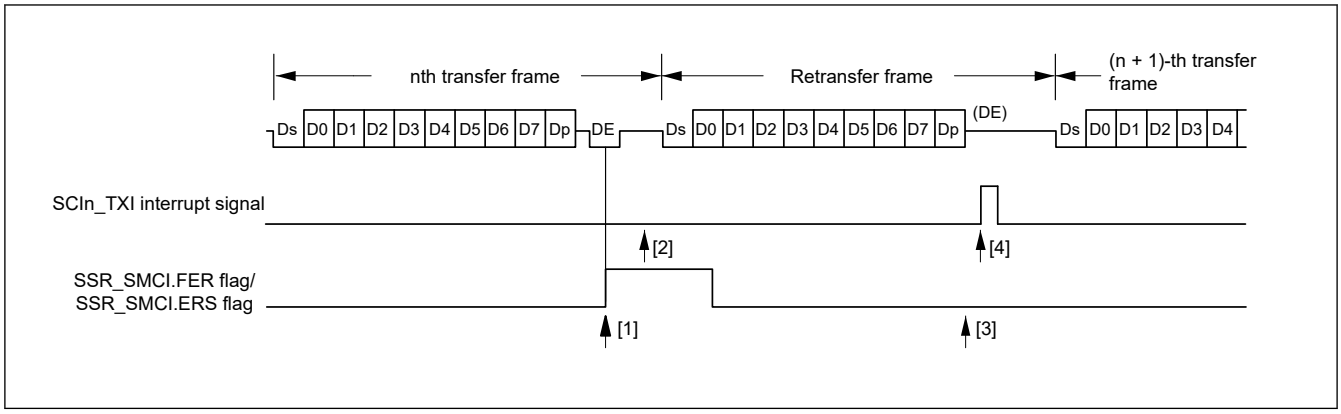


Figure 24.52 Data re-transfer operation in smart card interface transmission mode

The SSR_SMCI.TEND flag is set at different timings depending on the SMR_SMCI.GM bit setting. Figure 24.53 shows the TEND flag generation timing.

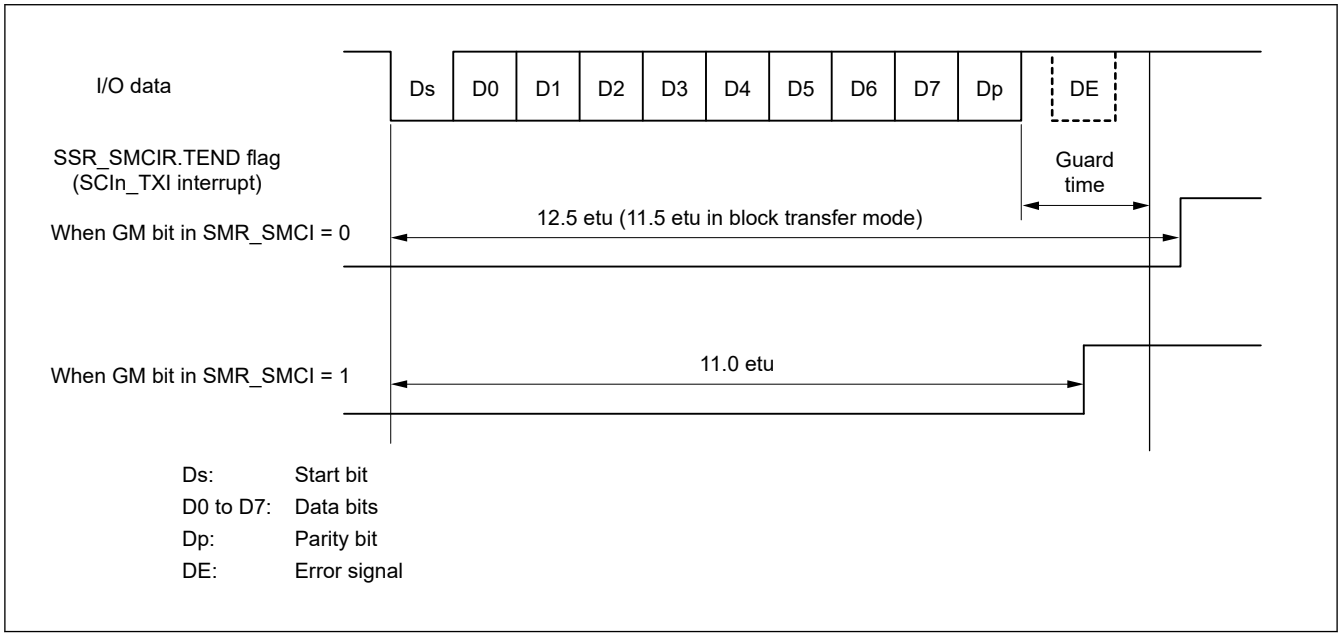


Figure 24.53 SSR.TEND flag generation timing during transmission

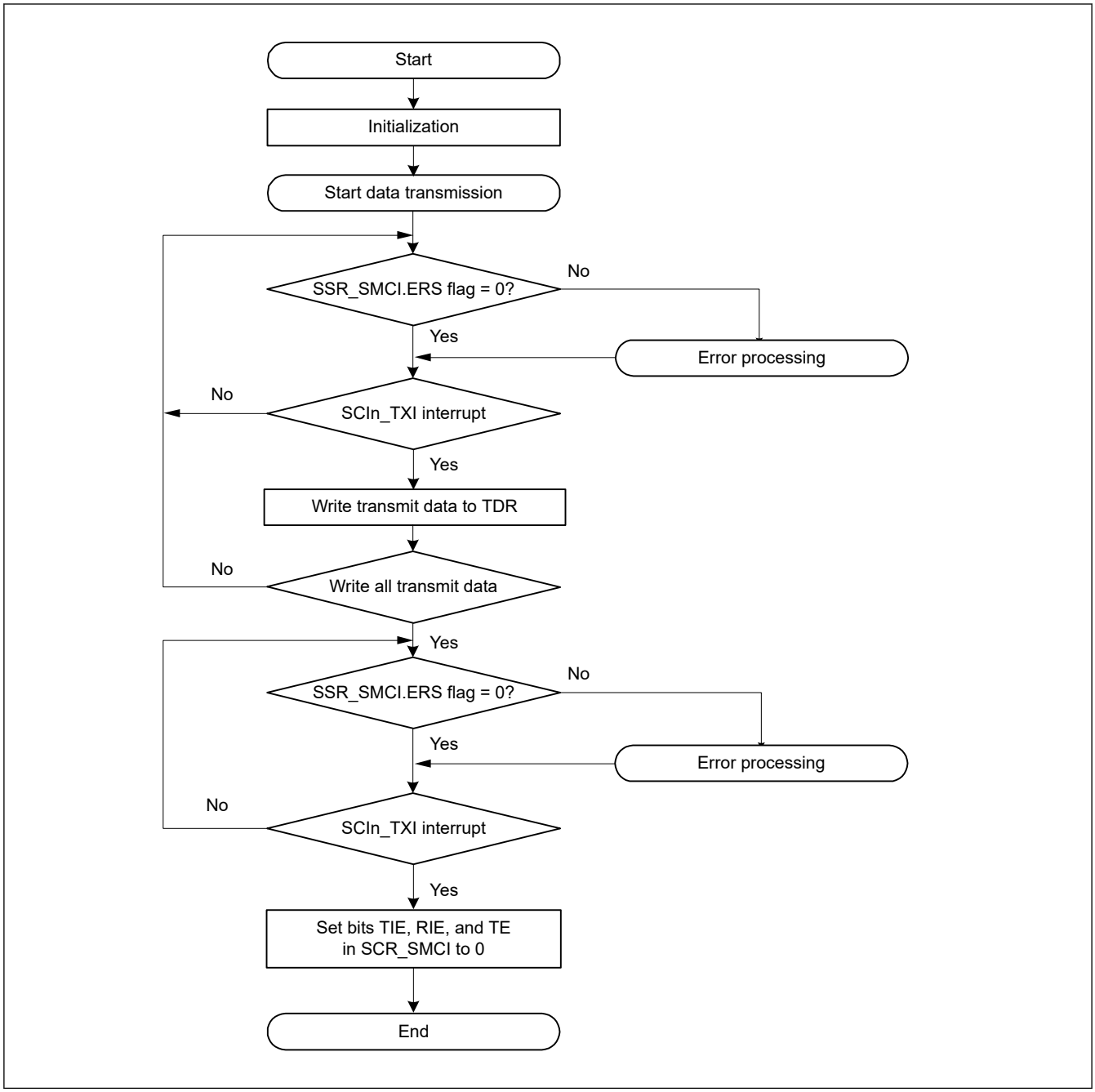


Figure 24.54 Example flow of smart card interface transmission

24.6.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. Figure 24.55 shows the data re-transfer operation in reception mode.

1. If a parity error is detected in the receive data, the SSR_SMCI.PER flag is set to 1. When the SCR_SMCI.RIE bit is 1, an SCIn_ERI interrupt request is generated. Clear the PER flag to 0 before the next parity bit is sampled.
2. For a frame in which a parity error is detected, no SCIn_RXI interrupt is generated.
3. When no parity error is detected, the SCR_SMCI.PER flag is not set to 1.
4. In this case, data is determined to be received successfully. When the SCR_SMCI.RIE bit is 1, an SCIn_RXI interrupt request is generated.

Figure 24.56 shows an example flow of serial data reception. All the processing steps are automatically performed using an SCIn_RXI interrupt request to activate the DTC.

In reception, setting the RIE bit to 1 allows an SCIn_RXI interrupt request to be generated. The DTC is activated by an SCIn_RXI interrupt request if the SCIn_RXI interrupt request is previously specified as a source of DTC activation, allowing the transfer of receive data.

If an error occurs during reception and either the ORER or PER flag in SSR_SMCI is set to 1, a receive error interrupt (SCIn_ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DTC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC is transferred.

If a parity error occurs and the PER flag is set to 1 during reception, the receive data is transferred to RDR, allowing the data to be read.

When a reception is forced to terminate by setting SCR_SMCI.RE to 0 during operation, read the RDR register because the received data that is not yet read might be left in the RDR.

Note: For operations in block transfer mode, see section 24.3.9. Serial Data Reception in Asynchronous Mode.

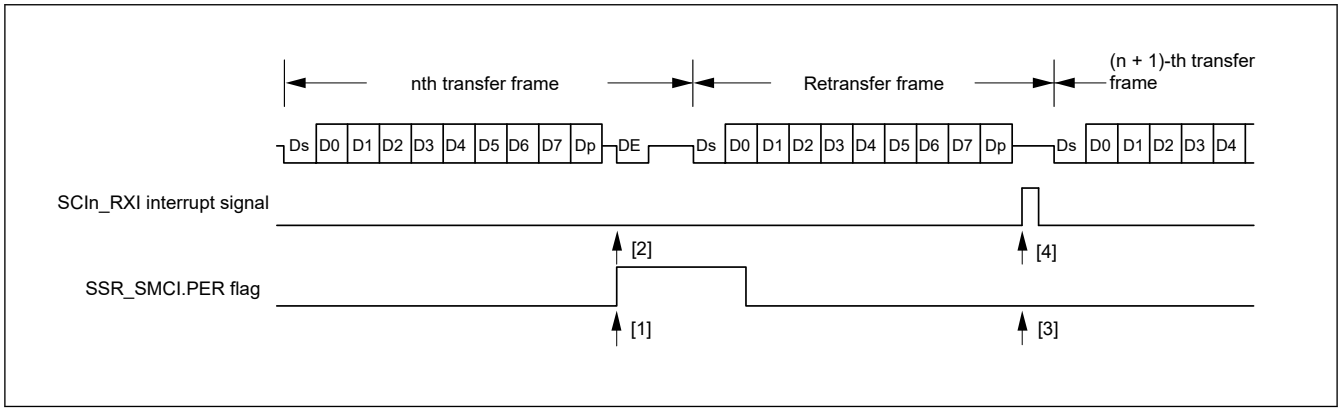


Figure 24.55 Data re-transfer operation in smart card interface reception mode

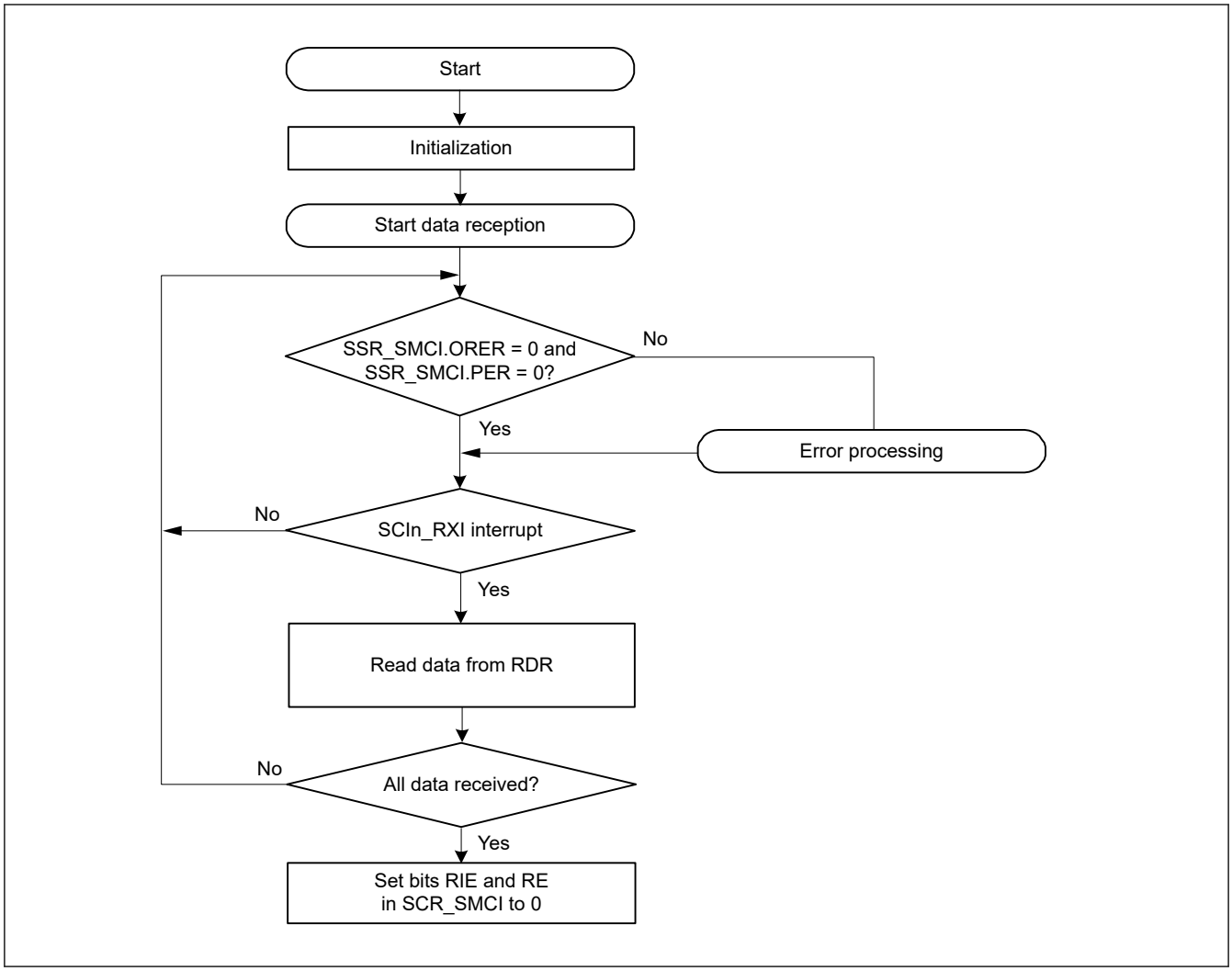


Figure 24.56 Example flow of smart card interface reception

24.6.8 Clock Output Control

When the GM bit in SMR_SMCI is set to 1, the clock output can be controlled by the CKE[1:0] bits in SCR_SMCI. For details on the CKE[1:0] bits, see section 24.2.12. SCR_SMCI : Serial Control Register for Smart Card Interface Mode (SCMR.SMIF = 1). When setting the clock output, the base clock described in section 24.6.4. Receive Data Sampling Timing and Reception Margin is applied.

Figure 24.57 shows an example timing for the clock output control when the CKE[1] bit in SCR_SMCI is set to 0 and the CKE[0] bit in SCR_SMCI is controlled.

When the GM bit in SMR_SMCI is 0, output control by the CKE[0] bit in SCR_SMCI is immediately reflected on the SCKn pin, so there is a possibility that pulses with an unintended width may be output from the SCKn pin.

When the GM bit in SMR_SMCI is 1, the clock with the same pulse width as the base clock is output even if the CKE[0] bit in SCR_SMCI is changed.

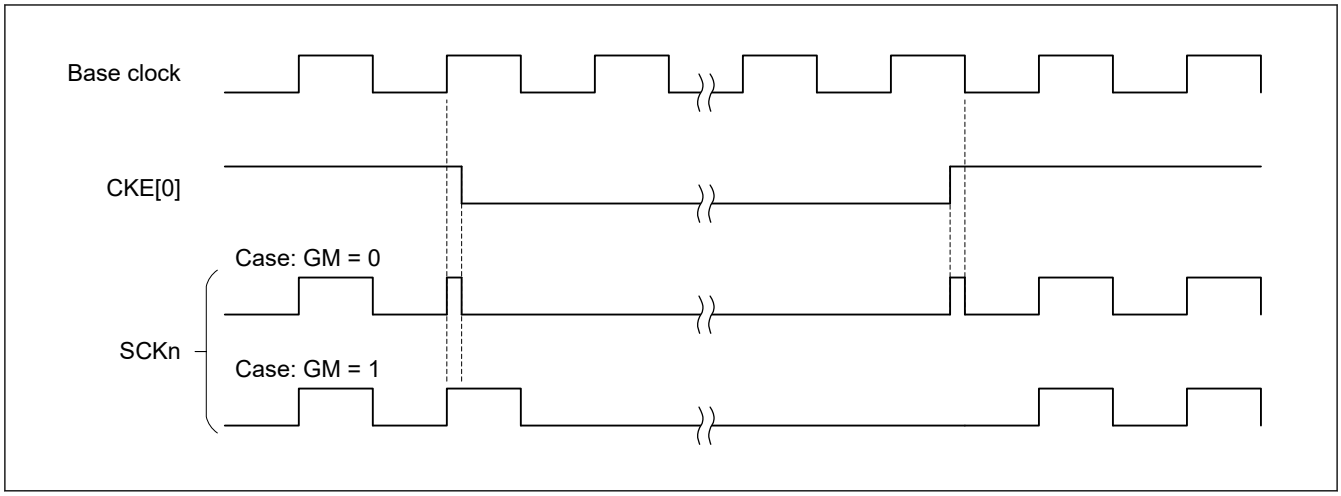


Figure 24.57 Clock Output timing

24.7 Operation in Simple IIC Mode

Simple IIC mode format is composed of 8 data bits and an acknowledge bit. By continuing into a slave-address frame after a start condition or restart condition, a master device can specify a slave device as the partner for communications. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied. The 8 data bits in all frames are transmitted in order from the MSB.

The I^2C bus format and timing of the I^2C bus are shown in Figure 24.58 and Figure 24.59.

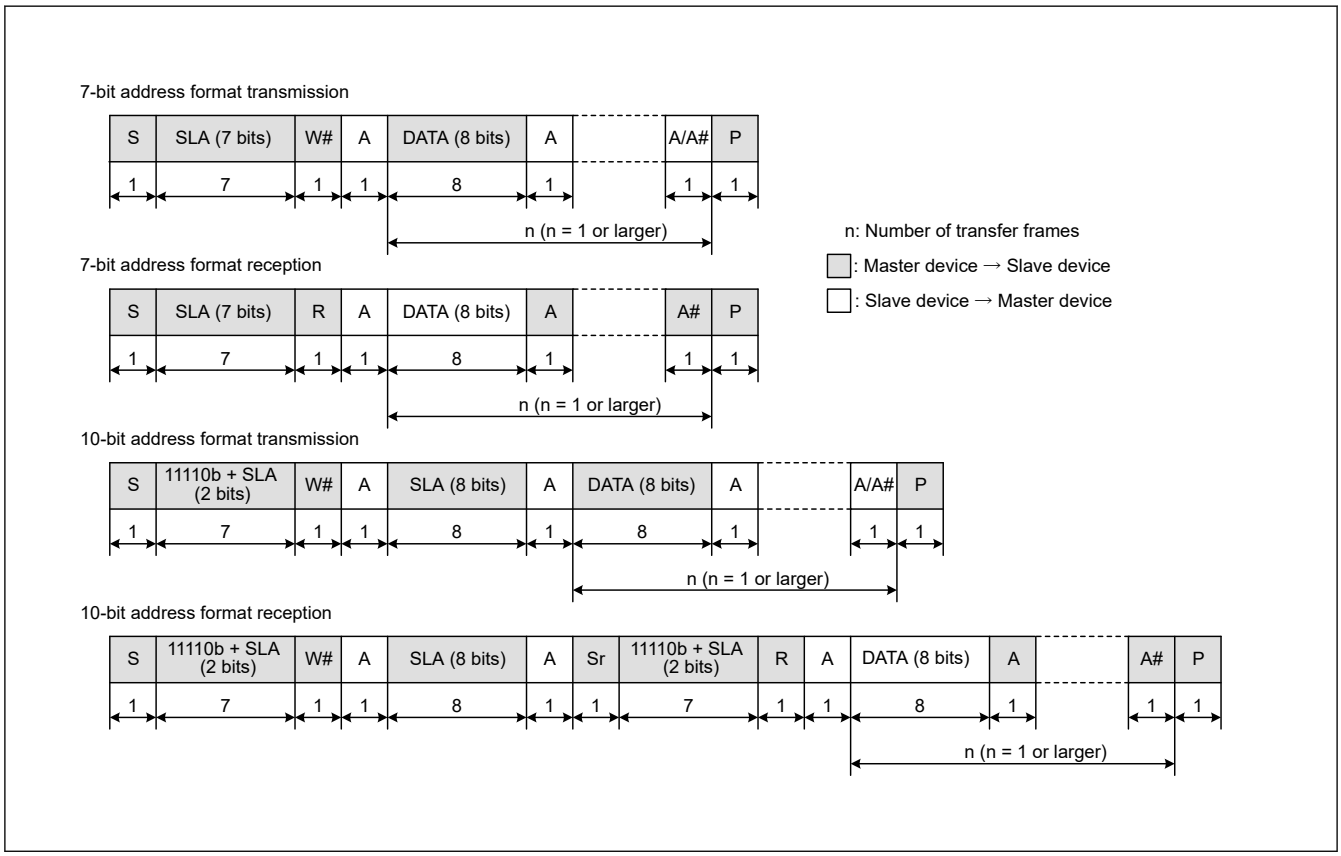


Figure 24.58 I^2C bus format

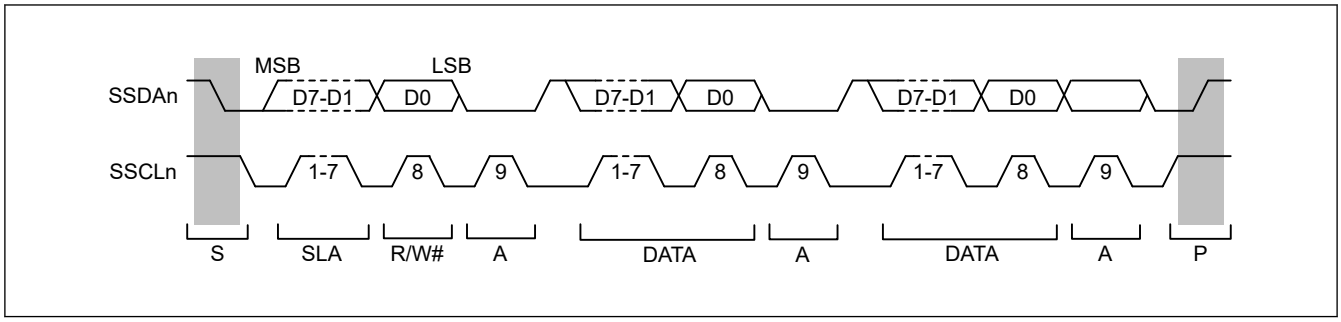


Figure 24.59 I²C bus timing when SLA is 7 bits

- S: Indicates a start condition, when the master device changes the level on the SDAn line from high to low while the SCLn line is high
- SLA: Indicates a slave address, by which the master device selects a slave device
- R/W#: Indicates the direction of transfer (reception or transmission). The value 1 indicates transfer from the slave device to the master device and 0 indicates transfer from the master device to the slave device.
- A/A#: Indicates an acknowledge bit. This is returned by the slave device for master transmission and by the master device for master reception. Return low indicates ACK and return high indicates NACK.
- Sr: Indicates a restart condition, when the master device changes the level on the SDAn line from high to low while the SCLn line is high and after the setup time elapses
- DATA: Indicates the data being received or transmitted
- P: Indicates a stop condition, when the master device changes the level on the SDAn line from low to high while the SCLn line is high

24.7.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the SIMR3.IICSTAREQ bit causes the generation of a start condition. The generation of a start condition proceeds through the following operations:

- The level on the SDAn line falls (from the high level to the low level) and the SCLn line is kept in the released state
- The hold time for the start condition is set as half of a bit period at the bit rate determined by the BRR setting
- The level on the SCLn line falls (from the high level to the low level), the IICSTAREQ bit in SIMR3 is set to 0, and a start-condition generated interrupt is output

Writing 1 to the IICRSTAREQ bit in SIMR3 causes the generation of a restart condition. The generation of a restart condition proceeds through the following operations:

- The SDAn line is released and the SCLn line is kept at the low level
- The period at low level for the SCLn line is set as half of a bit period at the bit rate determined by the BRR setting
- The SCLn line is released (transition from the low to the high level)
- When a high level is detected on the SCLn line, the setup time for the restart condition is set as half of a bit period at the bit rate determined by the BRR setting
- The level on the SDAn line falls (from the high level to the low level)
- The hold time for the restart condition is set as half of a bit period at the bit rate determined by the BRR setting
- The level on the SCLn line falls (from the high level to the low level), the SIMR3.IICRSTAREQ bit is set to 0, and a restart-condition generated interrupt is output

Writing 1 to the SIMR3.IICSTPREQ bit causes the generation of a stop condition. The generation of a stop condition proceeds through the following operations:

- The level on the SDAn line falls (from the high level to the low level) and the SCLn line is kept at the low level
- The period at low level for the SCLn line is set as half of a bit period at the bit rate determined by the BRR setting
- The SCLn line is released (transition from the low to the high level)

- When a high level is detected on the SCLn line, the setup time for the stop condition is set as half of a bit period at the bit rate determined by the BRR setting
- The SDAn line is released (transition from the low to the high level), the SIMR3.IICSTPREQ bit is set to 0, and a stop-condition generated interrupt is output

Figure 24.60 shows the timing of operations in the generation of start, restart, and stop conditions.

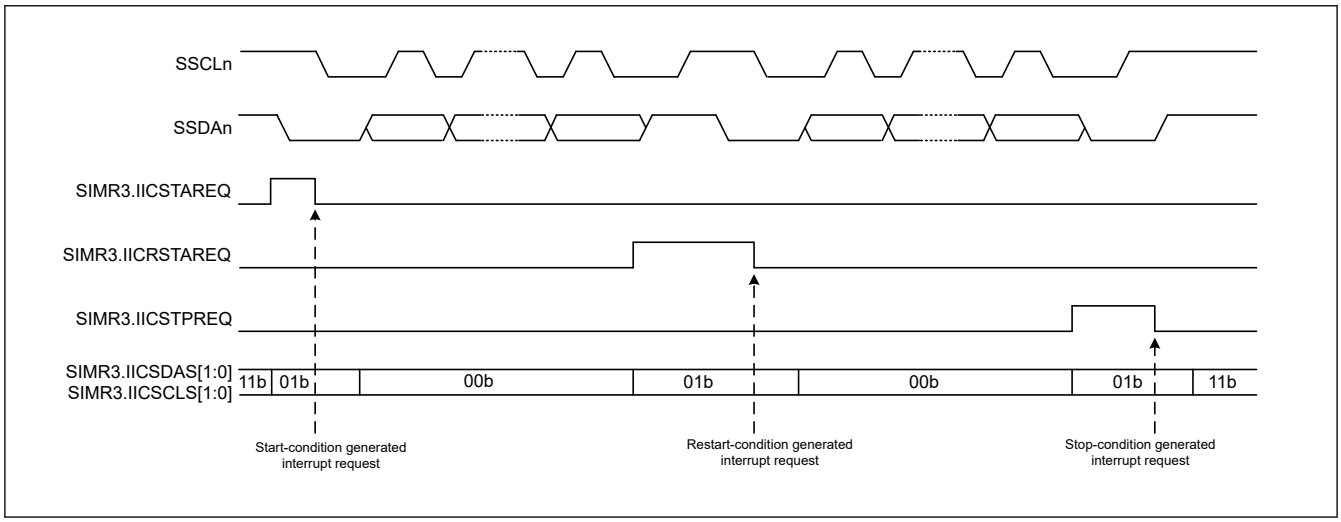


Figure 24.60 Timing of operations in generation of start, restart, and stop conditions

24.7.2 Clock Synchronization

The SCLn line can be driven low if a wait is inserted by a slave device at the other side of the transfer. Setting the SIMR2.IICCSC bit to 1 applies control to obtain synchronization when a difference arises between the levels of the internal SCLn clock signal and the level being input on the SCLn pin.

When the SIMR2.IICCSC bit is set to 1, the level of the internal SCLn clock signal changes from low to high. Counting to determine the period at a high level stops while the low level is being input on the SCLn pin. Counting to determine the period at a high level starts after the transition of the input on the SCLn pin to the high level.

The interval from this time until counting to determine the period at high level starts on the transition of the SCLn pin to the high level, is the total of the delay of SCLn output, delay for noise filtering of the input on the SCLn pin (2 or 3 cycles of sampling clock for the noise filter), and delay for internal processing (1 or 2 cycles of PCLK). The period at high level of the internal SCLn clock is extended even when other devices do not place the low level on the SCLn line.

If the SIMR2.IICCSC bit is 1, synchronization is obtained for the transmission and reception of data by taking the logical AND of the input on the SCLn pin and the internal SCLn clock. If the SIMR2.IICCSC bit is 0, synchronization with the internal SCLn clock is obtained for the transmission and reception of data.

If a slave device inserts a wait period into the interval until the transition of the internal SCLn clock signal from the low to the high level after a request for the generation of a start, restart, or stop condition is issued, the time until generation is prolonged by that period.

If a slave device inserts a wait period after the transition of the internal SCLn clock signal from the low to the high level, although the generation-completed interrupt is issued without stopping the waiting period, generation of the condition itself is not guaranteed.

Figure 24.61 shows an example operation for synchronizing the clocks.

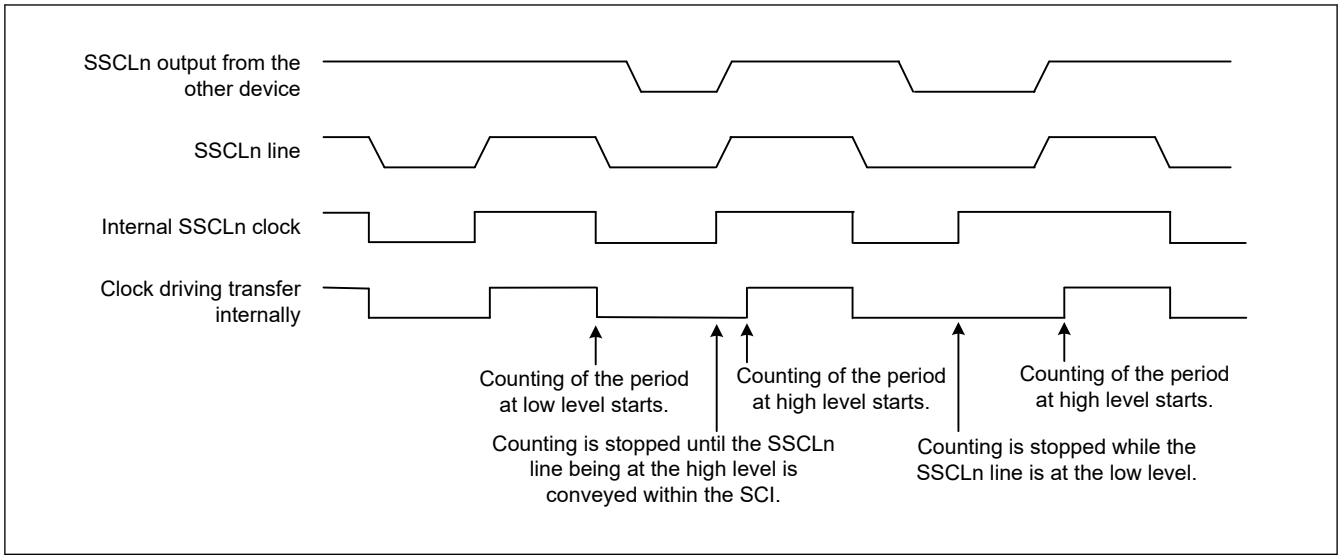


Figure 24.61 Example operations for clock synchronization

24.7.3 SDAn Output Delay

The SIMR1.IICDL[4:0] bits can be used to set a delay for output on the SDAn pin relative to falling edges of output on the SCLn pin. Delay settings from 0 to 31 are selectable, representing periods of the corresponding numbers of cycles of the clock signal from the on-chip baud rate generator (derived by frequency-dividing the base clock, PCLK, by the divisor selected in the SMR.CKS[1:0] bits). A delay for output on the SDAn pin applies to the start condition/restart condition/stop condition signal, 8-bit transmit data, and acknowledge bit.

If the SDAn output delay is shorter than the time for the level on the SCLn pin to fall, the change of the output on the SDAn pin starts while the output level on the SCLn pin is falling, creating a possibility of erroneous operation for slave devices. Ensure that settings for the delay of output on the SDAn pin specify times greater than the time output on the SCLn pin takes to fall (300 ns for IIC in normal mode and fast mode).

Figure 24.62 shows the timing of delays in SDAn output.

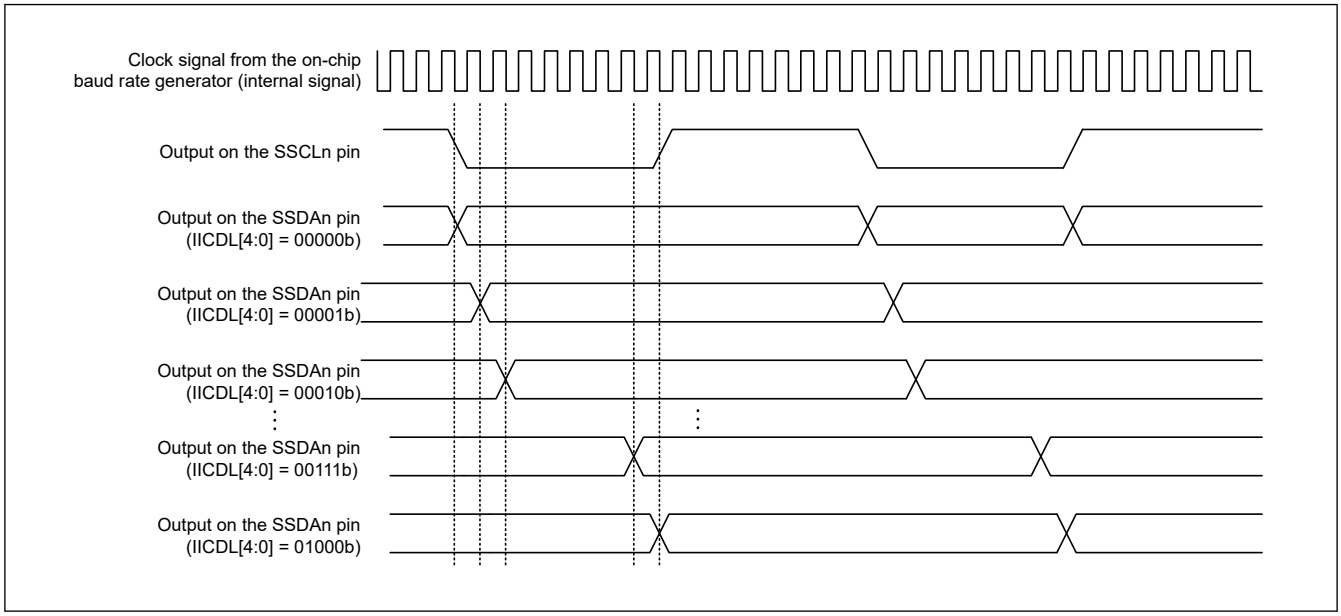


Figure 24.62 Timing of delays in SDAn output

24.7.4 SCI Initialization in Simple IIC Mode

Before transferring data, write the initial value 0x00 to SCR and initialize the interface following the example shown in Table 24.31.

Before making any changes to the operating mode or transfer format, be sure to set SCR to its initial value. In simple IIC mode, the open-drain setting for the communication ports should be made on the port side.

Table 24.31 Example flow of SCI initialization in simple IIC mode

No.	Step Name	Description
1	Start of initialization	
2	Set the TIE, RIE, TE, RE, TEIE and CKE[1:0] bits in SCR to 0	
3	Set the I/O port functions	Set the I/O port to allow use (on N-channel open-drain output pins) of the SSCLn and SSDAn pin functions.
4	Set the IICSDAS[1:0] and IICSCLS[1:0] bits in SIMR3 to 11b	Place the SSCLn and SSDAn pins in the high-impedance state until a start condition is to be generated.
5	Set up the transfer or reception format in SMR and SCMR	Set the format for transmission and reception in SMR and SCMR. In SMR, set the CKS[1:0] bits to the target value and set the other bits to 0. In SCMR, set the SDIR bit to 1 and the SINV and SMIF bits to 0.
6	Set the value in BRR	Write the value for the targeted bit rate to BRR.
7	Set a value in MDDR	Write the value obtained by correcting a bit rate error in MDDR. This step is not required if the BRME bit in SEMR is set to 0.
8	Set the values in SEMR, SNFR, SIMR1, SIMR2, and SPMR	Set the values in SEMR, SNFR, SIMR1, SIMR2, and SPMR. Set the NFEN and BRME bits in SEMR. In SNFR, set the NFCS[2:0] bits. In SIMR1, set the IICM bit to 1 and the IICDL[4:0] bits as required. In SIMR2, set the IICACKT and IICCSC bits to 1 and the IICINTM bits as required. In SPMR, set all the bits to 0.
9	Set the SCR.RE and TE bit to 1 and set the SCR.TIE, RIE and TEIE bits	Set the RE and TE bits in the SCR to 1. Then, set the SCR.TIE, RIE, and TEIE bits (for transmission and when the SIMR2.IICINTM bit is 1, set the RIE bit to 0). Setting the TE and RE bits to 1 enables the SSCLn and SSDAn pin functions.
10	Start of transmission or reception	

24.7.5 Operation in Master Transmission in Simple IIC Mode

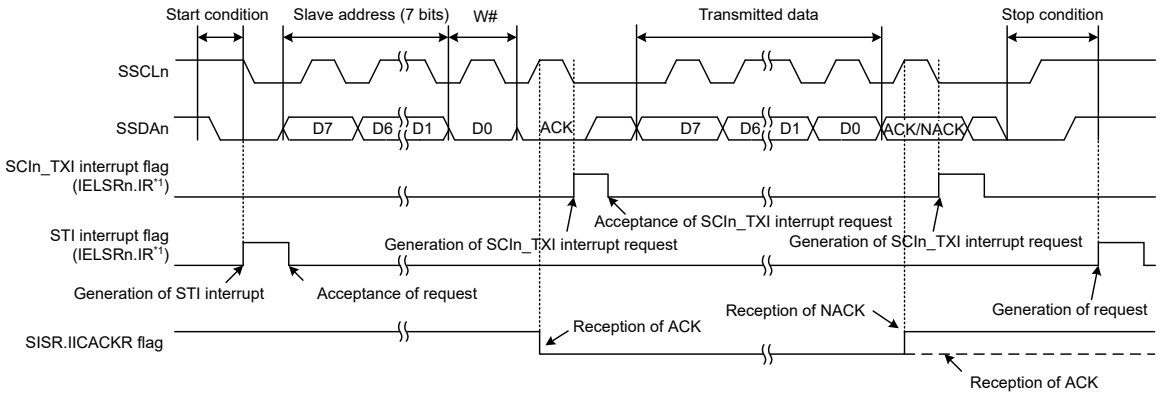
Figure 24.63 and Figure 24.64 show examples of master transmission and Figure 24.65 shows an example flow of data transmission.

Figure 24.63 shows the operation example when SIMR2.IICINTM bit is 1 (use reception and transmission interrupts) and the value of the SCR.RIE bit is assumed to be 0 (SCIn_RXI and SCIn_ERI interrupt requests are disabled).

See Table 24.36 for more information on the STI interrupt.

Figure 24.65 shows a flow chart in the case of SIMR2.IICINTM is 1 and address transmission by CPU and data transmission by DTC. When 10-bit slave addresses are in use, steps [3] and [4] are repeated twice.

In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame is complete, unlike the SCIn_TXI interrupt request generation timing during clock synchronous transmission.



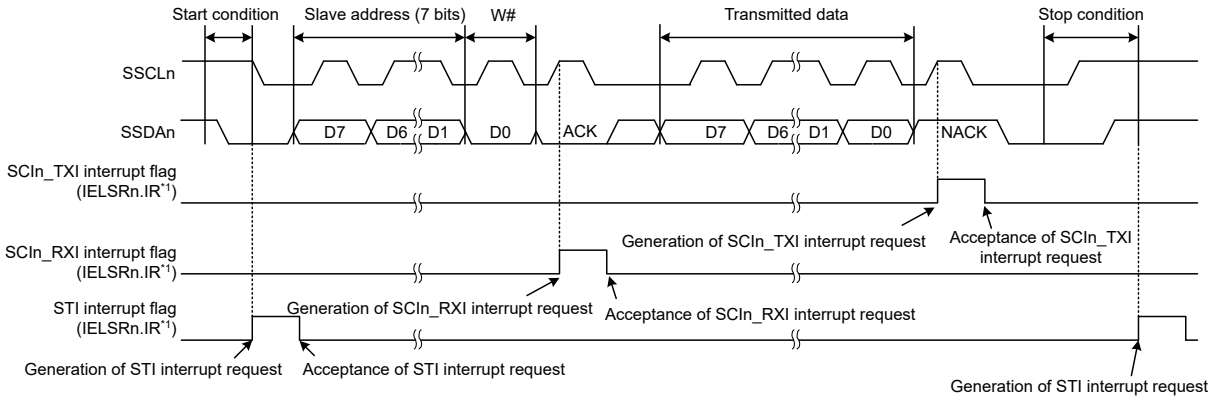
Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 24.63 Example 1 of operations for master transmission in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts

When the SIMR2.IICINTM bit is set to 0 (use ACK/NACK interrupts) during master transmission, the DTC is activated by the ACK interrupt as the trigger and required number of data bytes are transmitted. When the NACK is received, error processing such as transmission stop and retransmission is performed using the NACK interrupt as the trigger.

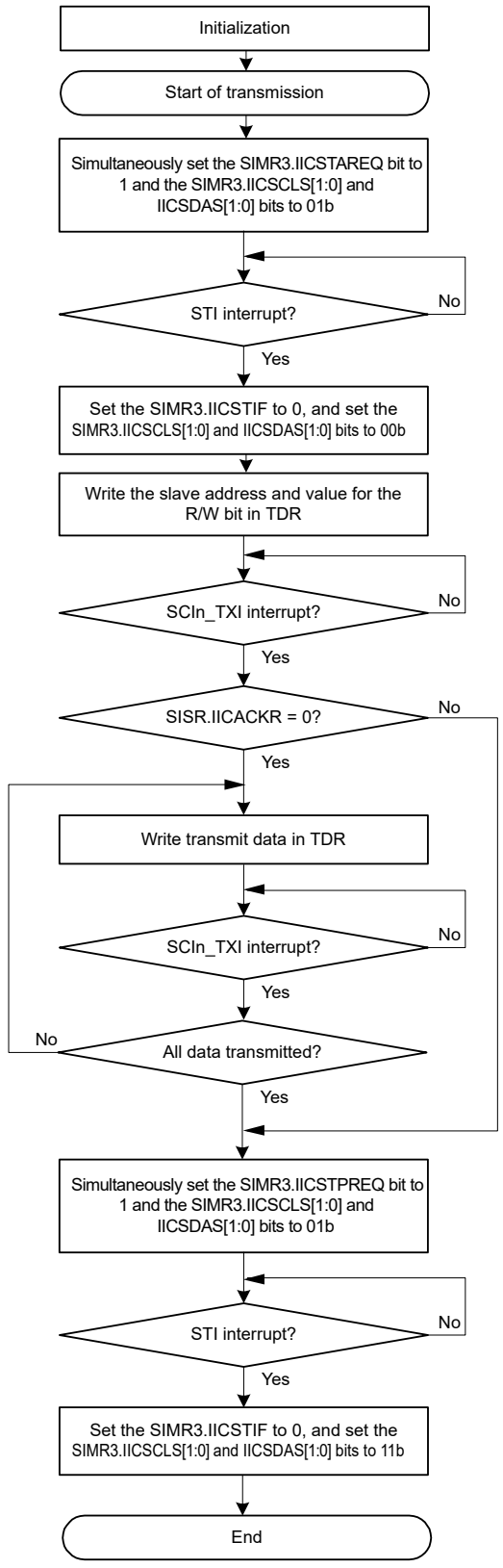
To restart communication for some reason after writing data in the TDR register, use the following procedure:

1. Set the TE and RE bits in the SCR register to 0 to stop communication.
2. Set 0xF0 in the SIMR3 register, release the I^2C bus, and clear the generation of a condition.
3. If the RDRF flag in the SSR register is set to 1, clear it.
4. Set the TE and RE bits in the SCR register to 1 and start the next communication.



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 24.64 Example 2 of operations for master transmission in simple IIC mode with 7-bit slave addresses, ACK interrupts, and NACK interrupts



[1] [1] Initialization in simple IIC mode:
For transmission, set the SCR.RIE to 0 to disable the RXI and ERI interrupts.

[2] [2] Generate a start condition.

[3] [3] Writing to TDR:
Write the slave address and value for the R/W bit to TDR.

[4] [4] Confirming ACK response from the slave device:
Check the SISR.IICACKR bit. If SISR.IICACKR is 0, it indicates that the slave device responded with ACK and operations proceed. If SISR.IICACKR is 1, it indicates that there was no response from the slave device so the next transition is to generate a stop condition.

[4] If 10-bit slave addresses are in use, processing of [3] and [4] is repeated twice.

[5] [5] Continuing with serial transmission:
When transmission is to continue, write additional transmit data to TDR. Except for the first data to be transmitted, a TXI interrupt request can activate the DTC to handle writing of data to TDR.

[6] [6] Generate a stop condition.

Note: In simple IIC mode, the SCIn_TXI interrupt is generated when communication completes.

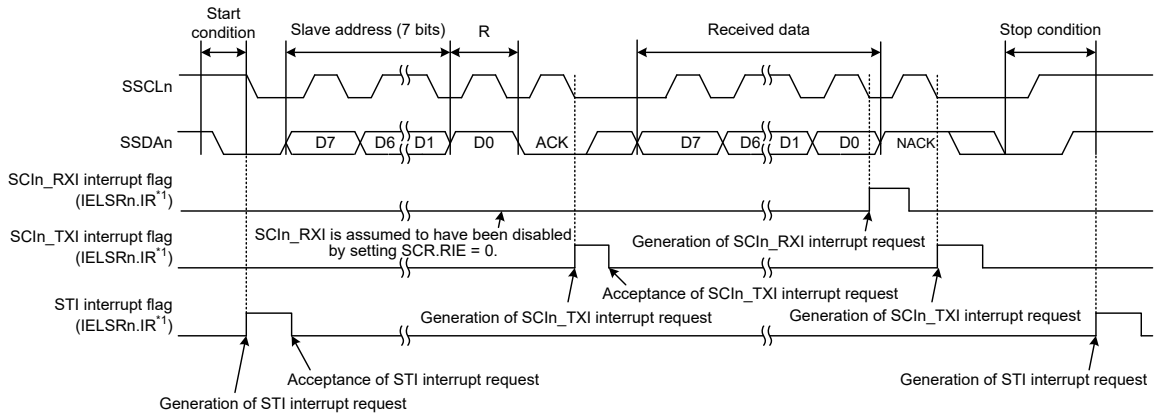
Figure 24.65 Example flow of master transmission in simple IIC mode with transmission interrupts and reception interrupts

24.7.6 Master Reception in Simple IIC Mode

Figure 24.66 shows an example operation in simple IIC mode master reception and Figure 24.67 shows an example flow of master reception.

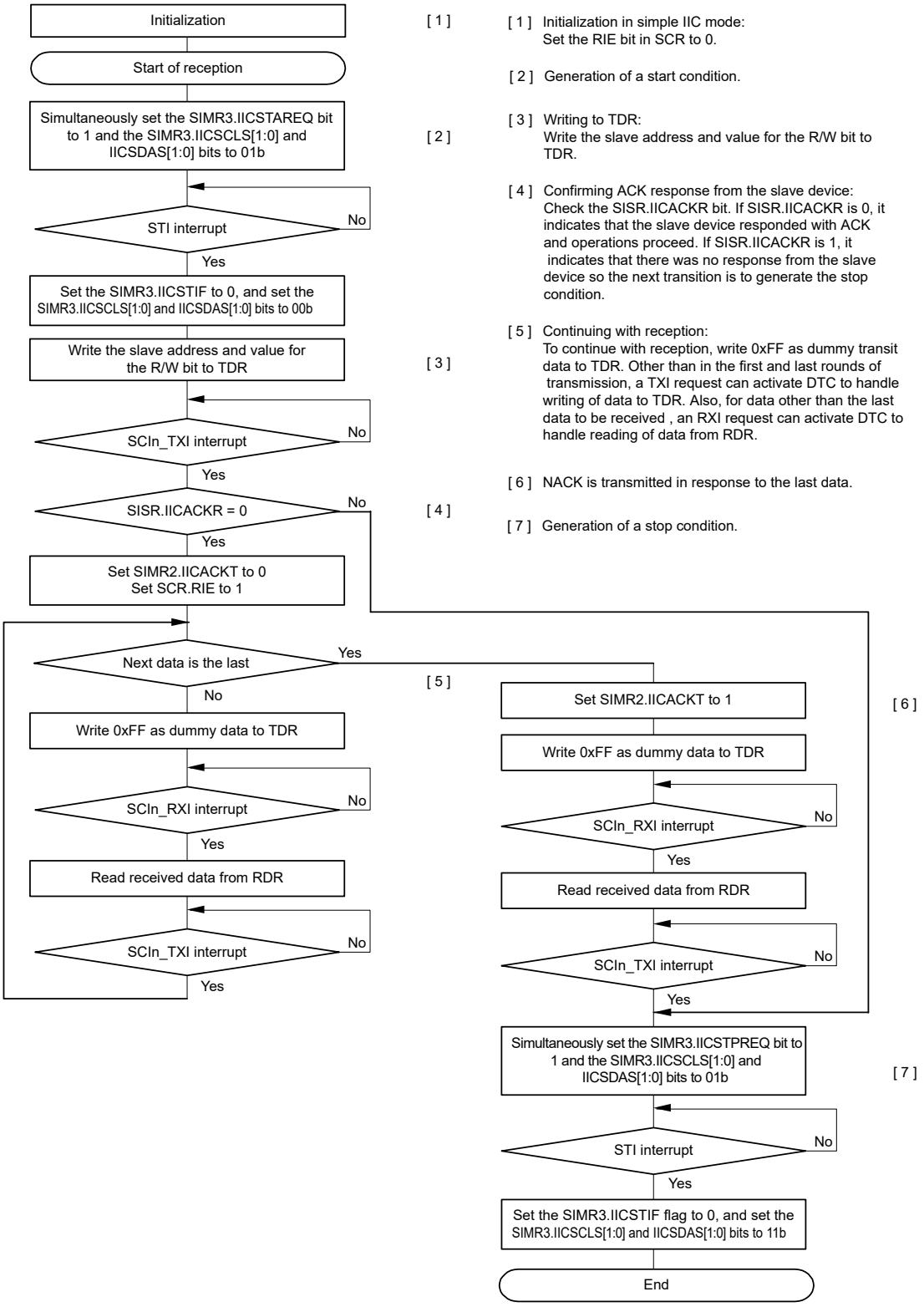
The value of the SIMR2.IICINTM bit is assumed to be 1 (use reception and transmission interrupts).

In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame is complete, unlike the SCIn_TXI interrupt request generation timing during clock synchronous transmission.



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 24.66 Example operations for master reception in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts



[1] Initialization in simple IIC mode:
Set the RIE bit in SCR to 0.

[2] Generation of a start condition.

[3] Writing to TDR:
Write the slave address and value for the R/W bit to TDR.

[4] Confirming ACK response from the slave device:
Check the SISR.IICACKR bit. If SISR.IICACKR is 0, it indicates that the slave device responded with ACK and operations proceed. If SISR.IICACKR is 1, it indicates that there was no response from the slave device so the next transition is to generate the stop condition.

[5] Continuing with reception:
To continue with reception, write 0xFF as dummy transit data to TDR. Other than in the first and last rounds of transmission, a TXI request can activate DTC to handle writing of data to TDR. Also, for data other than the last data to be received , an RXI request can activate DTC to handle reading of data from RDR.

[6] NACK is transmitted in response to the last data.

[7] Generation of a stop condition.

Note: In simple IIC mode, the SCIn_TXI interrupt request is generated when communication is complete.

Figure 24.67 Example flow of master reception in simple IIC mode with transmission interrupts and reception interrupts

24.8 Operation in Simple SPI Mode

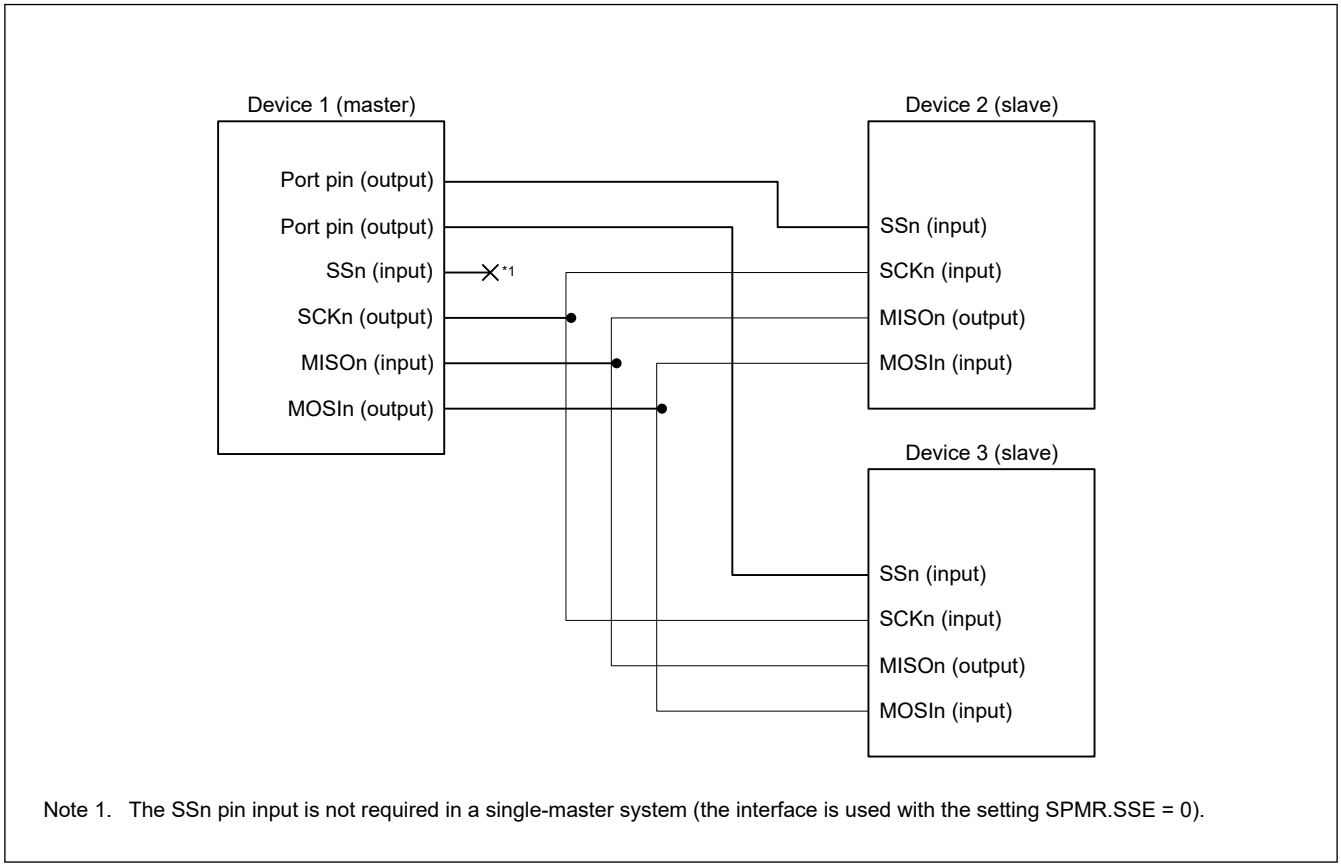
As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Using the settings for clock synchronous mode (SCMR.SMIF = 0, SIMR1.IICM = 0, SMR.CM = 1) and setting the SPMR.SSE bit to 1 place the SCI in simple SPI mode. However, the SSn pin function on the master side is not required for connection of the device used as the master in simple SPI mode when the configuration only has a single master. Therefore, set the SPMR.SSE bit to 0 in such cases.

Figure 24.68 shows an example of connections for simple SPI mode. Control a general port pin to produce the SSn output signal from the master.

In simple SPI mode, data is transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended. The data can be inverted by setting the SCMR.SINV bit to 1.

Because the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a shared clock signal. Additionally, because both the transmitter and receiver have a buffered structure, writing the next transmit data while transmission is in progress and reading previously received data while reception is in progress are both possible. This enables continuous transfer.



Note 1. The SSn pin input is not required in a single-master system (the interface is used with the setting SPMR.SSE = 0).

Figure 24.68 Example connections using simple SPI mode in single master mode with SPMR.SSE bit = 0

24.8.1 States of Pins in Master and Slave Modes

The direction (input or output) of pins for the simple SPI mode interface differs according to whether the device is a master (SCR.CKE[1:0] = 00b or 01b and SPMR.MSS = 0) or slave (SCR.CKE[1:0] = 10b or 11b and SPMR.MSS = 1).

Table 24.32 lists the relationship between the pin states, mode, and level on the SSn pin.

Table 24.32 States of pins by mode and input level on SSn pin

Mode	Input on SSn pin	State of MOSIn pin	State of MISOn pin	State of SCKn pin
Master mode*1	High level (transfer can proceed)	Output for data transmission*2	Input for received data	Clock output*3
	Low level (transfer cannot proceed)	High-impedance	Input for received data (but disabled)	High-impedance
Slave mode	High level (transfer cannot proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer can proceed)	Input for received data	Output for data transmission	Clock input

Note 1. When there is only a single master (SPMR.SSE = 0), transfer is possible regardless of the input level on the SSn pin. This is equivalent to input of a high level on the SSn pin.
Note 2. The MOSIn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE bit = 0).
Note 3. The SCKn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE = 0 and SCR.RE = 0) in a multi-master configuration (SPMR.SSE = 1).

24.8.2 SS Function in Master Mode

Setting the CKE[1:0] bits in the SCR to 00b or 01b and the MSS bit in the SPMR to 0 selects master mode operation. The SSn pin is not used in single-master configurations (SPMR.SSE = 0), so transmission or reception can proceed regardless of the value of the SSn pin.

When the level on the SSn pin is high in a multi-master configuration (SPMR.SSE = 1), a master device outputs clock signals from the SCKn pin before starting transmission or reception to indicate that there are no other masters or another master is performing reception or transmission.

When the level on the SSn pin is low in a multi-master configuration (SPMR.SSE = 1), there are other masters, and a transmission or reception is in progress. The MOSIn output and SCKn pins are placed in the high-impedance state and starting transmission or reception is not possible. In addition, the value of the SPMR.MFF bit is 1, indicating a mode fault error. In a multi-master configuration, start error processing by reading SPMR.MFF flag. If a mode fault error occurs while transmission or reception is in progress, transmission or reception does not stop, but the MOSIn and SCKn outputs are in the high-impedance state after completion of the transfer.

Use a general port pin to produce the SS output signal from the master.

24.8.3 SS Function in Slave Mode

Setting the SCR.CKE[1:0] bits to 10b or 11b and the SPMR.MSS bit to 1 selects slave operation. When the SSn pin is high, the MISOn output pin is in the high-impedance state and clock input through the SCKn pin is ignored. When the SSn pin is low, clock input through the SCKn pin is valid and transmission or reception can proceed.

If the input on the SSn pin changes from low to high during transmission or reception, the MISOn output pin is placed in the high-impedance state. Meanwhile, the internal processing for transmission or reception continues at the rate of the clock input through the SCKn pin until processing for the character being transmitted or received is complete, after which it stops, and the appropriate interrupt (SCIn_TXI, SCIn_RXI, or SCIn_TEI) is generated.

24.8.4 Relationship between Clock and Transmit/Receive Data

The CKPOL and CKPH bits in the SPMR register can be used to set up the clock for use in transmission and reception in four different ways. The relation between the clock signal and the transmission and reception of data is shown in Figure 24.69. The relation is the same for both master and slave operation. This is the same as when the level on the SSn pin is high.

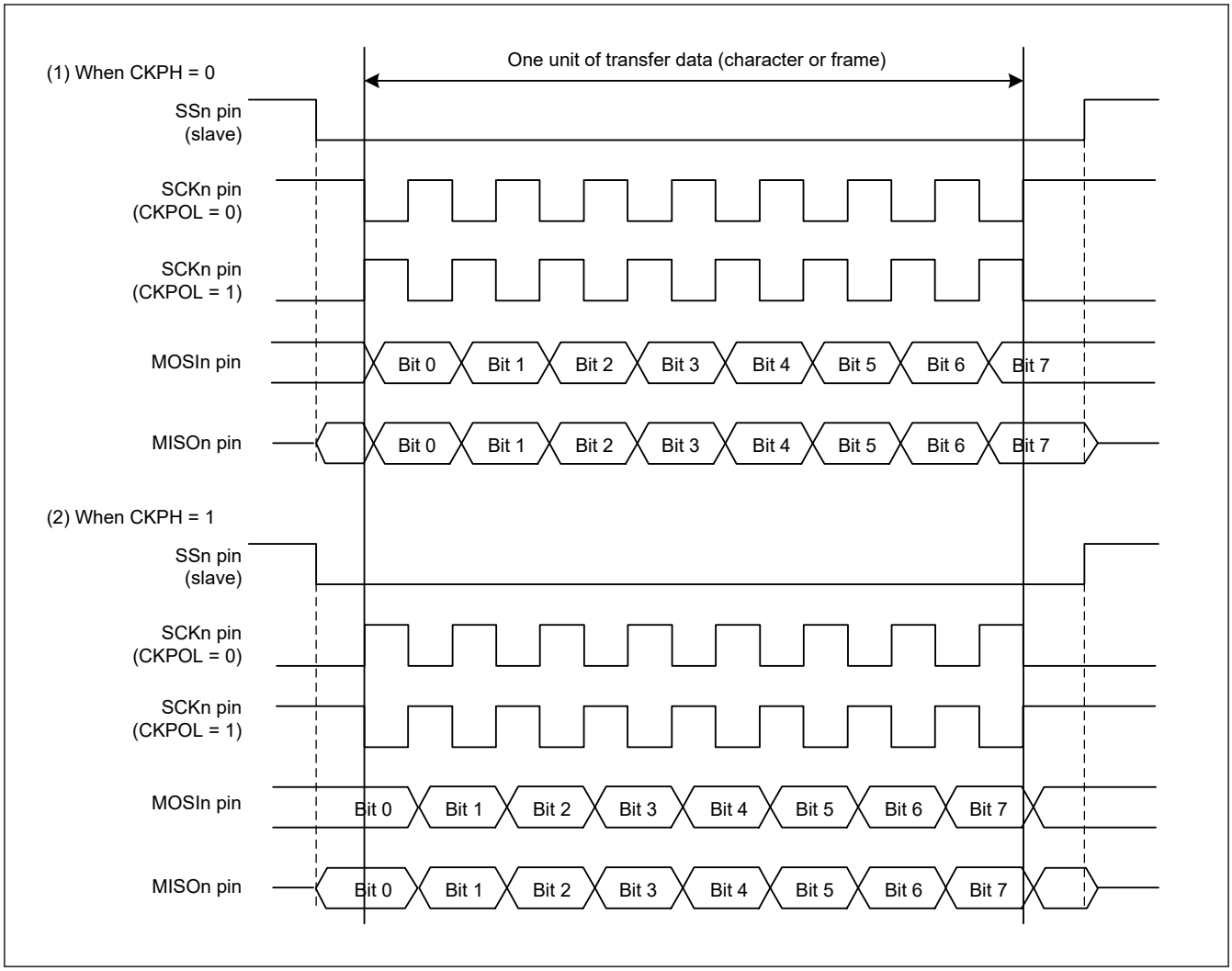


Figure 24.69 Relation between clock signal and transmit or receive data in simple SPI mode

24.8.5 SCI Initialization in Simple SPI Mode

Initialization in simple SPI mode is the same as in clock synchronous mode. See section 24.5.3. SCI Initialization in Clock Synchronous Mode for an example initialization flow. The CKPOL and CKPH bits in the SPMR register must be set to ensure that the clock signal is suitable for both master and slave devices.

Always initialize the SCR register before making any changes to the operating mode or transfer format.

Note: Only the RE bit is set to 0. The SSR.ORER, FER, PER, and RDR flags are not initialized.

Changing the value of the TE bit from 1 to 0 or from 0 to 1 when the TIE bit in the SCR register is 1 at the same time, leads to the generation of a transmit data empty interrupt (SCIn_TXI).

24.8.6 Transmission and Reception of Serial Data in Simple SPI Mode

In master operation, ensure that the SSn pin of the slave device on the other side of the transfer is at the low level before starting the transfer and at the high level on completion of the transfer. Otherwise, the procedures are the same as in clock synchronous mode.

24.9 Bit Rate Modulation Function

Using the bit rate modulation function, the bit rate can be evenly corrected using the number specified in the MDDR register among 256 clock cycles of internal clocks which is selected by the CKS[1:0] bits in SMR/SMR_SMCI.

Figure 24.70 shows an example where the PCLK is selected in the CKS[1:0] bits in SMR/SMR_SMCI, the BRR bit is set to 0, and the MDDR is set to 160 in asynchronous mode. In this example, the cycle of the base clock is evenly corrected (256/160) and the bit rate is also corrected (160/256).

Note: Enabling an internal clock causes bias, and expansion and contraction are generated in the pulse width of the internal base clock.

Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).

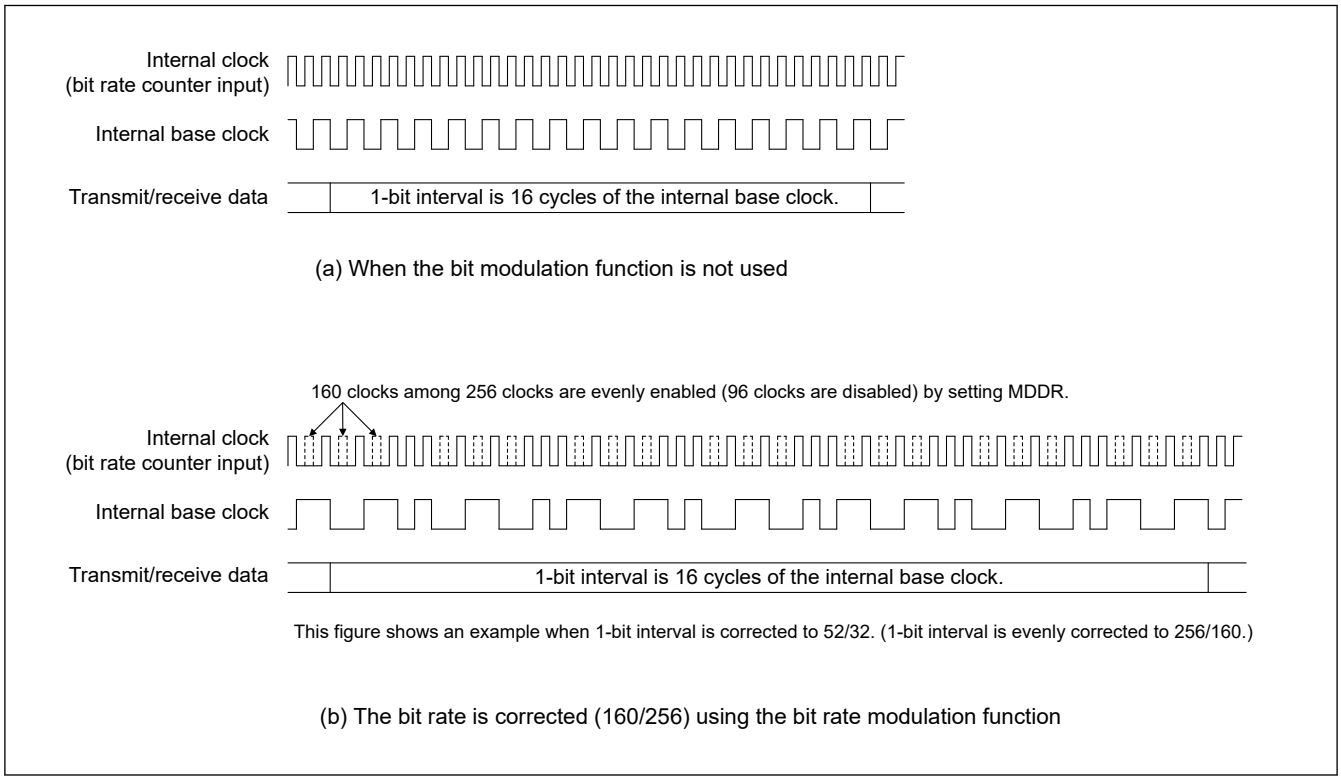


Figure 24.70 Example internal base clock when bit rate modulation function is used

24.10 Interrupt Sources

24.10.1 Buffer Operation for SCIn_TXI and SCIn_RXI Interrupts (Non-FIFO Selected)

If the conditions for an SCIn_TXI and SCIn_RXI interrupt are satisfied while the interrupt status flag in the ICU is 1, the ICU does not output the interrupt request but saves it internally with a capacity for retention of one request per source.

When the interrupt status flag in the ICU is set to 0, the interrupt request retained within the ICU is output. The internally retained interrupt request is automatically discarded when the actual interrupt is output. Clearing of the associated interrupt enable bit (the TIE or RIE bit in the SCR/SCR_SMCI) can also be used to discard an internally retained interrupt request.

24.10.2 Buffer Operation for SCIn_TXI and SCIn_RXI Interrupts (FIFO Selected)

When an interrupt status flag in the ICU is set to 1, the SCIn_TXI and SCIn_RXI interrupts do not output interrupt requests to the ICU. When an interrupt status flag of the ICU is set to 0, and if the conditions for an SCIn_TXI and SCIn_RXI interrupts are satisfied, an interrupt request is generated.

24.10.3 Interrupts in Asynchronous, Clock Synchronous, and Simple SPI Modes

(1) Non-FIFO selected

Table 24.33 lists interrupt sources in asynchronous mode, clock synchronous mode, and simple SPI mode.

A different interrupt vector can be assigned to each interrupt source. Individual interrupt sources can be enabled or disabled with the enable bits in the SCR register.

If the SCR.TIE bit is 1, an SCIn_TXI interrupt request is generated when transmit data is transferred from the TDR or TDRHL register[*1] to the TSR register. An SCIn_TXI interrupt request can also be generated by using a single instruction to set the SCR.TE and SCR.TIE bits to 1 at the same time. An SCIn_TXI interrupt request can activate the DTC to handle data transfer.

An SCIn_TXI interrupt request is not generated by setting the SCR.TE bit to 1 when SCR.TIE is 0 or by setting the SCR.TIE bit to 1 when the SCR.TE is 1.[*2]

When new data is not written by the time of transmission of the last bit of the current transmit data and SCR.TEIE is 1, the SSR.TEND flag is set to 1 and an SCIn_TEI interrupt request is generated. Additionally, when SCR.TE is 1, the SSR.TEND flag retains the value 1 until more transmit data is written to the TDR or TDRHL register[*1], and setting SCR.TEIE to 1 leads to the generation of an SCIn_TEI interrupt request.

Writing data to the TDR or TDRHL register[*1] leads to clearing of the SSR.TEND flag and, after a certain time, discarding of the SCIn_TEI interrupt request.

If the SCR.RIE bit is 1, an SCIn_RXI interrupt request is generated when received data is stored in the RDR register. An SCIn_RXI interrupt request can activate the DTC to handle data transfer.

Setting any of the SSR.ORER, FER, PER flags to 1 when the SCR.RIE bit is 1 leads to the generation of an SCIn_ERI interrupt request.

An SCIn_RXI interrupt request is not generated in this case. Clearing all these flags (ORER, FER, PER) leads to discarding of the SCIn_ERI interrupt request.

Note 1. When asynchronous mode and 9-bit data length are selected.
Note 2. To temporarily prohibit SCIn_TXI interrupts on transmission of the last of the data when a new round of transmission is to be started, after handling the transmission-completed interrupt, control activation of the interrupt by using the interrupt request enable bit in the ICU rather than using the SCR.TIE bit. This approach can prevent the suppression of SCIn_TXI interrupt requests in the transfer of new data.

(2) FIFO selected

Table 24.34 lists interrupt sources in FIFO selected mode.

If the SCR.TIE bit is 1, an SCIn_TXI interrupt request is generated when the stored amount of data in the FTDRL register becomes the threshold value indicated in FCR.TTRG or below. An SCIn_TXI interrupt request can also be generated by using a single instruction to set the SCR.TIE and SCR.TE bits to 1 simultaneously or by setting SCR.TIE to 1 when SCR.TE is 1.

An SCIn_TXI interrupt request is not generated by setting SCR.TE to 1 when SCR.TIE is 0.

If SCR.TEIE is 1 and if the next data is not written to the FTDRL register by the time the last bit of the transmit data is sent, the SSR_FIFO.TEND flag is set to 1 and the SCIn_TEI interrupt request is generated.

If SCR.RIE is 1, the SCIn_RXI interrupt request is generated when the stored amount of data in the FRDRL register is equal to or greater than the threshold value indicated in FCR.RTRG. When RTRG is 0, an SCIn_RXI interrupt does not occur even when the amount of data in the receive FIFO is equal to 0.

If the SCR.RIE bit is 1, when the SSR_FIFO.ORER flag is set to 1 or data with a framing error or a parity error is stored in the FRDRL register, the SCIn_ERI interrupt request is generated. When the amount of data stored in the FRDRL register is at the threshold value or above, the SCIn_RXI interrupt request is also generated. The SCIn_ERI interrupt request can be canceled, in which case SSR_FIFO.ORER, FER, and PER flags are all cleared.

Table 24.33 SCI interrupt sources with non-FIFO selected (1 of 2)

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation
SCIn_ERI (n = 0 to 2, 9)	Receive error[*1]	SSR.ORER, SSR.FER, SSR.PER, DCCR.DFER, DCCR.DPER	SCR.RIE	Not possible
SCIn_RXI (n = 0 to 2, 9)	Receive data full	SSR.RDRF	SCR.RIE	Possible
	Address match	DCCR.DCMF	SCR.RIE	Possible
SCIn_AM (n = 0 to 2, 9)	Address match	DCCR.DCMF	—	Not possible
SCIn_TXI (n = 0 to 2, 9)	Transmit data empty	SSR.TDRE	SCR.TIE	Possible

Table 24.33 SCI interrupt sources with non-FIFO selected (2 of 2)

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation
SCIn_TEI (n = 0 to 2, 9)	Transmit end	SSR.TEND	SCR.TEIE	Not possible

Note 1. The interrupt flag is only ORER when in clock synchronous and simple SPI mode.

Table 24.34 SCI interrupt sources with FIFO selected

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation
SCIn_ERI (n = 0 to 2, 9)	Receive error*1	SSR_FIFO.ORER, SSR_FIFO.FER, SSR_FIFO.PER, DCCR.DFER, DCCR.DPER	SCR.RIE	Not possible
		SSR_FIFO.DR (when FCR.DRES = 1)	SCR.RIE	Not possible
SCIn_RXI (n = 0 to 2, 9)	Receive data full	SSR_FIFO.RDF	SCR.RIE	Possible
	Receive data ready	SSR_FIFO.DR (when FCR.DRES = 0)	SCR.RIE	Possible
	Address match	DCCR.DCMF	SCR.RIE	Possible
SCIn_AM (n = 0 to 2, 9)	Address match	DCCR.DCMF	—	Not possible
SCIn_TXI (n = 0 to 2, 9)	Transmit data empty	SSR_FIFO.TDFE	SCR.TIE	Possible
SCIn_TEI (n = 0 to 2, 9)	Transmit end	SSR_FIFO.TEND	SCR.TEIE	Not possible

Note 1. The interrupt flag is only ORER when in clock synchronous and simple SPI mode.

24.10.4 Interrupts in Smart Card Interface Mode

Table 24.35 lists interrupt sources in smart card interface mode. A transmit end interrupt (SCIn_TEI) request and an address match (SCIn_AM) request cannot be used in this mode.

Table 24.35 SCI Interrupt sources

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation
SCIn_ERI (n = 0 to 2, 9)	Receive error or error signal detection	SSR_SMCI.ORER, SSR_SMCI.PER, SSR_SMCI.ERS	SCR_SMCI.RIE	Not possible
SCIn_RXI (n = 0 to 2, 9)	Receive data full	SSR_SMCI.RDRF	SCR_SMCI.RIE	Possible
SCIn_TXI (n = 0 to 2, 9)	Transmit data empty	SSR_SMCI.TEND	SCR_SMCI.TIE	Possible

Data transmission or reception using the DTC is also possible in smart card interface mode, similar to normal SCI mode. In transmission, when the SSR_SMCI.TEND flag is set to 1, an SCIn_TXI interrupt request is generated. This SCIn_TXI interrupt request activates the DTC, allowing transfer of transmit data if the SCIn_TXI request is previously specified as a source of DTC activation. The TEND flag is automatically set to 0 when the DTC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept at 0 and the DTC is not activated. Therefore, the SCI and DTC automatically transmit the specified number of bytes, including retransmission after an error occurrence. However, the SSR_SMCI.ERS flag is not automatically set to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the SCR_SMCI.RIE bit to 1 to enable an SCIn_ERI interrupt request to be generated at error occurrence.

When transmitting or receiving data using the DTC, always enable the DTC before making the SCI settings. For DTC settings, see section 15, Data Transfer Controller (DTC).

In reception, an SCIn_RXI interrupt request is generated when receive data is set to the RDR register. This SCIn_RXI interrupt request activates the DTC, allowing transfer of the receive data if the SCIn_RXI request is previously specified as a source of DTC activation. If an error occurs, the error flag is set. Therefore, the DTC is not activated and an SCIn_ERI interrupt request is issued to the CPU instead. The error flag must be cleared.

24.10.5 Interrupts in Simple IIC Mode

Table 24.36 lists the interrupt sources in simple IIC mode. The STI interrupt is allocated to the transmit end interrupt (SCIn_TEI) request. The receive error interrupt (SCIn_ERI) and the address match (SCIn_AM) request cannot be used.

The DTC can also be used to handle transfer in simple IIC mode.

When the SIMR2.IICINTM bit is 1:

- An SCIn_RXI request is generated on the falling edge of the SCLn signal for the 8th bit. If SCIn_RXI is previously set up as an activation source for the DTC, the SCIn_RXI request activates the DTC to handle transfer of the received data.
- An SCIn_TXI request is generated on the falling edge of the SCLn signal for the 9th bit (acknowledge bit). If SCIn_TXI is previously set up as an activation source for the DTC, the SCIn_TXI request activates the DTC to handle transfer of the transmit data.

When the SIMR2.IICINTM bit is 0:

- An SCIn_RXI request (ACK detection) is generated if the input on the SDAn pin is low on the rising edge of the SCLn signal for the 9th bit (acknowledge bit)
- An SCIn_TXI request (NACK detection) is generated if the input on the SDAn pin is high on the rising edge of the SCLn signal for the 9th bit (acknowledge bit)
- If SCIn_RXI is previously set up as an activation source for the DTC, the SCIn_RXI request activates the DTC to handle transfer of the received data.

If the DTC is used for data transfer in reception or transmission, always set up and enable the DTC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in SIMR3 are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 24.36 SCI interrupt sources

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation
SCIn_RXI (n = 0 to 2, 9)	Reception, ACK detection	—	SCMR.RIE	Possible*1
SCIn_TXI (n = 0 to 2, 9)	Transmission, NACK detection	—	SCMR.TIE	Possible
SCIn_TEI(STIn) (n = 0 to 2, 9)	Completion of generation of a start, restart, or stop condition	SIMR3.IICSTIF	SCMR.TEIE	Not possible

Note 1. Activation of the DTC is only possible when the SIMR2.IICINTM bit is 1 (use reception and transmission interrupts)

24.11 Event Linking

By using interrupt request signals as event signals, the SCIn can provide linked operation through the ELC for modules selected in advance.

Event signals can be output regardless of the values of the associated interrupt request enable bits.

(1) Error event output (receive error or error signal detected) (SCIn_ERI, n = 0 to 2, 9)

- Indicates abnormal termination because of a parity error during reception in asynchronous mode
- Indicates abnormal termination because of a framing error during reception in asynchronous mode
- Indicates abnormal termination because of an overrun error during reception
- Indicates detection of the error signal during transmission in smart card interface mode
- The SSR_FIFO.FER and PER flags are 0, and receive data less than the receive FIFO data trigger number is set in a reception FIFO buffer, and it indicates that 15 etus elapse when FIFO is selected and the FCR.DRES bit is 1

(2) Receive data full event output (SCIn_RXI, n = 0 to 2, 9)

- Indicates that ACK is detected if the SIMR2.IICINTM bit is 0 in simple IIC mode

- Indicates that the 8th-bit SCLn falling edge is detected if the SIMR2.IICINTM bit is 1 in simple IIC mode
- When the SIMR2.IICINTM bit is 1 during master transmission in simple IIC mode, set the ELC so that receive data full events are not used

Non-FIFO selected

- Indicates that received data is set in the Receive Data Register (RDR or RDRHL).

FIFO selected

- Using this event output is prohibited.

(3) Transmit data empty event output (SCIn_TXI, n = 0 to 2, 9)

- Indicates that the SCR/SCR_SMCI.TE bit is changed from 0 to 1
- Indicates that transmission is complete in smart card interface mode
- Indicates that NACK is detected if the SIMR2.IICINTM bit is 0 in simple IIC mode
- Indicates that the 9th-bit SCLn falling edge is detected if the SIMR2.IICINTM bit is 1 in simple IIC mode

Non-FIFO selected

- Indicates that transmit data is transferred from the Transmit Data Register (TDR or TDRHL) to the Transmit Shift Register (TSR).

FIFO selected

- Using this event output is prohibited.

(4) Transmit end event output (SCIn_TEI, n = 0 to 2, 9)

- Indicates the completion of transmission
- Indicates that the starting condition, resumption condition, or termination condition is generated in simple IIC mode

Note: When FIFO is selected, using this event output is prohibited

(5) Address match event output (SCIn_AM, n = 0 to 2, 9)

- Indicates a match of the comparison data (CDR.CMPD) with one frame of receive data when DCCR.DCME is set to 1 in asynchronous mode, including multi-processor mode.

24.12 Address Non-match Event Output (SCI0_DCUF)

SCI0_DCUF indicates the non-match of comparison data (CDR.CMPD) with receive data that is one frame of the data that is received when DCCR.DCME is set to 1 in asynchronous mode, including multi-processor mode. This event can be used for Snooze end request only. For details, see section 10, Low Power Modes.

24.13 Noise Cancellation Function

Figure 24.71 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of a 2-stage flip-flop circuit and a match detection circuit. When the input signals of the noise filter and the output signals of the 2-stage flip-flop circuits completely match, the matched level is conveyed as an internal signal. Unless otherwise matched, the previous value is retained. When the same level is retained for 3 cycles or longer on the sampling clock of the noise filter, it is considered as a valid receive signal. A change in pulse for 3 cycles or shorter is considered as noise, not as a receive signal.

In asynchronous mode, the noise cancellation function can be applied to the receive signal input to the RXDn pin. The receive level of the RXDn is taken in the flip-flop circuit of the noise filter on the base clock of asynchronous mode.

- When SEMR.ABCS = 0 and SEMR.ABCSE = 0, the cycle is 1/16 of a 1-bit period.
- When SEMR.ABCS = 1 and SEMR.ABCSE = 0, the cycle is 1/8 of a 1-bit period.
- When SEMR.ABCSE = 1, the cycle is 1/6 of a 1-bit period.

In simple IIC mode, this function can be used for each input on SDAn and SCLn. The sampling clock is selected from divided clock of baud rate generator settings SNFR.NFCS[2:0].

If the base clock is stopped once with the noise filter enabled and then the base clock input is restarted again, the noise filter operation resumes from the state where the clock was stopped. When SCR.TE and SCR.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, the function determines that a level match is detected and the result is conveyed as an internal signal. When the level being input corresponds to 0, the initial output of the noise filter is retained until the level matches in three consecutive sampling cycles.

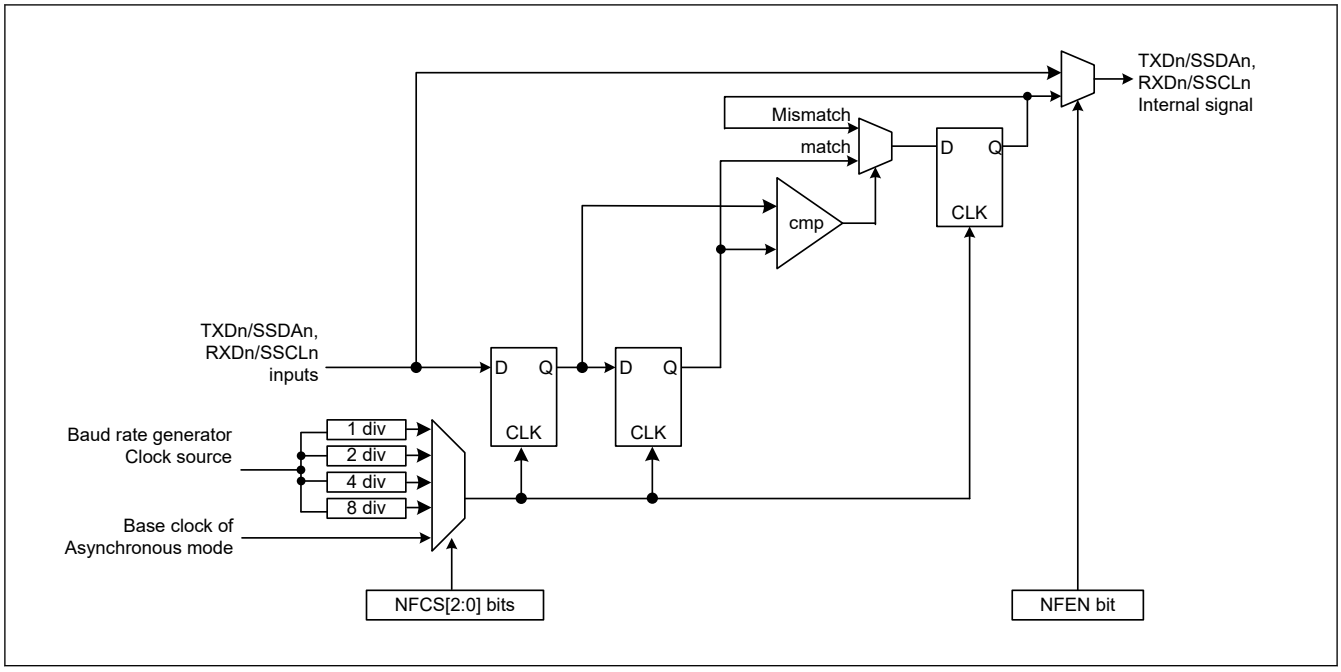


Figure 24.71 Digital noise filter circuit block diagram

24.14 Usage Notes

24.14.1 Settings for the Module-Stop Function

The Module Stop Control Register B (MSTPCRB) can enable or disable SCI operation. The SCI is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

24.14.2 SCI Operation during Low Power State

(1) Transmission

When setting the module to the stopped state or in transitions to Software Standby, stop operations (by setting the TIE, TE, and TEIE bits in the SCR/SCR_SMCI to 0) after switching the TXDn pin to the general I/O port pin function. When setting I/O port as an SCI connection, the SPTR register can control the state of the TXDn pin. Setting the TE bit to 0 initializes the TSR register and the TEND bit in the SSR/SSR_SMCI is initialized to 1 with non-FIFO selected, and the value is retained, with FIFO selected. Depending on the port settings and SPTR register settings, output pins might output the level before a transition to the low-power state is made after release from the module-stopped state or Software Standby mode. When transitions to these states are made during transmission, the transmitted data becomes indeterminate.

To transmit data in the same transmission mode after cancellation of the low-power state:

1. Set the TE bit to 1.
2. Read SSR/SSR_FIFO/SSR_SMCI.
3. Write data to TDR sequentially to start data transmission.

To transmit data with a different transmission mode, initialize the SCI first.

Figure 24.72 shows an example flow of transition to Software Standby mode during transmission. Figure 24.73 and Figure 24.74 show the port pin states during transition to Software Standby mode.

Before specifying the module-stop state or transitioning to Software Standby mode from the transmission mode using DTC transfer, stop the transmit operations (TE = 0). To start transmission after cancellation using the DTC, set the TE bit to 1. The SCIn_TXI interrupt flag is set to 1 and transmission starts using the DTC.

(2) Reception

When address match function is not used as wakeup condition

Before specifying the module-stop state or transitioning to Software Standby mode, stop the receive operations (RE = 0 in SCR/SCR_SMCI). If transition is made during data reception, the received data is invalid.

Figure 24.75 shows an example flow of transition to Software Standby mode during reception.

When address match function is used as wakeup condition

Before specifying the module-stop state or transitioning to Software Standby mode:

1. Set the operations after cancellation of the low power state.
2. Set CDR.CMPD and DCCR.DCME to 1.
3. Set the receive operations (RE = 1 in SCR/SCR_SMCI).
4. Set the module-stop state or Software Standby mode.

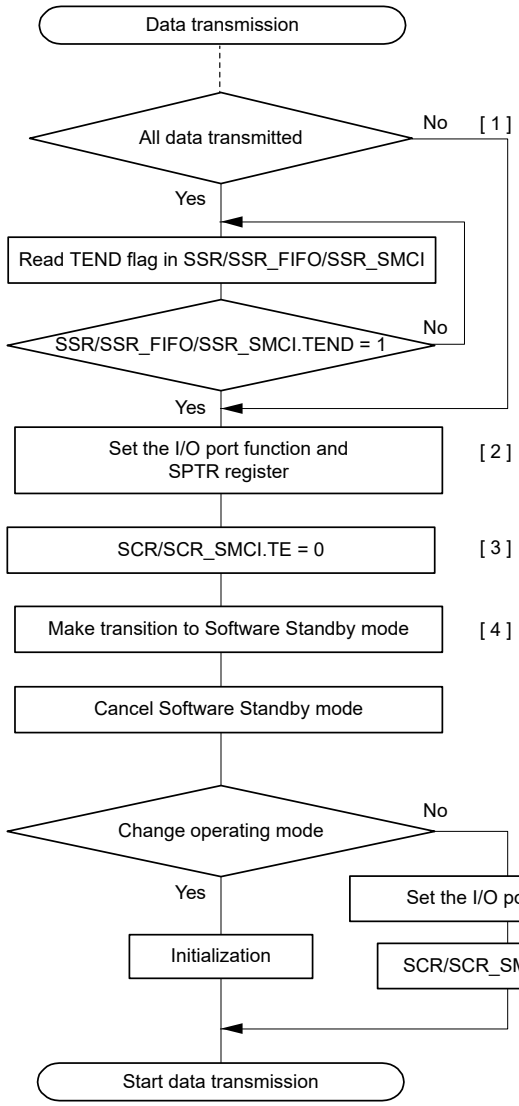
When SCI transfers to low power mode, if the receive data pin (RXD) is at the low level, set SEMR.RXDESEL = 0.

When setting SEMR.RXDESEL = 1, there is a possibility that a start bit (falling edge of RXD pin) cannot be detected on release of the low power mode.

Figure 24.76 shows an example flow of transition to Software Standby mode during reception with address match.

When using SCI0 in Snooze mode

When using SCI0 in Snooze mode, some restrictions apply, including maximum bit rates. For details, see section 10, Low Power Modes.



[1] Data being transmitted is lost. Data can be normally transmitted from the CPU by setting the TE bit in SCR/SCR_SMCI to 1, reading SSR/SSR_FIFO/ SSR_SMCI, and writing in Software Standby mode. However, if the DTC is activated, the data remaining in the DTC is transmitted when both the TE and TIE bits in SCR/SCR_SMCI are set to 1.

[2] Set the I/O port function and SPTR register settings to switch the TXDn pin to operate as a general I/O port.

[3] Set SCR/SCR_SMCI.TE bit to 0. If SCR/SCR_SMCI.TIE = 1 and SCR/SCR_SMCI.TEIE = 1, these are set to 0 simultaneously with the SCR.TE bit.

[4] This includes the setting for the module-stop state.

Figure 24.72 Example flow of transition to Software Standby mode during transmission

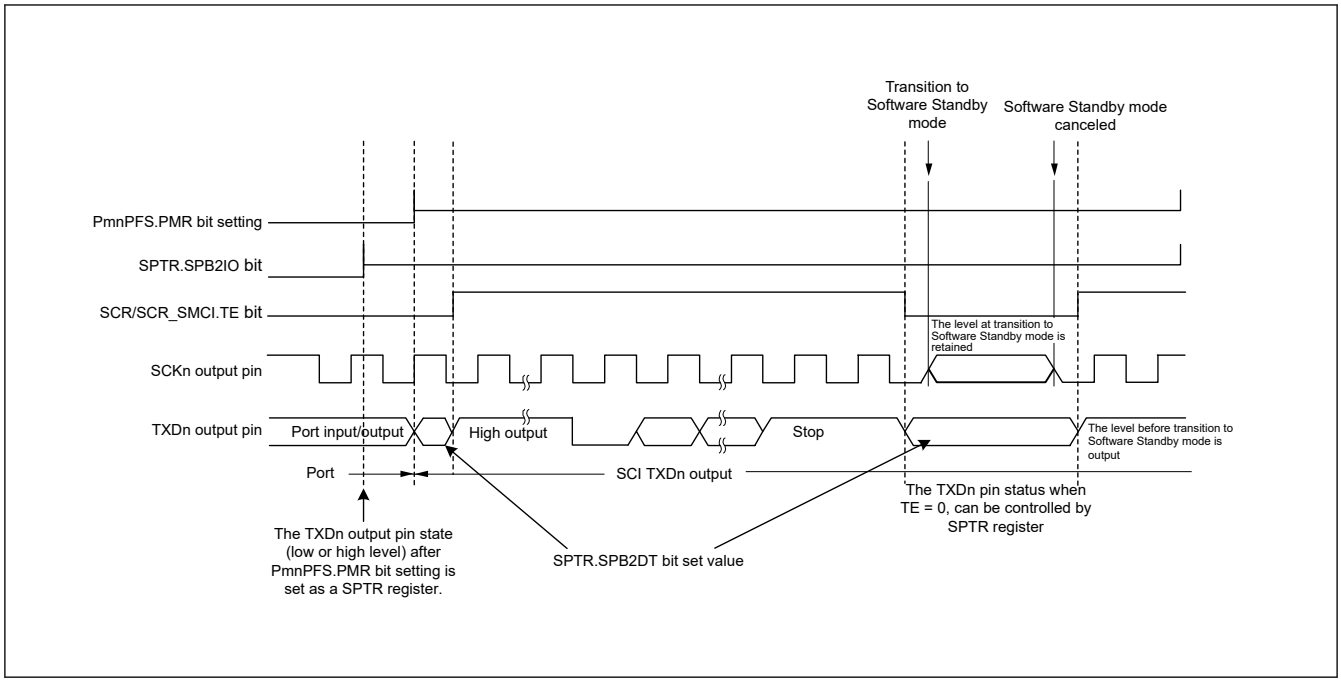


Figure 24.73 Port pin states during transition to Software Standby mode with internal clock and asynchronous transmission

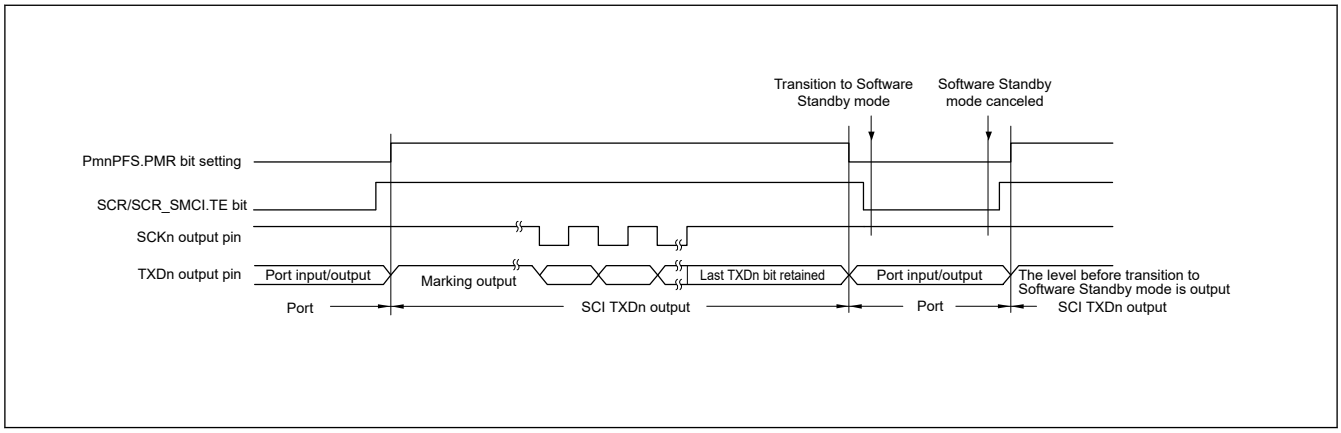


Figure 24.74 Port pin states during transition to Software Standby mode with internal clock and clock synchronous transmission

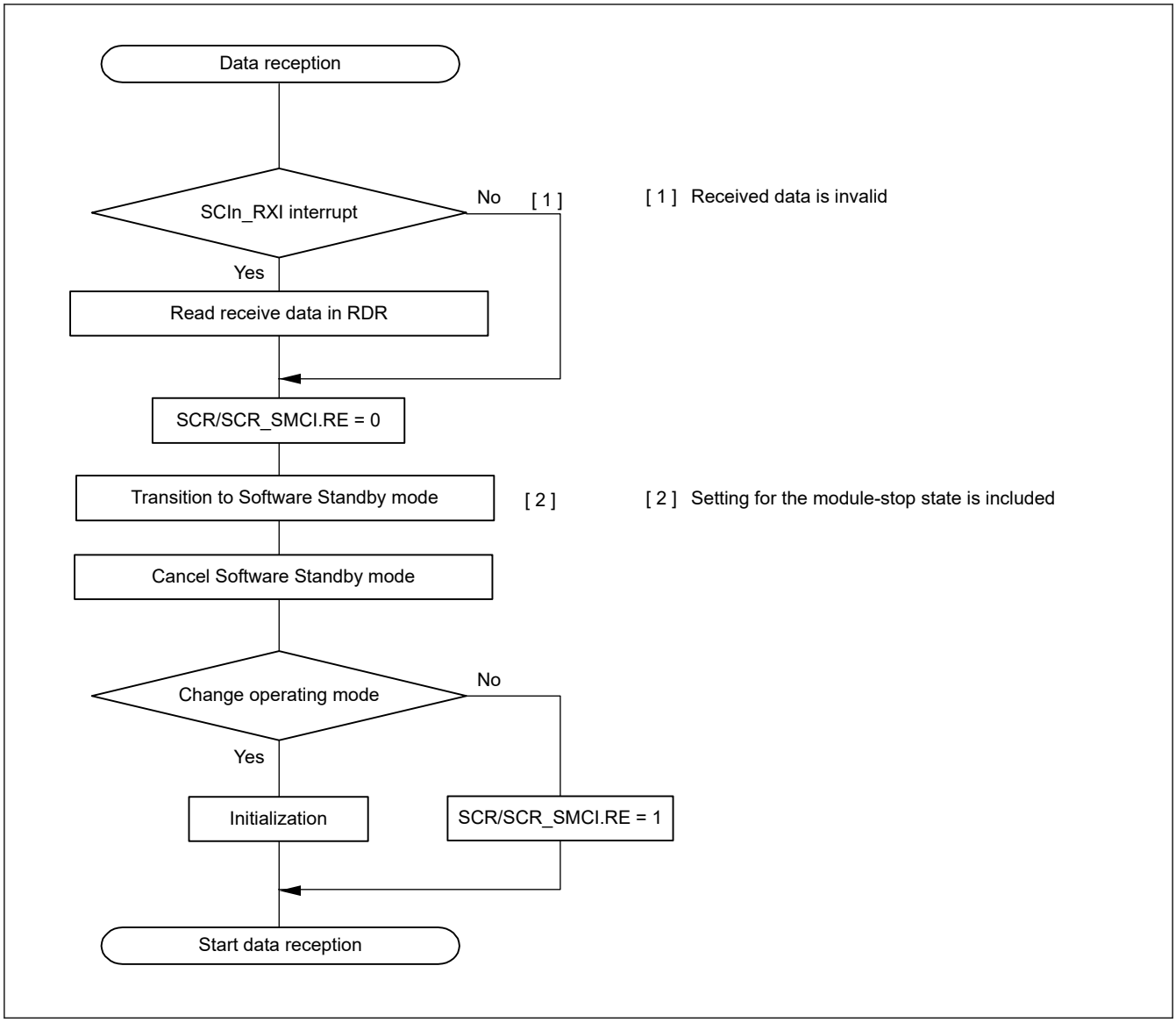


Figure 24.75 Example flow of transition to Software Standby mode during reception

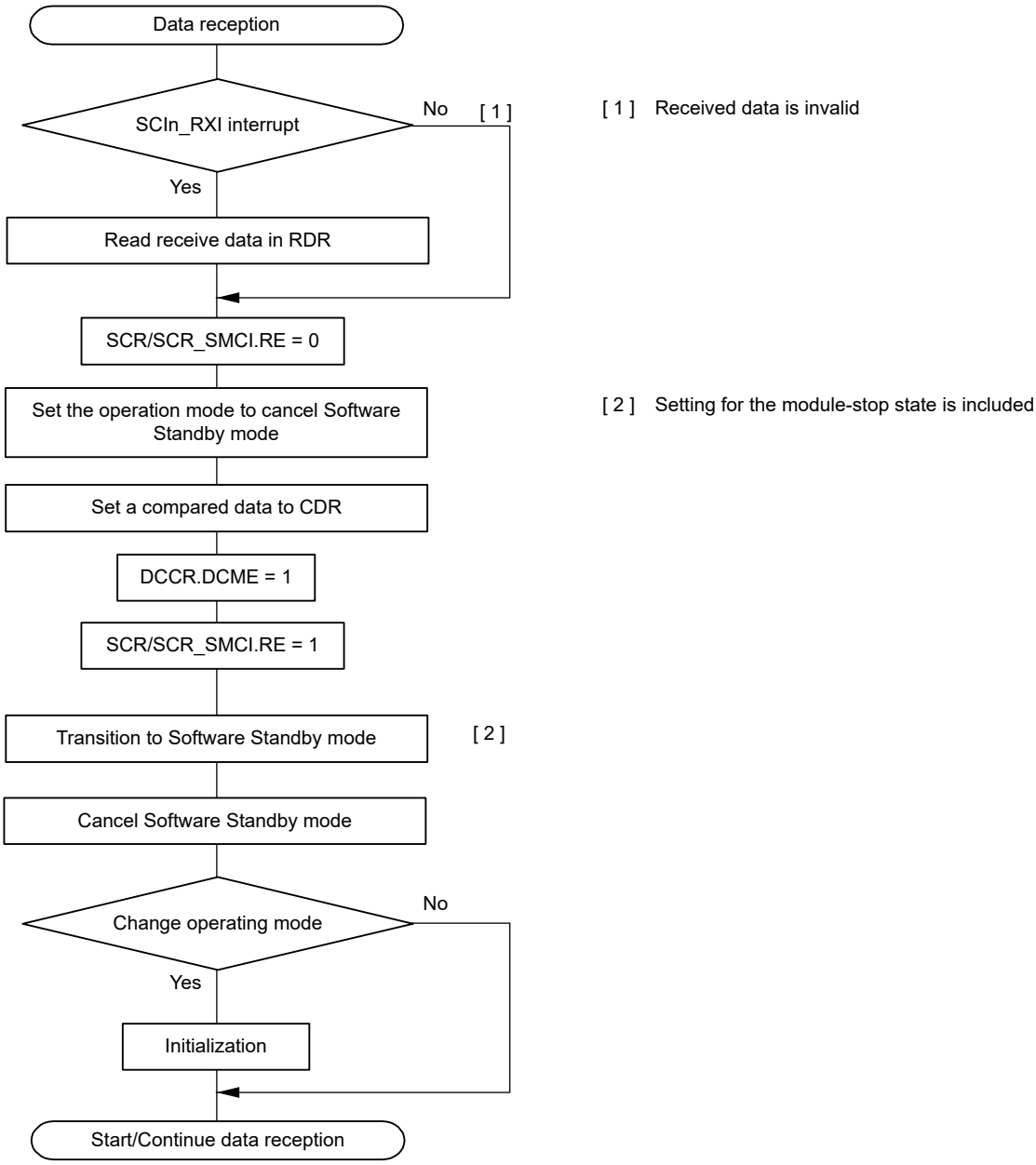


Figure 24.76 Example flow of transition to Software Standby mode during reception with address match

24.14.3 Break Detection and Processing

(1) Non-FIFO selected

When a framing error is detected, a break can be detected by reading the RXDn pin value directly. In a break, the input from the RXDn pin becomes all 0s, and the SSR.FER flag is set to 1 to indicate a framing error, and the SSR.PER flag might also be set to 1 to indicate a parity error. The SCI continues the receive operation even after a break is received. Therefore, even if the FER flag is 0, indicating that no framing error occurred, it is set to 1 again. When the SEMR.RXDESEL bit is 1, the SCI sets the SSR.FER flag to 1 and stops receiving operations until a start bit of the next data frame is detected. If the SSR.FER flag is set to 0, the SSR.FER flag retains 0 during the break.

When the RXDn pin is set to 1 and the break ends, detecting the beginning of the start bit on the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

(2) FIFO selected

After a framing error is detected and when the SCI detects that continuous receive data is 0 for 1 frame, reception stops. When a framing error is detected, a break can be detected by reading the SPTR.RXDMON flag value. After the RXDn signal is in high and the break is finished, data reception to the FRDRHL register resumes.

24.14.4 Mark State and Production of Breaks

When the SCR/SCR_SMCI.TE bit is 0, disabling serial transmission, the state of the TXDn pin can be set using the SPTR.SPB2IO and SPTR.SPB2DT bits. With this approach, a TXDn pin can be placed in the mark state to transmit a break.

Before setting the SCR/SCR_SMCI.TE bit to 1, enabling serial transmission, set the SPB2IO and SPB2DT bits to put the communication line in the mark state (the state of 1), and change the TXDn pin using I/O port function. To output a break on data transmission, after setting the TXDn pin to output 0 by setting the SPB2IO and SPB2DT bits, change the TXDn pin using the I/O port function and set the SCR/SCR_SMCI.TE bit to 0. When the SCR/SCR_SMCI.TE bit is set to 0, the transmitter is initialized regardless of the current state of transmission.

24.14.5 Receive Error Flags and Transmit Operation in Clock Synchronous Mode and Simple SPI Mode

Transmission cannot start when a receive error flag (ORER) in SSR/SSR_FIFO is set to 1, even when data is written to TDR or FTDRL[*1]. Always set the receive error flags to 0 before starting transmission.

Note: The receive error flags cannot be set to 0 when the RE bit in SCR/SCR_SMCI is set to 0 (serial reception is disabled).

Note 1. Do not use the FTDRH register in simple SPI mode.

24.14.6 Restrictions on Clock Synchronous Transmission in Clock Synchronous Mode and Simple SPI Mode

When the external clock source is used as a synchronization clock, the following restrictions apply.

(1) Start of transmission

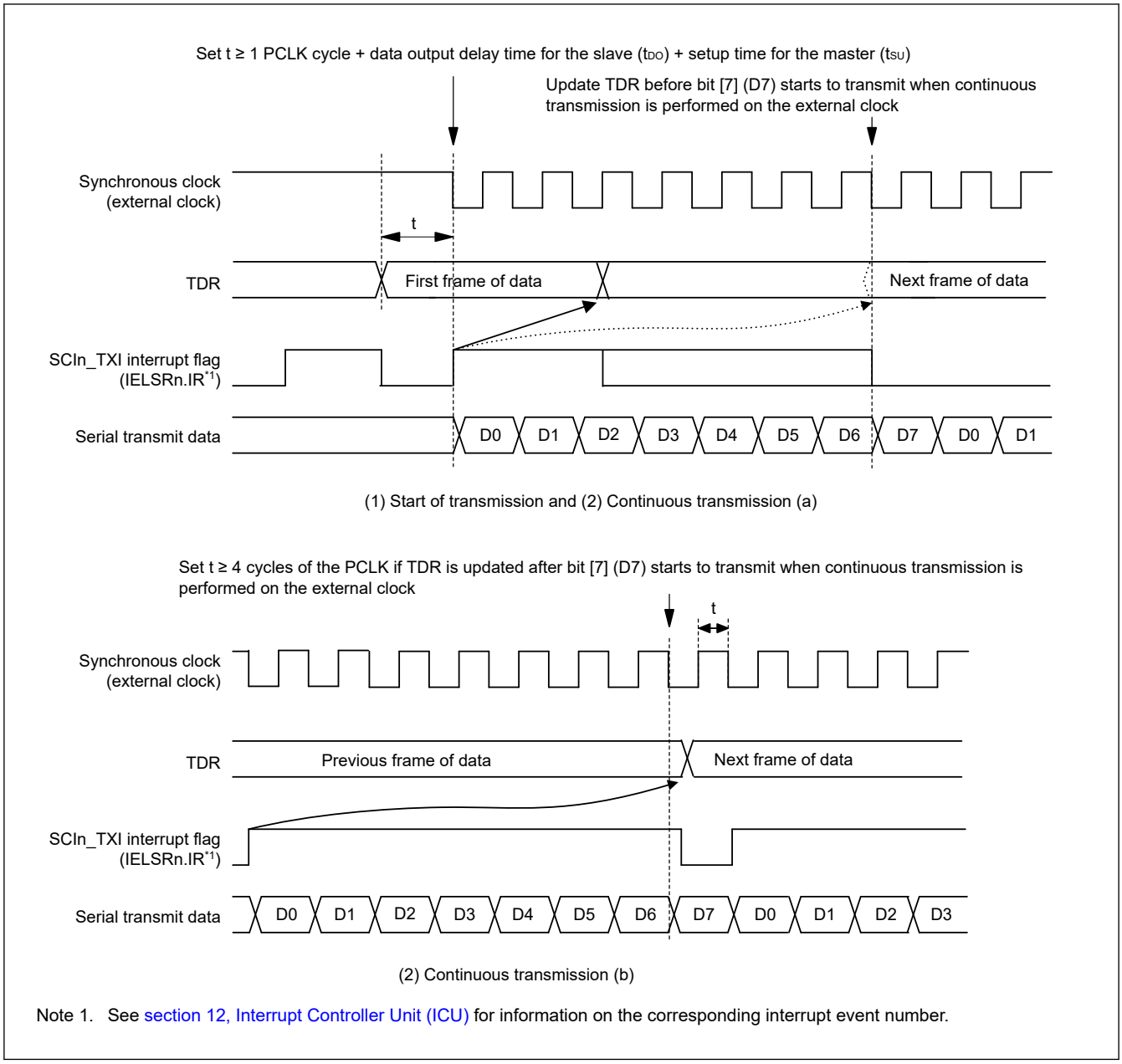
Wait at least the following time from writing transmit data to TDR to the start of the external clock input:

1 PCLK cycle + data output delay time for the slave (t_{DO}) + setup time for the master (t_{SU}). See Figure 24.77.

(2) Continuous transmission

Write the next transmit data to TDR or TDRHL before the falling edge of the transmit clock for bit [7] . See Figure 24.77.

When updating TDR after bit [7] has started to transmit, update TDR while the synchronization clock is in the low-level period, and set the high-level width of the transmit clock (bit [7]) to 4 PCLK cycles or longer. See Figure 24.77.



Note 1. See section 12, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 24.77 Restraints on use of external clock in clock synchronous transmission

24.14.7 Restrictions on Using DTC

During transmission or reception operations using the DTC, do not set transfer data for the DTC.

(1) Writing data to TDR (FTDRHL)

Non-FIFO selected

Data can be written to TDR and TDRHL. However, if new data is written to TDR or TDRHL when transmit data remains in TDR or TDRHL, the previous data in TDR and TDRHL is lost because it was not transferred to TSR yet. When using DTC, always write transmit data to TDR or TDRHL in the SCIn_TXI interrupt request handling routine.

FIFO selected

It is possible to write data to the FTDRH and FTDRL registers when SCR.TE is 1. Confirm the amount of writable data using the FDR.T[4:0] bits.

(2) Reading data from RDR (FRDRHL)

When using the DTC to read RDR and RDRHL, always set the receive data full interrupt (SCIn_RXI) as the activation source of the relevant SCI.

24.14.8 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IELSRn.IR flag) in the ICU is 1, follow the procedure in this section to clear interrupt requests before permitting operations (by setting the SCR/SCR_SMCI.TE or SCR/SCR_SMCI.RE bit to 1). For details on the interrupt status flag, see section 12, Interrupt Controller Unit (ICU).

1. Confirm that transfer has stopped (the SCR/SCR_SMCI.TE or SCR/SCR_SMCI.RE bit is 0)
2. Set the associated interrupt enable bit (SCR/SCR_SMCI.TIE or SCR/SCR_SMCI.RIE bit) to 0
3. Read the associated interrupt enable bit (SCR/SCR_SMCI.TIE or SCR/SCR_SMCI.RIE bit) to check that it actually becomes 0
4. Set the interrupt status flag, IELSRn.IR, in the ICU to 0

24.14.9 External Clock Input in Clock Synchronous Mode and Simple SPI Mode

In clock synchronous mode and simple SPI mode, the external clock SCKn must be input as follows:

High-pulse period, low-pulse period = 2 PCLK cycles or more, period = 6 PCLK cycles or more.

24.14.10 Limitations on Simple SPI Mode

(1) Master mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set in the SPMR.CKPH and CKPOL bits when the SPMR.SSE bit is 1.

  This prevents the clock line from being placed in the high-impedance state when the SCR.TE bit is set to 0 or unexpected edges from being generated on the clock line when the SCR.TE bit changes from 0 to 1. When the SPMR.SSE bit is 0 in single master mode, pulling up or pulling down the clock line is not required because the clock line is not placed in the high-impedance state even when the SCR.TE bit is set to 0.
- For the clock delay setting (SPMR.CKPH bit is 1), the receive data full interrupt (SCIn_RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 24.78. If the TE and RE bits in the SCR register become 0 before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Additionally, an SCIn_RXI interrupt might lead to the input signal on the SSn pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master configuration, the SCKn pin output goes to high-impedance while the input on the SSn pin is at the low level if a mode fault error occurs while a character is being transferred, stopping supply of the clock signal to the connected slave. Reset the connected slave to avoid misaligned bits when transfer is restarted.

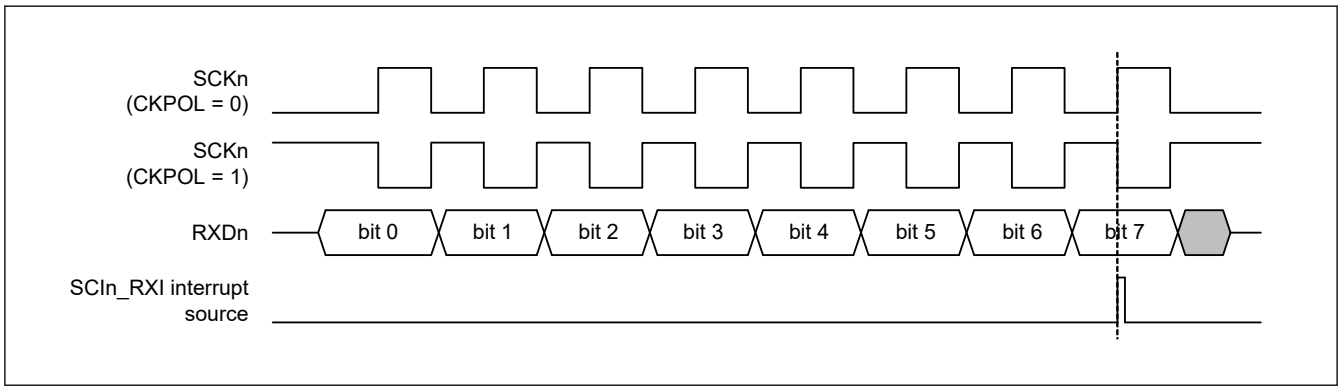


Figure 24.78 Timing of SCIn_RXI interrupt in simple SPI mode with clock delay

(2) Slave mode

- Wait at least the following time from writing transmit data in the TDR register to the start of the external clock input.

  1 PCLK cycle + data output delay for the slave (t_{DO}) + setup time for the master (t_{SU})

  Also wait at least 5 PCLK cycles from the input of the low level on the SSn pin to the start of the external clock input.
- Provide an external clock signal to the master the same as the data length for transfer
- Control the input on the SSn pin before the start and after the end of data transfer
- When the input level on the SSn pin is to be changed from low to high while a character is being transferred, set the TE and RE bits in the SCR register to 0 and, after restoring the settings, restart transfer of the first byte

24.14.11 Notes on Transmit Enable Bit (SCR.TE)

In initial register value, when SCR.TE = 0, the state of the TXDn pin is high impedance. The TXDn line should not be high impedance by the following one of ways.

1. The pull-up resistance is connected to the TXDn line.
2. Before setting the SCR.TE bit to 0, the function of the pin should be changed to a general-purpose output port. After that, set the SCR.TE bit to 1, and then change the function of the pin to TXDn.
3. In asynchronous mode, set SPTR and decided level of TXDn pin during SCR.TE = 0.

In the Simple SPI mode slave operation, the MISOn pin operates in the same way as the above TXDn pin. The MISOn pin, the same as TXDn pin, should not be high impedance by the above list number 1 or list number 2.

24.14.12 Note on Stopping Reception When Using the RTS Function in Asynchronous Mode

One clock cycle of PCLK is required for the time from setting the SCR.RE bit to 0 to stopping the RTS signal generator in asynchronous mode.

When reading the RDR (or RDRL) register after setting the SCR.RE bit to 0, confirm that the RE bit has been set to 0 before reading the RDR (or RDRL) register to prevent these two processes from being performed consecutively.

25. I²C Bus Interface (IIC)

25.1 Overview

The I²C bus interface (IIC) has 1 channel. The IIC module conforms with and provides a subset of the NXP I²C (Inter-Integrated Circuit) bus interface functions.

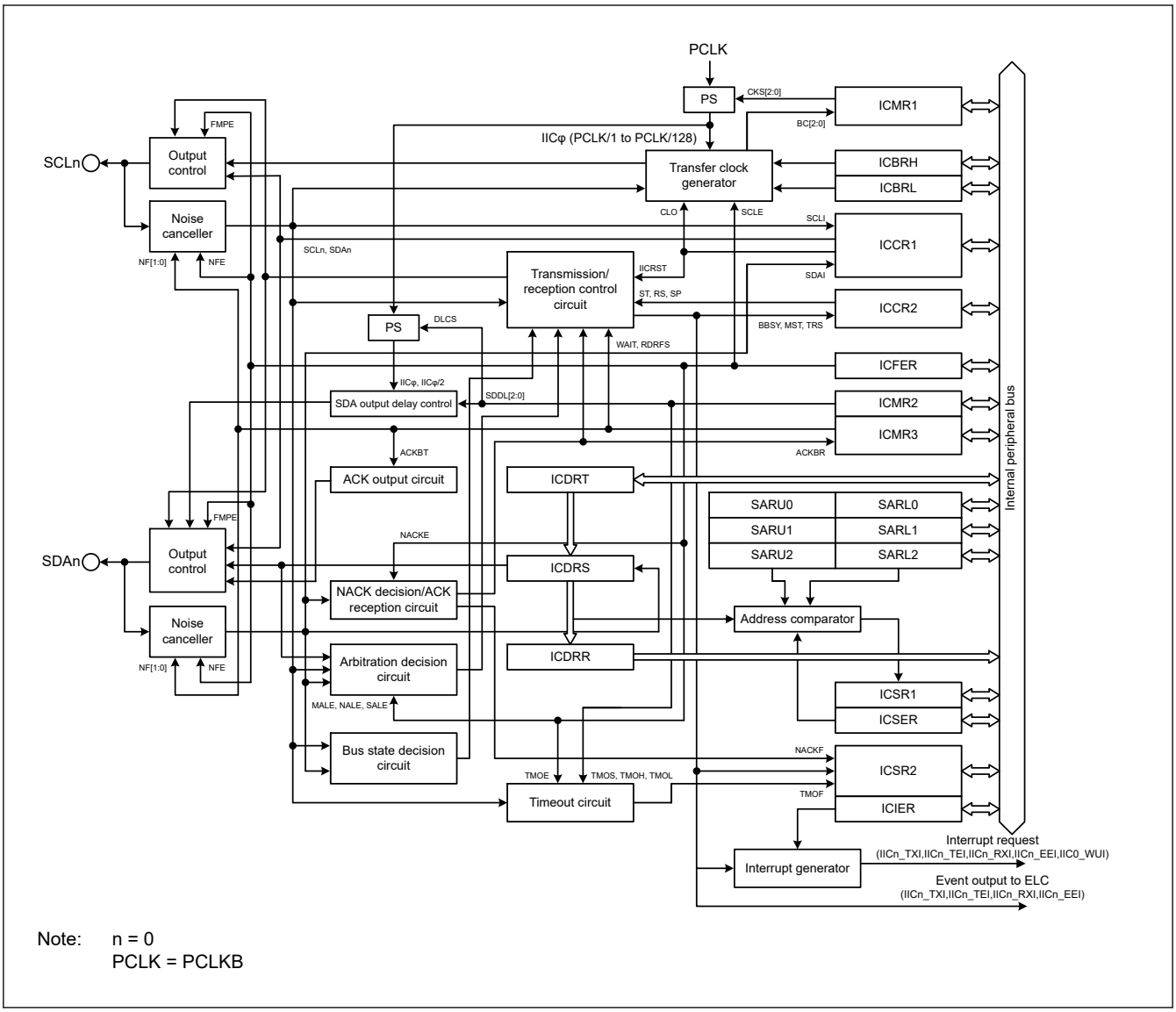
Table 25.1 lists the IIC specifications, Figure 25.1 shows a block diagram, and Figure 25.2 shows an example of I/O pin connections to external circuits, with an I²C bus configuration. Table 25.2 lists the I/O pins.

Table 25.1 IIC specifications (1 of 2)

Parameter	Specifications
Communications format	• I²C-bus format or SMBus format • Master or slave mode selectable • Automatic securing of the setup times, hold times, and bus-free times for the transfer rate
Transfer rate	• Fast-mode supported, up to 400 kbps
SCL clock	For master operation, the duty cycle of the SCL clock is selectable in the range from 4% to 96%
Issuing and detecting conditions	• Start, restart, and stop conditions are automatically generated • Start conditions (including restart conditions) and stop conditions are detectable
Slave address	• Configurable for up to three different slave addresses • 7- and 10-bit address formats supported, including simultaneous use • General call addresses, device ID addresses, and SMBus host addresses detectable
Acknowledgment	• For transmission, automatic loading of the acknowledge bit Transfer of the next transmit data can be automatically suspended on detection of a not-acknowledge bit. • For reception, automatic transmission of the acknowledge bit If a wait between the 8th and 9th clock cycles is selected, the software can control the value in the acknowledge field in response to the received value.
Wait function	During reception, the following wait periods are available by holding the SCL clock low: • Waiting between the eighth and ninth clock cycles • Waiting between the ninth clock cycle and the 1st clock cycle of the next transfer
SDA output delay function	Output timing of transmitted data, including the acknowledge bit, can be delayed
Arbitration	• For multi-master operation: – SCL clock synchronization is possible when conflict occurs with the SCL signal from another master – When issuing the start condition creates conflict on the bus, loss of arbitration is detected by testing for a mismatch between the internal signal for the SDA line and the level on the SDA line – In master operation, loss of arbitration is detected by testing for non-matching between the signal on the SDA line and the internal signal for the SDA line • Loss of arbitration because the start condition occurs while the bus is busy is detectable, to prevent the issuing of double start conditions • Loss of arbitration is detectable on transfer of a not-acknowledge bit because the internal signal for the SDA line and the level on the SDA line do not match • Loss of arbitration because a mismatch of internal and line levels for data is detectable in slave transmission
Timeout function	Internal detection of long-interval stops of the SCL clock
Noise cancellation	• Digital noise filters for both the SCL and SDA signals • Programmable window for noise cancellation by the filters
Interrupt sources	• Transfer error or event occurrence (arbitration-lost, NACK, timeout, start or restart condition, or stop condition) • Receive data full, including matching with a slave address • Transmit data empty, including matching with a slave address • Transmit end
Module-stop function	Module-stop state can be set
IIC operating modes	• Master transmit • Master receive • Slave transmit • Slave receive

Table 25.1 IIC specifications (2 of 2)

Parameter	Specifications
Event link function (output)	• Transfer error or event occurrence (arbitration-lost, NACK, timeout, start or restart condition, or stop condition) • Receive data full, including matching with a slave address • Transmit data empty, including matching with a slave address • Transmit end
Wakeup function	CPU can return from Software Standby mode using a wakeup event



Note: n = 0
PCLK = PCLKB

Figure 25.1 IIC block diagram

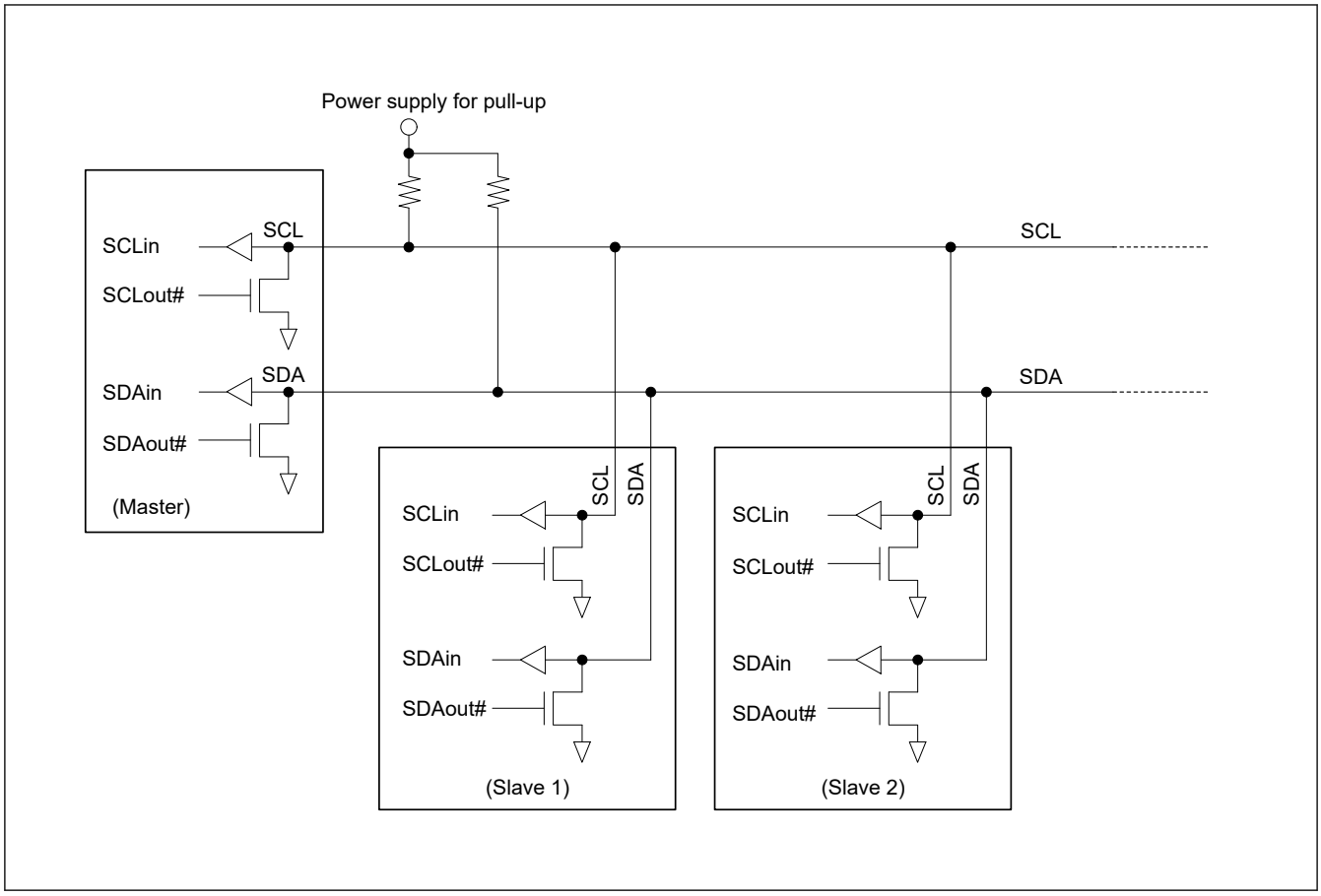


Figure 25.2 I/O pin connection to an external circuit (I^2C bus configuration example)

The input level of the signals for IIC is CMOS when I^2C bus is selected (ICMR3.SMBS = 0), or TTL when SMBus is selected (ICMR3.SMBS = 1).

Table 25.2 IIC I/O pins

Channel	Pin name	I/O	Function
IICn	SCLn	I/O	IICn serial clock I/O pin
	SDAn	I/O	IICn serial data I/O pin

Note: n = 0

25.2 Register Descriptions

25.2.1 ICCR1 : I^2C Bus Control Register 1

Base address: IIC0 = 0x4005_3000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Function	R/W
0	SDAI	SDA Line Monitor 0: SDA0 line is low 1: SDA0 line is high	R

Bit	Symbol	Function	R/W
1	SCLI	SCL Line Monitor 0: SCL0 line is low 1: SCL0 line is high	R
2	SDAO	SDA Output Control/Monitor 0: Read: IIC drives SDA0 pin low Write: IIC drives SDA0 pin low 1: Read: IIC releases SDA0 pin Write: IIC releases SDA0 pin	R/W
3	SCLO	SCL Output Control/Monitor Use an external pull-up resistor to drive the signal high. 0: Read: IIC drives SCL0 pin low Write: IIC drives SCL0 pin low 1: Read: IIC releases SCL0 pin Write: IIC releases SCL0 pin	R/W
4	SOWP	SCLO/SDAO Write Protect This bit is read as 1. 0: Write enable SCLO and SDAO bits 1: Write protect SCLO and SDAO bits	W
5	CLO	Extra SCL Clock Cycle Output This bit clears automatically after 1 clock cycle is output. 0: Do not output extra SCL clock cycle (default) 1: Output extra SCL clock cycle	R/W
6	IICRST	I2C Bus Interface Internal Reset This setting clears the bit counter and the SCL0/SDA0 output latch. 0: Release IIC reset or internal reset 1: Initiate IIC reset or internal reset	R/W
7	ICE	I2C Bus Interface Enable Used in combination with the IICRST bit to select either IIC or internal reset. 0: Disable (SCL0 and SDA0 pins in inactive state) 1: Enable (SCL0 and SDA0 pins in active state)	R/W

SDAO bit (SDA Output Control/Monitor) and SCLO bit (SCL Output Control/Monitor)

The SDAO and SCLO bits directly control the SDA0 and SCL0 signals output from the IIC. When writing to these bits, also write 0 to the SOWP bit. Setting these bits results in input to the IIC by the input buffer. When slave mode is selected, a start condition might be detected and the bus might be released, depending on the bit settings.

Do not rewrite these bits during a start condition, stop condition, restart condition, transmission, or reception. Operation after rewriting under these conditions is not guaranteed. When reading these bits, the state of signals output from the IIC can be read.

CLO bit (Extra SCL Clock Cycle Output)

The CLO bit allows output of an extra SCL clock cycle for debugging or error processing. Normally, set this bit to 0. Setting the bit to 1 in a normal communication state causes a communication error. For details on this function, see section 25.12.2. Extra SCL Clock Cycle Output Function.

IICRST bit (I2C Bus Interface Internal Reset)

The IICRST bit initiates an internal state reset of the IIC. Setting this bit to 1 initiates an IIC reset or internal reset. Whether an IIC reset or internal reset is initiated is determined by the settings of this bit in combination with the ICE bit. Table 25.3 lists the IIC resets.

The IIC reset initializes all registers except ICCR1.ICE and ICCR1.IICRST bits and internal states of the IIC. In addition to the internal states of the IIC, the internal reset initializes the following:

- Bit counter (ICMR1.BC[2:0] bits)
- I2C Bus Shift Register (ICDRS)
- I2C Bus Status Registers (ICSR1 and ICSR2)
- SDAO and SCLO Output Control/Monitor (ICCR1.SDAO and ICCR1.SCLO bits)

- I2C Bus Control Register 2 (except ICCR2.BBSY bit)

For the reset conditions for each register, see section 25.15. State of Registers When Issuing Each Condition.

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the IIC without initializing the port settings and the control and setting registers of the IIC. If the IIC hangs in a low-level output state, resetting the internal states cancels the low-level output state and releases the bus with the SCL0 pin and SDA0 pin at high impedance.

Note: If an internal reset is initiated using the IICRST bit for a bus hang-up that occurs during communication with the master device in slave mode, the slave and master devices might enter different states, because the bit counter information differs. For this reason, do not initiate an internal reset in slave mode. Initiate recovery processing from the master device. If an internal reset is required because the IIC hangs with the SCL0 line in a low-level output state in slave mode, initiate an internal reset, then issue a restart condition from the master device, or issue a stop condition and resume communication from the start condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start or restart condition from the master device, synchronization is lost because the master and slave devices operate asynchronously.

Table 25.3 IIC resets

IICRST	ICE	State	Specifications
1	0	IIC reset	Resets all registers except ICCR1.IICRST and ICCR1.ICE bits, and the internal states of the IIC
	1	Internal reset	Reset the following: • ICMR1.BC[2:0] bits • ICSR1, ICSR2, ICDRS registers • SDAO and SCLO Output Control/Monitor (ICCR1.SDAO and ICCR1.SCLO bits) • I2C Bus Control Register 2 (except ICCR2.BBSY bit) • Internal states of the IIC

ICE bit (I2C Bus Interface Enable)

The ICE bit selects the active or inactive state of the SCL0 and SDA0 pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 25.3 for the reset descriptions.

Set the ICE bit to 1 when using the IIC. The SCL0 and SDA0 pins are placed in the active state when the ICE bit is set to 1. Set the ICE bit to 0 when the IIC is not used. The SCL0 and SDA0 pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCL0 or SDA0 pin to the IIC when setting up the pin function control. Slave address comparison is performed if the pins are assigned to the IIC.

25.2.2 ICCR2 : I2C Bus Control Register 2

Base address: IIC0 = 0x4005_3000

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	ST	Start Condition Issuance Request 0: Do not issue a start condition request 1: Issue a start condition request	R/W
2	RS	Restart Condition Issuance Request 0: Do not issue a restart condition request 1: Issue a restart condition request	R/W

Bit	Symbol	Function	R/W
3	SP	Stop Condition Issuance Request 0: Do not issue a stop condition request 1: Issue a stop condition request	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	TRS	Transmit/Receive Mode 0: Receive mode 1: Transmit mode	R/W*1
6	MST	Master/Slave Mode 0: Slave mode 1: Master mode	R/W*1
7	BBSY	Bus Busy Detection Flag 0: I2C bus released (bus free state) 1: I2C bus occupied (bus busy state)	R

Note 1. The MST and TRS bits can be written to when the ICMR1.MTWP bit is set to 1.

ST bit (Start Condition Issuance Request)

The ST bit requests transition to master mode and triggers a start condition. When this bit is set to 1, a start condition is issued when the BBSY flag is set to 0 (bus free state). For details on this function, see section 25.11. Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the ST bit.

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition is issued (a start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Only set the ST bit to 1 (start condition request) when the BBSY flag is set to 0 (bus free state). Arbitration might be lost if the ST bit is set to 1 (start condition request) when the BBSY flag is 1 (bus busy state).

RS bit (Restart Condition Issuance Request)

The RS bit requests that a restart condition be issued in master mode. When this bit is set to 1 to request a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on this function, see section 25.11. Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the RS bit with the BBSY flag in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition is issued (a start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Do not set the RS bit to 1 while issuing a stop condition.
Note: If 1 (restart condition request) is written to the RS bit in slave mode, the restart condition is not issued, but the RS bit remains set to 1. If the operating mode changes to master mode without the bit being cleared, a restart condition might be issued.

SP bit (Stop Condition Issuance Request)

The SP bit requests that a stop condition be issued in master mode. When this bit is set to 1, a stop condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on this function, see section 25.11. Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the MST bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition is issued (a stop condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Writing to the SP bit is not possible while the BBSY flag is 0 (bus free state).
Note: Do not set the SP bit to 1 while a restart condition is being issued.

TRS bit (Transmit/Receive Mode)

The TRS bit indicates transmit or receive mode. The IIC is in receive mode when the TRS bit is 0 and in transmit mode when the bit is 1. The combination of this bit and the MST bit indicates the IIC operating mode.

The value of the TRS bit automatically changes to 1 for transmit mode or 0 for receive mode when a start condition is issued or detected and the R/W# bit is set. Although writing to the TRS bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not required during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1)
- When a restart condition is issued normally because of a restart condition request (when a restart condition is detected with the RS bit set to 1)
- When the R/W# bit appended to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in ICSER, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the MTWP bit in ICMR1 set to 1.

[Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When the R/W# bit appended to the slave address is set to 1 in master mode
- In slave mode, on a match between the received address and the address enabled in ICSER when the value of the received R/W# bit is 0, including when the received address is the general call address
- In slave mode, when a restart condition is detected (a restart condition is detected with ICCR2.BBSY = 1 and ICCR2.MST = 0)
- When 0 is written to the TRS bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

MST bit (Master/Slave Mode)

The MST bit indicates master or slave mode. The IIC is in slave mode when the MST bit is 0 and is in master mode when the bit is 1. The combination of this bit and the TRS bit indicates the IIC operating mode.

The value of the MST bit automatically changes to 1 for master mode or 0 for slave mode when a start condition is issued or a stop condition is issued or detected. Although writing to the MST bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not required during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the MTWP bit in ICMR1 set to 1.

[Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 0 is written to the MST bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

BBSY flag (Bus Busy Detection Flag)

The BBSY flag indicates whether the I^2C bus is occupied (bus busy state) or released (bus free state). The flag is set to 1 when the SDA0 line changes from high to low when the SCL0 line is high, assuming that a start condition was issued. The flag is set to 0 if a start condition is not detected for the bus free time (ICBRL setting), assuming that a stop condition was issued.

[Setting condition]

- When a start condition is detected.

[Clearing conditions]

- When a start condition is not detected for the bus free time (ICBRL setting) after detecting a stop condition
- When 1 is written to the IICRST bit in ICCR1 with the ICE bit in ICCR1 set to 0 (IIC reset).

25.2.3 ICMR1 : I^2C Bus Mode Register 1

Base address: IIC0 = 0x4005_3000

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
2:0	BC[2:0]	Bit Counter 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits	R/W*1
3	BCWP	BC Write Protect This bit is read as 1. 0: Write enable BC[2:0] bits 1: Write protect BC[2:0] bits	W*1
6:4	CKS[2:0]	Internal Reference Clock Select Select the internal reference clock source (IICφ) for the IIC. IICφ = (PCLKB / $2^{CKS[2:0]}$) clock	R/W
7	MTWP	MST/TRS Write Protect 0: Write protect MST and TRS bits in ICCR2 1: Write enable MST and TRS bits in ICCR2	R/W

Note 1. Rewrite the BC[2:0] bits and set the BCWP bit to 0 at the same time.

BC[2:0] bits (Bit Counter)

The BC[2:0] bits function as a counter indicating the number of bits remaining to be transferred on detection of a rising edge on the SCL0 line. Although BC[2:0] are read/write bits, it is not required to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one, for an additional acknowledge bit, between transferred frames when the SCL0 line is at a low level. The value in the BC[2:0] bits returns to 000b at the end of a data transfer, including the acknowledge bit, or when a start or restart condition is detected.

25.2.4 ICMR2 : I2C Bus Mode Register 2

Base address: IIC0 = 0x4005_3000

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
0	TMOS	Timeout Detection Time Select 0: Select long mode 1: Select short mode	R/W
1	TMOL	Timeout L Count Control 0: Disable count while SCL0 line is low 1: Enable count while SCL0 line is low	R/W
2	TMOH	Timeout H Count Control 0: Disable count while SCL0 line is high 1: Enable count while SCL0 line is high	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	SDDL[2:0]	SDA Output Delay Counter 0 0 0: No output delay 0 0 1: 1 IICφ cycle (When ICMR2.DLCS = 0 (IICφ)) 1 or 2 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 0 1 0: 2 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 3 or 4 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 0 1 1: 3 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 5 or 6 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 0 0: 4 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 7 or 8 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 0 1: 5 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 9 or 10 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 1 0: 6 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 11 or 12 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 1 1: 7 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 13 or 14 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2))	R/W
7	DLCS	SDA Output Delay Clock Source Select 0: Select internal reference clock (IICφ) as the clock source for SDA output delay counter 1: Select internal reference clock divided by 2 (IICφ/2) as the clock source for SDA output delay counter*1	R/W

Note 1. The setting DLCS = 1 (IICφ/2) is only valid when SCL is low. When SCL is high, the DLCS = 1 setting becomes invalid and the clock source becomes the internal reference clock (IICφ).

TMOS bit (Timeout Detection Time Select)

The TMOS bit selects long or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE = 1). When this bit is set to 0, long mode is selected. When it is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a 14- bit counter. While the SCL0 line is in the state that enables this counter as specified in the TMOH and TMOL bits, the counter counts up in synchronization with the internal reference clock (IICφ) as a count source. For details on this function, see section 25.12.1. Timeout Function.

TMOL bit (Timeout L Count Control)

The TMOL bit enables or disables up-counting on the internal counter of the timeout function while the SCL0 line is held low and the timeout function is enabled (ICFER.TMOE = 1).

TMOH bit (Timeout H Count Control)

The TMOH bit enables or disables up-counting on the internal counter of the timeout function while the SCL0 line is held high and the timeout function is enabled (ICFER.TMOE = 1).

SDDL[2:0] bits (SDA Output Delay Counter)

The SDDL[2:0] bits can be used to delay the SDA output. This counter works with the clock source selected in the DLCS bit. This setting can be used for all types of SDA output, including transmission of the acknowledge bit.

Set the SDA output delay to meet the I2C bus standard for the data enable time/acknowledge enable time,*1 or the SMBus standard, within [data hold time (300 ns or more + the SCL clock low-level period) - the data setup time (250 ns)]. If a value outside the standard is set, communication between devices might malfunction or falsely indicate a start or stop condition, depending on the bus state.

For details on this function, see section 25.5. SDA Output Delay Function.

Note 1. Data enable time/acknowledge enable time.
3,450 ns for up to 100 kbps: Standard mode (Sm)
900 ns for up to 400 kbps: Fast mode (Fm)

25.2.5 ICMR3 : I2C Bus Mode Register 3

Base address: IIC0 = 0x4005_3000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	NF[1:0]	Noise Filter Stage Select 0 0: Filter out noise of up to 1 IICφ cycle (single-stage filter) 0 1: Filter out noise of up to 2 IICφ cycles (2-stage filter) 1 0: Filter out noise of up to 3 IICφ cycles (3-stage filter) 1 1: Filter out noise of up to 4 IICφ cycles (4-stage filter)	R/W
2	ACKBR	Receive Acknowledge 0: 0 received as the acknowledge bit (ACK reception) 1: 1 received as the acknowledge bit (NACK reception)	R
3	ACKBT	Transmit Acknowledge 0: Send 0 as the acknowledge bit (ACK transmission) 1: Send 1 as the acknowledge bit (NACK transmission)	R/W*1
4	ACKWP	ACKBT Write Protect 0: Write protect ACKBT bit 1: Write enable ACKBT bit	R/W
5	RDRFS	RDRF Flag Set Timing Select Low-hold is released by writing to ACKBT. 0: Set the RDRF flag on the rising edge of the 9th SCL clock cycle. The SCLn line is not held low on the falling edge of the 8th clock cycle. 1: Set the RDRF flag on the rising edge of the 8th SCL clock cycle. The SCLn line is held low on the falling edge of the 8th clock cycle.	R/W*2
6	WAIT	Low-hold is released by reading ICDRR. 0: No wait (The SCLn line is not held low during the period between the 9th clock cycle and the 1st clock cycle.) 1: Wait (The SCLn line is held low during the period between the 9th clock cycle and the 1st clock cycle.)	R/W*2

Bit	Symbol	Function	R/W
7	SMBS	SMBus/I²C Bus Select 0: Select I²C Bus 1: Select SMBus	R/W

Note 1. Write to the ACKBT bit only while the ACKWP bit is already 1. If the application writes 1 to the ACKWP and ACKBT bits at the same time, the ACKBT bit is not set to 1.
Note 2. The WAIT and RDRFS bits are only valid in receive mode (invalid in transmit mode).

NF[1:0] bits (Noise Filter Stage Select)

The NF[1:0] bits select the number of stages in the digital noise filter. For details on this function, see section 25.6. Digital Noise Filter Circuits

Note: Set the noise range to be filtered within a range less than the SCL0 line high- or low-level period. If the noise range is set to a value of [SCL clock width: high- or low-level period, whichever is shorter] - [1.5 internal reference clock (IICφ) cycles + analog noise filter: 120 ns (reference values)] or more, the SCL clock is regarded as noise, which might prevent the IIC from operating normally.

ACKBR bit (Receive Acknowledge)

The ACKBR bit stores the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

- When 1 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (IIC reset).

ACKBT bit (Transmit Acknowledge)

The ACKBT bit sets the acknowledge bit to be sent in receive mode

[Setting condition]

- When 1 is written to this bit with the ACKWP bit set to 1.

[Clearing conditions]

- When 0 is written to this bit with the ACKWP bit set to 1
- When stop condition request is detected with the SP bit in ICCR2 set to 1
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (IIC reset).

ACKWP bit (ACKBT Write Protect)

The ACKWP bit controls write enabling of the ACKBT bit.

RDRFS bit (RDRF Flag Set Timing Select)

The RDRFS bit selects the RDRF flag set timing in receive mode and also selects whether to hold the SCL0 line low on the falling edge of the 8th SCL clock cycle.

When the RDRFS bit is 0, the SCL0 line is not held low on the falling edge of the 8th SCL clock cycle, and the RDRF flag is set to 1 on the rising edge of the 9th SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 on the rising edge of the 8th SCL clock cycle, and the SCL0 line is held low on the falling edge of the 8th SCL clock cycle. The low-hold of the SCL0 line is released by a write to the ACKBT bit.

After data is received with this setting, the SCL0 line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1), based on the receive data.

WAIT bit (WAIT)

The WAIT bit controls whether to force a low-hold between the ninth SCL clock cycle and the first SCL clock cycle, until the receive data buffer (ICDRR) is completely read each time single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation continues without a low-hold between the ninth and the first SCL clock cycle. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCL0 line is held low from the falling edge of the ninth clock cycle until the ICDRR value is read each time single-byte data is received. This enables receive operation in byte units.

Note: When the value of the WAIT bit is to be read, always read ICDRR first.

SMBS bit (SMBus/I2C Bus Select)

Setting the SMBS bit to 1 selects the SMBus and enables the HOAE bit in ICSER.

25.2.6 ICFER : I2C Bus Function Enable Register

Base address: IIC0 = 0x4005_3000

Offset address: 0x05

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Function	R/W
0	TMOE	Timeout Function Enable 0: Disable 1: Enable	R/W
1	MALE	Master Arbitration-Lost Detection Enable 0: Disable the arbitration-lost detection function and disable automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost 1: Enable the arbitration-lost detection function and enable automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost	R/W
2	NALE	NACK Transmission Arbitration-Lost Detection Enable 0: Disable 1: Enable	R/W
3	SALE	Slave Arbitration-Lost Detection Enable 0: Disable 1: Enable	R/W
4	NACKE	NACK Reception Transfer Suspension Enable 0: Do not suspend transfer operation during NACK reception (disable transfer suspension) 1: Suspend transfer operation during NACK reception (enable transfer suspension)	R/W
5	NFE	Digital Noise Filter Circuit Enable 0: Do not use the digital noise filter circuit 1: Use the digital noise filter circuit	R/W
6	SCLE	SCL Synchronous Circuit Enable 0: Do not use the SCL synchronous circuit 1: Use the SCL synchronous circuit	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

TMOE bit (Timeout Function Enable)

The TMOE bit enables or disables the timeout function. For details on this function, see section 25.12.1. Timeout Function.

MALE bit (Master Arbitration-Lost Detection Enable)

The MALE bit specifies whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE bit (NACK Transmission Arbitration-Lost Detection Enable)

The NALE bit specifies whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode, for example when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with a different number of receive bytes.

SALE bit (Slave Arbitration-Lost Detection Enable)

The SALE bit specifies whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode, for example when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs because of noise.

NACKE bit (NACK Reception Transfer Suspension Enable)

The NACKE bit specifies whether to continue or discontinue the transfer operation when NACK is received in transmit mode. Normally, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended. When the NACKE bit is 0, the next transfer operation continues regardless of the received acknowledge content.

For details, see section 25.9.2. NACK Reception Transfer Suspension Function.

SCLE bit (SCL Synchronous Circuit Enable)

The SCLE bit specifies whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.

When the SCLE bit is set to 0 (no SCL synchronous circuit used), the IIC does not synchronize the SCL clock with the SCL input clock. With this setting, the IIC outputs the SCL clock at the transfer rate set in ICBRH and ICBRL, regardless of the SCL0 line state. For this reason, if the bus load of the I^{2}C bus line is much larger than the specification value, or if the SCL clock output overlaps in multiple masters, a short-cycle SCL clock that does not meet the specification might be output. When no SCL synchronous circuit is used, it also affects the issuing of the start, restart, and stop conditions, and the continuous output of extra SCL clock cycles.

Do not set this bit to 0 except when checking the output of the set transfer rate.

25.2.7 ICSER : I^{2}C Bus Status Enable Register

Base address: IIC0 = 0x4005_3000

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Function	R/W
0	SAR0E	Slave Address Register 0 Enable 0: Disable slave address in SARL0 and SARU0 1: Enable slave address in SARL0 and SARU0	R/W
1	SAR1E	Slave Address Register 1 Enable 0: Disable slave address in SARL1 and SARU1 1: Enable slave address in SARL1 and SARU1	R/W
2	SAR2E	Slave Address Register 2 Enable 0: Disable slave address in SARL2 and SARU2 1: Enable slave address in SARL2 and SARU2	R/W
3	GCAE	General Call Address Enable 0: Disable general call address detection 1: Enable general call address detection	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DIDE	Device-ID Address Detection Enable 0: Disable device-ID address detection 1: Enable device-ID address detection	R/W

Bit	Symbol	Function	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	HOAE	Host Address Enable 0: Disable host address detection 1: Enable host address detection	R/W

SARyE bit (Slave Address Register y Enable) (y = 0 to 2)

The SARyE bit enables or disables the received slave address and the slave address set in SARLy and SARUy.

When this bit is set to 1, the slave address set in SARLy and SARUy is enabled and is compared with the received slave address. When this bit is set to 0, the slave address set in SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

GCAE bit (General Call Address Enable)

The GCAE bit specifies whether to ignore the general call address (0000 000b + 0 [W]: All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the IIC recognizes the received slave address as the general call address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs the data receive operation. When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

DIDE bit (Device-ID Address Detection Enable)

The DIDE bit specifies whether to recognize and execute the device-ID address when a device ID (1111 100b) is received in the first frame after a start or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the IIC recognizes that the device-ID address was received. When the next R/W# bit is 0 (W), the IIC recognizes the second and the subsequent frames as slave addresses and continues the receive operation. When this bit is set to 0, the IIC ignores the received first frame even if it matches the device-ID address, and it recognizes the first frame as a normal slave address.

For details on this function, see section 25.7.3. Device-ID Address Detection.

HOAE bit (Host Address Enable)

The HOAE bit specifies whether to ignore the received host address (0001 000b) when the SMBS bit in ICMR3 is 1.

When this bit is set to 1 while the SMBS bit in ICMR3 is 1, if the received slave address matches the host address, the IIC recognizes the received slave address as the host address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs the receive operation.

When the SMBS bit in ICMR3 or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

25.2.8 ICIER : I2C Bus Interrupt Enable Register

Base address: IIC0 = 0x4005_3000

Offset address: 0x07

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMOIE	Timeout Interrupt Request Enable 0: Disable timeout interrupt (TMOI) request 1: Enable timeout interrupt (TMOI) request	R/W
1	ALIE	Arbitration-Lost Interrupt Request Enable 0: Disable arbitration-lost interrupt (ALI) request 1: Enable arbitration-lost interrupt (ALI) request	R/W

Bit	Symbol	Function	R/W
2	STIE	Start Condition Detection Interrupt Request Enable 0: Disable start condition detection interrupt (STI) request 1: Enable start condition detection interrupt (STI) request	R/W
3	SPIE	Stop Condition Detection Interrupt Request Enable 0: Disable stop condition detection interrupt (SPI) request 1: Enable stop condition detection interrupt (SPI) request	R/W
4	NAKIE	NACK Reception Interrupt Request Enable 0: Disable NACK reception interrupt (NAKI) request 1: Enable NACK reception interrupt (NAKI) request	R/W
5	RIE	Receive Data Full Interrupt Request Enable 0: Disable receive data full interrupt (IIC0_RXI) request 1: Enable receive data full interrupt (IIC0_RXI) request	R/W
6	TEIE	Transmit End Interrupt Request Enable 0: Disable transmit end interrupt (IIC0_TEI) request 1: Enable transmit end interrupt (IIC0_TEI) request	R/W
7	TIE	Transmit Data Empty Interrupt Request Enable 0: Disable transmit data empty interrupt (IIC0_TXI) request 1: Enable transmit data empty interrupt (IIC0_TXI) request	R/W

TMOIE bit (Timeout Interrupt Request Enable)

The TMOIE bit enables or disables timeout interrupt (TMOI) requests when the TMOF flag in ICSR2 is 1. To cancel a TMOI interrupt request, set the TMOF flag or the TMOIE bit to 0.

ALIE bit (Arbitration-Lost Interrupt Request Enable)

The ALIE bit enables or disables arbitration-lost interrupt (ALI) requests when the AL flag in ICSR2 is 1. To cancel an ALI interrupt request, set the AL flag or the ALIE bit to 0.

STIE bit (Start Condition Detection Interrupt Request Enable)

The STIE bit enables or disables start condition detection interrupt (STI) requests when the START flag in ICSR2 is 1. To cancel an STI interrupt request, set the START flag or the STIE bit to 0.

SPIE bit (Stop Condition Detection Interrupt Request Enable)

The SPIE bit enables or disables stop condition detection interrupt (SPI) requests when the STOP flag in ICSR2 is 1. To cancel an SPI interrupt request, set the STOP flag or the SPIE bit to 0.

NAKIE bit (NACK Reception Interrupt Request Enable)

The NAKIE bit enables or disables NACK reception interrupt (NAKI) requests when the NACKF flag in ICSR2 is 1. To cancel an NAKI interrupt request, set the NACKF flag or the NAKIE bit to 0.

RIE bit (Receive Data Full Interrupt Request Enable)

The RIE bit enables or disables receive data full interrupt (IIC0_RXI) requests when the RDRF flag in ICSR2 is 1.

TEIE bit (Transmit End Interrupt Request Enable)

The TEIE bit enables or disables transmit end interrupt (IIC0_TEI) requests when the TEND flag in ICSR2 is 1. To cancel an IIC0_TEI interrupt request, set the TEND flag or the TEIE bit to 0.

TIE bit (Transmit Data Empty Interrupt Request Enable)

The TIE bit enables or disables transmit data empty interrupt (IIC0_TXI) requests when the TDRE flag in ICSR2 is 1.

25.2.9 ICSR1 : I²C Bus Status Register 1

Base address: IIC0 = 0x4005_3000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AAS0	Slave Address 0 Detection Flag 0: Slave address 0 not detected 1: Slave address 0 detected	R/(W)*1
1	AAS1	Slave Address 1 Detection Flag 0: Slave address 1 not detected 1: Slave address 1 detected	R/(W)*1
2	AAS2	Slave Address 2 Detection Flag 0: Slave address 2 not detected 1: Slave address 2 detected	R/(W)*1
3	GCA	General Call Address Detection Flag 0: General call address not detected 1: General call address detected	R/(W)*1
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DID	Device-ID Address Detection Flag This bit is set to 1 when the first frame received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0[W]). 0: Device-ID command not detected 1: Device-ID command detected	R/(W)*1
6	—	This bit is read as 0. The write value should be 0.	R/W
7	HOA	Host Address Detection Flag This bit is set to 1 when the received slave address matches the host address (0001 000b). 0: Host address not detected 1: Host address detected	R/(W)*1

Note 1. Only 0 can be written to clear the flag.

AASy flag (Slave Address y Detection flag) (y = 0 to 2)

The AASy flag indicates whether slave address y was detected.

[Setting conditions]

For 7-bit address format (SARUy.FS = 0):

- When the received slave address matches the SVA[6:0] value in SARLy, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

For 10-bit address format: (SARUy.FS = 1):

- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy), and the subsequent address matches the SARLy value, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy flag after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

For 7-bit address format (SARUy.FS = 0):

- When the received slave address does not match the SVA[6:0] value in SARLy, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.

For 10-bit address format (SARUy.FS = 1):

- When the received slave address does not match a value of (11110b + SVA[1:0] in SARUy), with the SARyE bit in ICSER set to 1 (slave address y detection enabled)
  The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy), and the subsequent address does not match the SARLy value, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.

GCA flag (General Call Address Detection Flag)

The GCA flag indicates whether the general call address was detected.

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 [W]), with the GCAE bit in ICSER set to 1 (general call address detection enabled).
  The GCA flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA flag after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 [W]), with the GCAE bit in ICSER set to 1 (general call address detection enabled)
  The GCA flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

DID flag (Device-ID Address Detection Flag)

The DID flag indicates whether the device-ID address was detected.

[Setting condition]

- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), with the DIDE bit in ICSER set to 1 (device-ID address detection enabled).
  The DID flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID flag after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start or restart condition is detected does not match a value of (device ID (1111 100b)), with the DIDE bit in ICSER set to 1 (device-ID address detection enabled)
  The DID flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), and the second frame does not match any slave address from 0 to 2, with the DIDE bit in ICSER set to 1 (device-ID address detection enabled)
  The DID flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

HOA flag (Host Address Detection Flag)

The HOA flag indicates whether the host address was detected.

[Setting condition]

- When the received slave address matches the host address (0001 000b), with the HOAE bit in ICSER set to 1 (host address detection enabled).

The HOA flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA flag after reading HOA = 1
- When a stop condition is detected
- When the received slave address does not match the host address (0001 000b), with the HOAE bit in ICSER set to 1 (host address detection enabled)
  The HOA flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

25.2.10 ICSR2 : I2C Bus Status Register 2

Base address: IIC0 = 0x4005_3000

Offset address: 0x09

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMOF	Timeout Detection Flag 0: Timeout not detected 1: Timeout detected	R/(W)*1
1	AL	Arbitration-Lost Flag 0: Arbitration not lost 1: Arbitration lost	R/(W)*1
2	START	Start Condition Detection Flag 0: Start condition not detected 1: Start condition detected	R/(W)*1
3	STOP	Stop Condition Detection Flag 0: Stop condition not detected 1: Stop condition detected	R/(W)*1
4	NACKF	NACK Detection Flag 0: NACK not detected 1: NACK detected	R/(W)*1
5	RDRF	Receive Data Full Flag 0: ICDRR contains no receive data 1: ICDRR contains receive data	R/(W)*1
6	TEND	Transmit End Flag 0: Data being transmitted 1: Data transmit complete	R/(W)*1
7	TDRE	Transmit Data Empty Flag 0: ICDRT contains transmit data 1: ICDRT contains no transmit data	R

Note 1. Only 0 can be written, to clear the flag.

TMOF flag (Timeout Detection Flag)

The TMOF flag is set to 1 when the IIC detects a timeout because the SCL0 line state remains unchanged for the set period.

[Setting condition]

- When the SCL0 line state remains unchanged for the period specified in the ICMR2.TMOH, TMOL, and TMOS bits while the ICFER.TMOE bit is 1 (timeout function enabled) in master or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF flag after reading TMOF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

AL flag (Arbitration-Lost Flag)

The AL flag indicates that bus mastership was lost in arbitration because of a bus conflict or some other reason when a start condition was issued or an address and data was transmitted. The IIC monitors the level on the SDA0 line during transmission and, if the level on the line does not match the value of the bit being output, is set the value of the AL flag to 1 to indicate that the bus is occupied by another device.

The IIC can also set the AL flag to indicate the detection of arbitration loss during NACK transmission or during data transmission.

[Setting conditions]

When master arbitration-lost detection is enabled (ICFER.MALE = 1):

- When the internal SDA output state does not match the SDA0 line level on the rising edge of the SCL clock except for the ACK period during data transmission in master transmit mode
- When a start condition is detected while the ST bit in ICCR2 is 1 (start condition requested) or the internal SDA output state does not match the SDA0 line level
- When the ST bit in ICCR2 is 1 (start condition requested), with the BBSY flag in ICCR2 set to 1.

When NACK arbitration-lost detection is enabled (ICFER.NALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock in the ACK period during NACK transmission in receive mode.

When slave arbitration-lost detection is enabled (ICFER.SALE = 1):

- When the internal SDA output state does not match the SDA0 line level on the rising edge of the SCL clock, except for the ACK period during data transmission in slave transmit mode.

[Clearing conditions]

- When 0 is written to the AL flag after reading AL = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Table 25.4 Relationship between arbitration-lost generation sources and arbitration-lost enable functions

ICFER			ICSR2		
MALE	NALE	SALE	AL	Error	Arbitration-lost generation source
1	x	x	1	Start condition issuance error	When internal SDA output state does not match SDA0 line level when a start condition is detected, while the ST bit in ICCR2 is 1
					When ST in ICCR2 is set to 1 while BBSY in ICCR2 is 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
x	1	x	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master or slave receive mode
x	x	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

x: Don’t care

START flag (Start Condition Detection Flag)

The START flag indicates whether a start or restart condition was detected.

[Setting condition]

- When a start or restart condition is detected.

[Clearing conditions]

- When 0 is written to the START flag after reading START = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

STOP flag (Stop Condition Detection Flag)

The STOP flag indicates whether a stop condition was detected.

[Setting condition]

- When a stop condition is detected.

[Clearing conditions]

- When 0 is written to the STOP flag after reading STOP = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

NACKF flag (NACK Detection Flag)

The NACKF flag indicates whether a NACK was detected.

[Setting condition]

- When acknowledge is not received (NACK received) from the receive device in transmit mode, with the NACKE bit in ICFER set to 1 (transfer suspension enabled).

[Clearing conditions]

- When 0 is written to the NACKF flag after reading NACKF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1, the IIC suspends data transmission and reception. Writing to ICDRT in transmit mode or reading from ICDRR in receive mode with the NACKF flag set to 1 does not enable data transmit or receive operation. To restart data transmission or reception, set the NACKF flag to 0.

RDRF flag (Receive Data Full Flag)

The RDRF flag indicates whether the ICDRR contains receive data.

[Setting conditions]

- When receive data is transferred from ICDRS to ICDRR
  The RDRF flag is set to 1 on the rising edge of the eighth or ninth SCL clock cycle (selected in the RDRFS bit in ICMR3).
- When the received slave address matches after a start or restart condition is detected with the TRS bit in ICCR2 set to 0.

[Clearing conditions]

- When 0 is written to the RDRF flag after reading RDRF = 1
- When data is read from ICDRR
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

TEND flag (Transmit End Flag)

The TEND flag indicates completion of transmission.

[Setting condition]

- On the rising edge of the ninth SCL clock cycle while the TDRE flag is 1.

[Clearing conditions]

- When 0 is written to the TEND flag after reading TEND = 1
- When data is written to ICDRT
- When a stop condition is detected

- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

TDRE flag (Transmit Data Empty Flag)

The TDRE flag indicates whether the ICDRT contains transmit data.

[Setting conditions]

- When data is transferred from ICDRT to ICDRS and ICDRT becomes empty
- When the TRS bit in ICCR2 is set to 1
- When the received slave address matches while the TRS bit is 1.

[Clearing conditions]

- When data is written to ICDRT
- When the TRS bit in ICCR2 is set to 0
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1 while the NACKE bit in ICFER is 1, the IIC suspends data transmission and reception. In this case, if the TDRE flag is 0 (next transmit data written), data is transferred to the ICDRS register and the ICDRT register becomes empty on the rising edge of the 9th clock cycle, but the TDRE flag does not set to 1.

25.2.11 ICWUR : I2C Bus Wakeup Unit Register

Base address: IIC0WU = 0x4005_3014

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	WUE	WUIE	WUF	WUACK	—	—	—	WUAFA
Value after reset:	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	WUAFA	Wakeup Analog Filter Additional Selection 0: Do not add the wakeup analog filter 1: Add the wakeup analog filter	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	WUACK	ACK Bit for Wakeup Mode Choice of four response modes in combination with ICCR1.IICRST and WUACK. See Table 25.5.	R/W
5	WUF	Wakeup Event Occurrence Flag 0: Slave address not matching during wakeup 1: Slave address matching during wakeup	R/W
6	WUIE	Wakeup Interrupt Request Enable 0: Disable wakeup interrupt request (IIC0_WUI) 1: Enable wakeup interrupt request (IIC0_WUI)	R/W
7	WUE	Wakeup Function Enable 0: Disable wakeup function 1: Enable wakeup function	R/W

Table 25.5 Wakeup mode (1 of 2)

IICRST	WUACK	Operation mode	Description
0	0	Normal wakeup mode 1	ACK response on 9th SCL, and SCL low-hold after 9th SCL.
0	1	Normal wakeup mode 2	No ACK response immediately and SCL low-hold between 8th and 9th SCL. SCL low-hold release and ACK response on 9th SCL.
1	0	Command recovery mode	ACK response on 9th SCL and no SCL low-hold.

Table 25.5 Wakeup mode (2 of 2)

IICRST	WUACK	Operation mode	Description
1	1	EEP response mode	NACK response on 9th SCL and no SCL low-hold.

WUF flag (Wakeup Event Occurrence Flag)

The WUF flag indicates whether the slave address is matching during wakeup.

[Setting condition]

- When PCLKB is supplied after a slave-address match in the first 8th SCL low during wakeup mode.

[Clearing conditions]

- When 0 is written to the WUF flag after reading WUF = 1
- When ICE = 0 and IICRST = 1.

25.2.12 ICWUR2 : I2C Bus Wakeup Unit Register 2

Base address: IIC0WU = 0x4005_3014

Offset address: 0x03

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

Bit	Symbol	Function	R/W
0	WUSEN	Wakeup Function Synchronous Enable 0: IIC asynchronous circuit enable 1: IIC synchronous circuit enable	R/W
1	WUASYF	Wakeup Function Asynchronous Operation Status Flag 0: IIC synchronous circuit enable condition 1: IIC asynchronous circuit enable condition	R
2	WUSYF	Wakeup Function Synchronous Operation Status Flag 0: IIC asynchronous circuit enable condition 1: IIC synchronous circuit enable condition	R
7:3	—	These bits are read as 1. The write value should be 1.	R/W

WUSEN bit (Wakeup Function Synchronous Enable)

The WUSEN bit is used in combination with the WUASYF flag (or WUSYF flag) to switch between the PCLKB synchronous and asynchronous operation, when the wakeup effective function is enabled (ICWUR.WUE = 1).

The PCLKB operation switches from synchronous to asynchronous operation:

When the ICCR2.BBSY flag is 0, if 0 is written to the WUSEN bit while the WUASYF flag is 0, the reception occurs independently of the operation of PCLKB (with PCLKB stopped) after it switches to the PCLKB asynchronous operation, on wakeup event detection.

The PCLKB operation switches from asynchronous to synchronous operation:

- When 1 is written to the WUSEN bit, with the WUASYF flag at 1, when a wakeup event is detected. After writing 1, the WUASYF flag immediately becomes 0.
- When the stop condition is detected with a wakeup event undetected.

WUASYF flag (Wakeup Function Asynchronous Operation Status Flag)

The WUASYF flag can place the IIC in PCLKB asynchronous operation when the wakeup effective function is enabled (ICWUR.WUE = 1).

[Setting condition]

- When the ICCR2.BBSY flag is 0, and the WUSEN bit is set to 0 with the ICWUR.WUE bit set to 1.

[Clearing conditions]

- When 1 is written to the WUSEN bit after detecting the wake-up event with ICWUR.WUE bit set to 1.
- When a stop condition is detected with WUSEN bit set to 1 before detecting the wake-up event with WUASY flag set to 1 with ICWUR.WUE bit set to 1.
- When you write 1 in the WUSEN bit with the WUASYF flag detected 1 and the wake-up event in the state of ICWUR.WUE = 1.
- ICCR1.ICE = 0 and ICCRST = 1 (ICC reset)
- ICWUR.WUE = 0.

WUSYF flag (Wakeup Function Synchronous Operation Status Flag)

It is shown that IIC is in the PCLKB synchronous operation at wake-up effective function (ICWUR.WUE = 1). This flag is a value in which the WUASYF flag is always reserved.

[Setting conditions]

- When 1 is written to the WUSEN bit after detecting the wake-up event with ICWUR.WUE bit set to 1 with WUSYF flag set to 0 with ICWUR.WUE bit set to 1.
- When a stop condition is detected with WUSEN bit set to 1 before detecting the wake-up event with WUSYF flag set to 0 with ICWUR.WUE bit set to 1.
- ICCR1.ICE = 0 and ICCRST = 1 (ICC reset)
- ICWUR.WUE = 0.

[Clearing condition]

- When the ICCR2.BBSY flag is 0 with the ICWUR.WUE bit set to 1 after writing 0 to the WUSEN bit.

25.2.13 SARLy : Slave Address Register L y (y = 0 to 2)

Base address: IIC0 = 0x4005_3000

Offset address: 0x0A + 0x02 × y

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SVA[6:0]							SVA0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SVA0	10-bit Address LSB Slave address setting.	R/W
7:1	SVA[6:0]	7-bit Address/10-bit Address Lower Bits Slave address setting.	R/W

SVA0 bit (10-bit Address LSB)

When the 10-bit address format is selected (SARUy.FS = 1), the SVA0 bit functions as the LSB of a 10-bit address and is combined with the SVA[6:0] bits to form the lower 8 bits of a 10-bit address.

This bit is valid when the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting in this bit is ignored.

SVA[6:0] bits (7-bit Address/10-bit Address Lower Bits)

When the 7-bit address format is selected (SARUy.FS = 0), the SVA[6:0] bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS = 1), these bits combine with the SVA0 bit to form the lower 8 bits of a 10-bit address.

When the SARyE bit in ICSER is 0, the setting in these bits is ignored.

25.2.14 SARUy : Slave Address Register U y (y = 0 to 2)

Base address: IIC0 = 0x4005_3000

Offset address: 0x0B + 0x02 × y

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

Bit	Symbol	Function	R/W
0	FS	7-bit/10-bit Address Format Select 0: Select 7-bit address format 1: Select 10-bit address format	R/W
2:1	SVA[1:0]	10-bit Address Upper Bits Slave address setting.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

FS bit (7-bit/10-bit Address Format Select)

The FS bit selects 7- or 10-bit format for slave address y (in SARLy and SARUy).

When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SVA[6:0] setting in SARLy is valid, and the SVA[1:0] and SVA0 settings in SARLy are ignored.

When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the SVA[1:0] and SARLy settings are valid.

When the SARyE bit in ICSER is 0 (SARLy and SARUy disabled), the SARUy.FS setting is invalid.

SVA[1:0] bits (10-bit Address Upper Bits)

When the 10-bit address format is selected (FS = 1), the SVA[1:0] bits function as the upper 2 bits of a 10-bit address.

These bits are valid when the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting in these bits is ignored.

25.2.15 ICBRL : I2C Bus Bit Rate Low-Level Register

Base address: IIC0 = 0x4005_3000

Offset address: 0x10

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

Bit	Symbol	Function	R/W
4:0	BRL[4:0]	Bit Rate Low-Level Period Low-level period of SCL clock.	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W

BRL[4:0] bits (Bit Rate Low-Level Period)

The BRL[4:0] bits set the low-level period of the SCL clock. ICBRL counts the low-level period with the internal reference clock source (IICφ) specified by the CKS[2:0] bits in ICMR1. ICBRL also generates the data setup time for automatic SCL low-hold operation, see section 25.9. Automatic Low-Hold Function for SCL. When the IIC is used in slave mode, the BRL[4:0] bits must be set to a value longer than the data setup time*1.

If the digital noise filter is enabled (NFE bit in ICFER is 1), set the BRL[4:0] bits to a value at least one greater than the number of stages in the noise filter. For details on the number of stages, see the description of the NF[1:0] bits in section 25.2.5. ICMR3 : I2C Bus Mode Register 3.

Note 1. Data setup time (t_{SU}: DAT)

250 ns for up to 100 kbps: Standard-mode (Sm)

100 ns for up to 400 kbps: Fast-mode (Fm)

25.2.16 ICBRH : I2C Bus Bit Rate High-Level Register

Base address: IIC0 = 0x4005_3000

Offset address: 0x11

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

Bit	Symbol	Function	R/W
4:0	BRH[4:0]	Bit Rate High-Level Period High-level period of SCL clock.	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W

BRH[4:0] bits (Bit Rate High-Level Period)

The BRH[4:0] bits set the high-level period of SCL clock. BRH[4:0] bits are valid in master mode. If the IIC is used only in slave mode, do not set the BRH[4:0] bits.

ICBRH counts the high-level period with the internal reference clock source (IICφ) specified in the CKS[2:0] bits in ICMR1.

If the digital noise filter is enabled (the NFE bit in ICFER is 1), set these bits to a value at least one greater than the number of stages in the noise filter. For the number of stages in the noise filter, see the description of the NF[1:0] bits in section 25.2.5. ICMR3 : I2C Bus Mode Register 3.

The IIC transfer rate and the SCL clock duty are calculated using the following expressions (1) to (5):

1. ICFER.SCLE = 0
   Transfer rate = 1/[{(BRH + 1) + (BRL + 1)}/IICφ*1 + tr*2 + tf*2]
   Duty cycle = [tr + {(BRH + 1)/IICφ}]/[tr + tf + {(BRH + 1) + (BRL + 1)}/IICφ]
2. ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] = 000b (IICφ = PCLKB)
   Transfer rate = 1/[{(BRH + 3) + (BRL+ 3)}/IICφ + tr + tf]
   Duty cycle = [tr + {(BRH + 3)/IICφ}]/[tr + tf + {(BRH + 3) + (BRL + 3)}/IICφ]
3. ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] = 000b (IICφ = PCLKB)
   Transfer rate = 1/[{(BRH + 3 + nf*3) + (BRL + 3 + nf)}/IICφ + tr + tf]
   Duty cycle = [tr + {(BRH + 3 + nf)/IICφ}]/[tr + tf + {(BRH + 3 + nf) + (BRL + 3 + nf)}/IICφ]
4. ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] ≠ 000b
   Transfer rate = 1/[{(BRH + 2) + (BRL + 2)}/IICφ + tr + tf]
   Duty cycle = [tr + {(BRH + 2)/IICφ}]/[tr + tf + {(BRH + 2) + (BRL + 2)}/IICφ]
5. ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] ≠ 000b
   Transfer rate = 1/[{(BRH + 2 + nf) + (BRL + 2 + nf)}/IICφ + tr + tf]
   Duty cycle = [tr + {(BRH + 2 + nf)/IICφ}]/[tr + tf + {(BRH + 2 + nf) + (BRL + 2 + nf)}/IICφ]

Note 1. IICφ = PCLKB × division ratio

Note 2. The SCLn line rise time (tr) and SCLn line fall time (tf) depend on the total bus line capacitance (Cb) and the pull-up resistor (Rp). For details, see the I2C bus standard from NXP Semiconductors.

Note 3. nf = Number of digital noise filters selected in the ICMR3.NF bit.

Table 25.6 Example of ICBRH/ICBRL Settings for Transfer Rate IIC when SCLE = 0

Transfer rate (kbps)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	PCLKB (MHz)	NF[1:0]	Computation expression
100	011b	18 (0xF2)	16 (0x10)	32	—	(1)
400	001b	9 (0xE9)	20 (0xF4)	32	—	(1)

Table 25.7 Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 1 and NFE = 0

Transfer rate (kbps)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	PCLKB (MHz)	NF[1:0]	Computation expression
100	011b	14 (0xEE)	17 (0xF1)	32	—	(4)
400	001b	8 (0xE8)	19 (0xF3)	32	—	(4)

Table 25.8 Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 1 and NFE = 1

Transfer rate (kbps)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	PCLKB (MHz)	NF[1:0]	Computation expression
100	011b	12 (0xEC)	15 (0xEF)	32	01b	(5)
400	001b	6 (0xE6)	17 (0xF1)	32	01b	(5)

25.2.17 ICDRT : I2C Bus Transmit Data Register

Base address: IIC0 = 0x4005_3000

Offset address: 0x12

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

When ICDRT detects a space in the I2C Bus Shift Register (ICDRS), it transfers the transmit data that was written to ICDRT to ICDRS and starts transmitting data in transmit mode. The double-buffer structure of ICDRT and ICDRS allows continuous transmit operation if the next transmit data is written to ICDRT while the ICDRS data is being transmitted.

ICDRT can always be read and written to. Write transmit data to ICDRT once when a transmit data empty interrupt (IIC0_TXI) request is generated.

25.2.18 ICDRR : I2C Bus Receive Data Register

Base address: IIC0 = 0x4005_3000

Offset address: 0x13

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

When 1 byte of data is received, the received data is transferred from the I2C Bus Shift Register (ICDRS) to ICDRR to enable the next data to be received. The double-buffer structure of ICDRS and ICDRR allows continuous receive operation if the received data is read from ICDRR while ICDRS is receiving data. ICDRR cannot be written to. Read data from ICDRR once when a receive data full interrupt (IIC0_RXI) request is generated.

If ICDRR receives the next receive data before the current data is read from ICDRR (while the RDRF flag in ICSR2 is 1), the IIC automatically holds the SCL clock low 1 cycle before the RDRF flag is set to 1 next.

25.2.19 ICDRS : I²C Bus Shift Register

Base address: n/a

Offset address: n/a

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

ICDRS is an 8-bit shift register for data transmit and receive. During transmission, transmit data is transferred from ICDRT to ICDRS and is transmitted from the SDA0 pin. During reception, data is transferred from ICDRS to ICDRR after 1 byte of data is received. ICDRS cannot be accessed directly.

25.3 Operation

25.3.1 Communication Data Format

The I²C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a start or restart condition is an address frame that specifies a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is issued.

Figure 25.3 shows the I²C bus format, and Figure 25.4 shows the I²C bus timing.

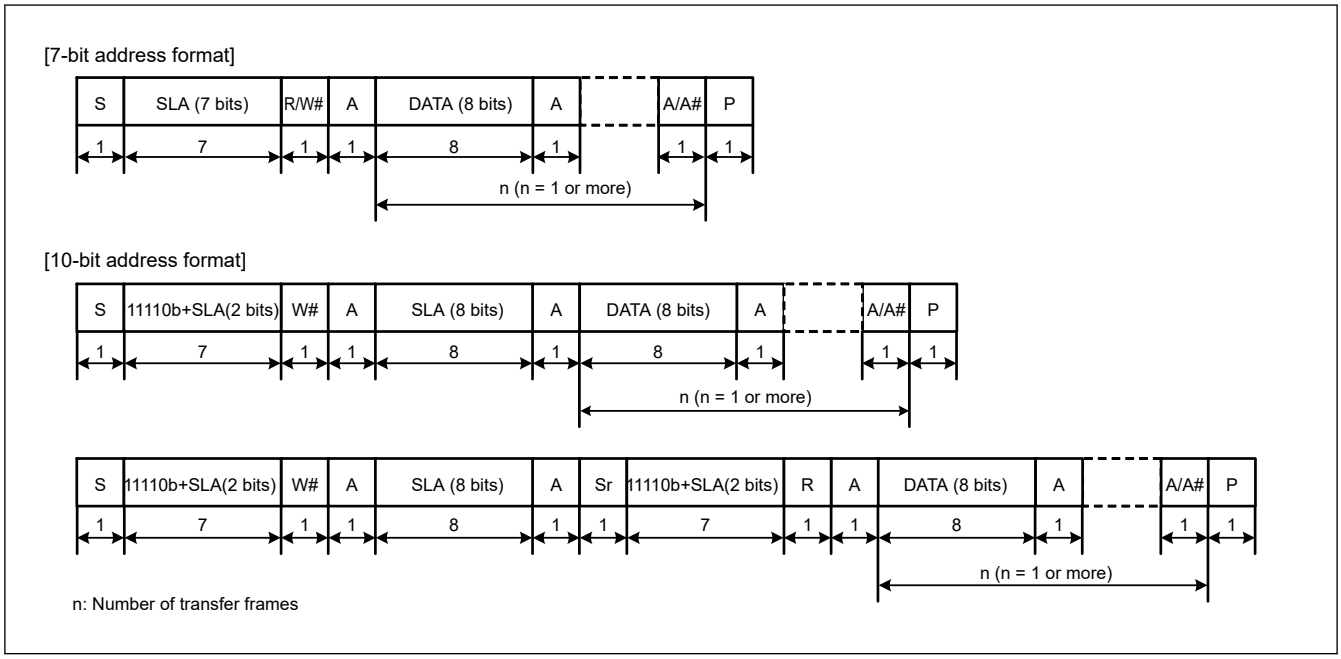


Figure 25.3 I²C bus format

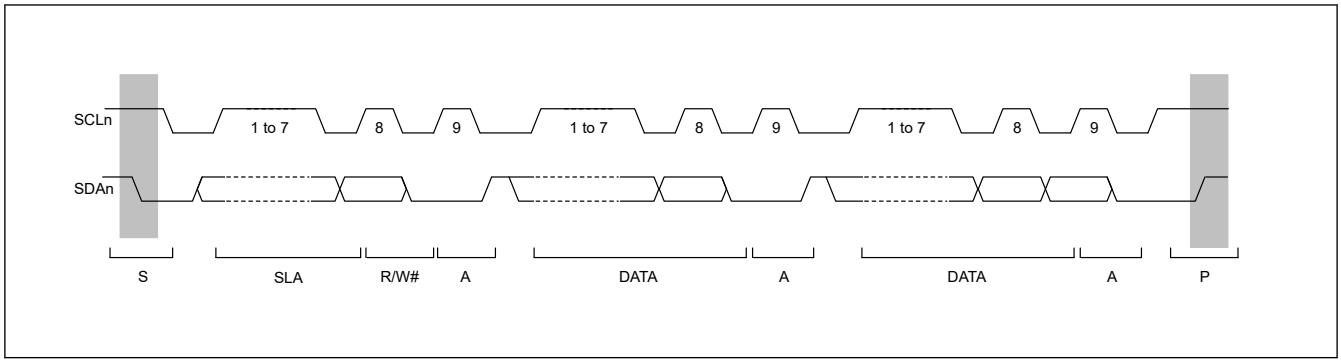


Figure 25.4 I²C bus timing when the SLA setting = 7 bits

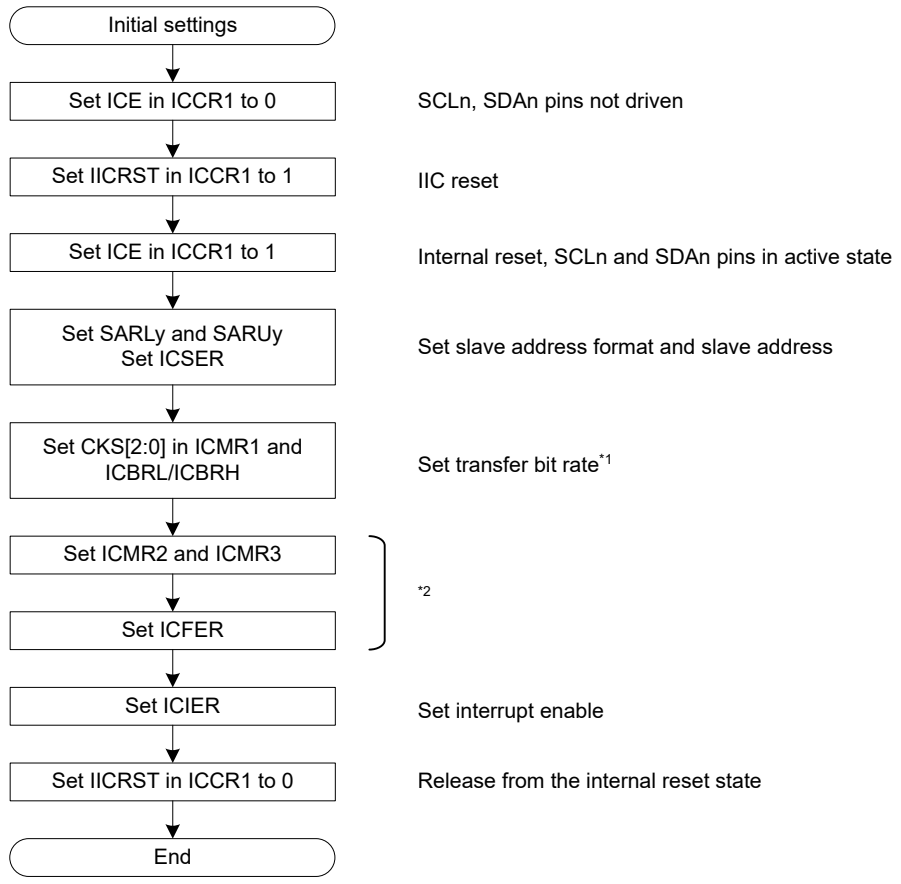
S:	Start condition. The master device drives the SDAn line low from high while the SCLn line is high.
SLA:	Slave address, by which the master device selects a slave device.
R/W#:	Indicates the direction of data transfer: from the slave device to the master device when R/W# is 1, or from the master device to the slave device when R/W# is 0.
A:	Acknowledge. The receive device drives the SDAn line low. In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.
A#:	Not Acknowledge. The receive device drives the SDAn line high.
Sr:	Restart condition. The master device drives the SDAn line low from the high level after the setup time has elapsed with the SCLn line high.
DATA:	Transmitted or received data.
P:	Stop condition. The master device drives the SDAn line high from low while the SCLn line is high.

25.3.2 Initial Settings

Before starting data transmission or reception, initialize the IIC using the procedure shown in Figure 25.5.

1. Set the ICCR1.ICE bit to 0 to set the SCLn and SDAn pins to the inactive state.
2. Set the ICCR1.IICRST bit to 1 to initiate IIC reset.
3. Set the ICCR1.ICE bit to 1 to initiate internal reset.
4. Set the SARLy, SARUy, ICSER, ICMR1, ICBRH, and ICBRL registers (y = 0 to 2), and set the other registers as required. For initial settings of the IIC, see Figure 25.5.
5. When the required register settings are complete, set the ICCR1.IICRST bit to 0 to release the IIC reset.

This procedure is not required if the IIC initialization is already complete.



Note: n = 0
Note 1. When the IIC is used only in slave mode, set the ICBRL register to a value longer than the data setup time.
Note 2. Set these registers as required.

Figure 25.5 Example IIC initialization flow

25.3.3 Master Transmit Operation

In master transmit operation, the IIC outputs the SCL clock and transmitted data signals as the master device, and the slave device returns acknowledgments. Figure 25.6 shows an example of master transmission, and Figure 25.7 to Figure 25.9 show the operation timing in master transmission.

To set up and perform master transmission:

1. Process initial settings. For details, see section 25.3.2. Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition request). On receiving the request, the IIC issues a start condition. At the same time, the BBSY and START flags in ICSR2 automatically set to 1 and the ST bit automatically is set to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that issuance of the start condition as requested by the ST bit is successfully complete, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 also automatically is set to 1 in response to the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag automatically is set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again is set to 1. After the byte containing the slave address and R/W# bit is transmitted, the value of the TRS bit automatically updates to select master transmit or master receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the IIC continues in master transmit mode.

Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
For data transmission with an address in the 10-bit format, start by writing 11110b, the 2 higher-order bits of the slave address, and W to ICDRT as the first address transmission. For the second address transmission, write the 8 lower-order bits of the slave address to ICDRT.

4. Check that the TDRE flag in ICSR2 is 1, and then write the transmit data to the ICDRT register. The IIC automatically holds the SCLn line low until the transmit data is ready or a stop condition is issued.
5. After all bytes of transmit data are written to the ICDRT register, wait until the value in the TEND flag in ICSR2 returns to 1, and then set the SP bit in ICCR2 to 1 (stop condition requested). On receiving a stop condition request, the IIC issues the stop condition. Regarding issuing a stop condition, see section 25.11.3. Issuing a Stop Condition.
6. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Additionally, it automatically sets the TDRE and TEND flags to 0, and sets the STOP flag in ICSR2 to 1.
7. Check that the ICSR2.STOP flag is 1, and then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

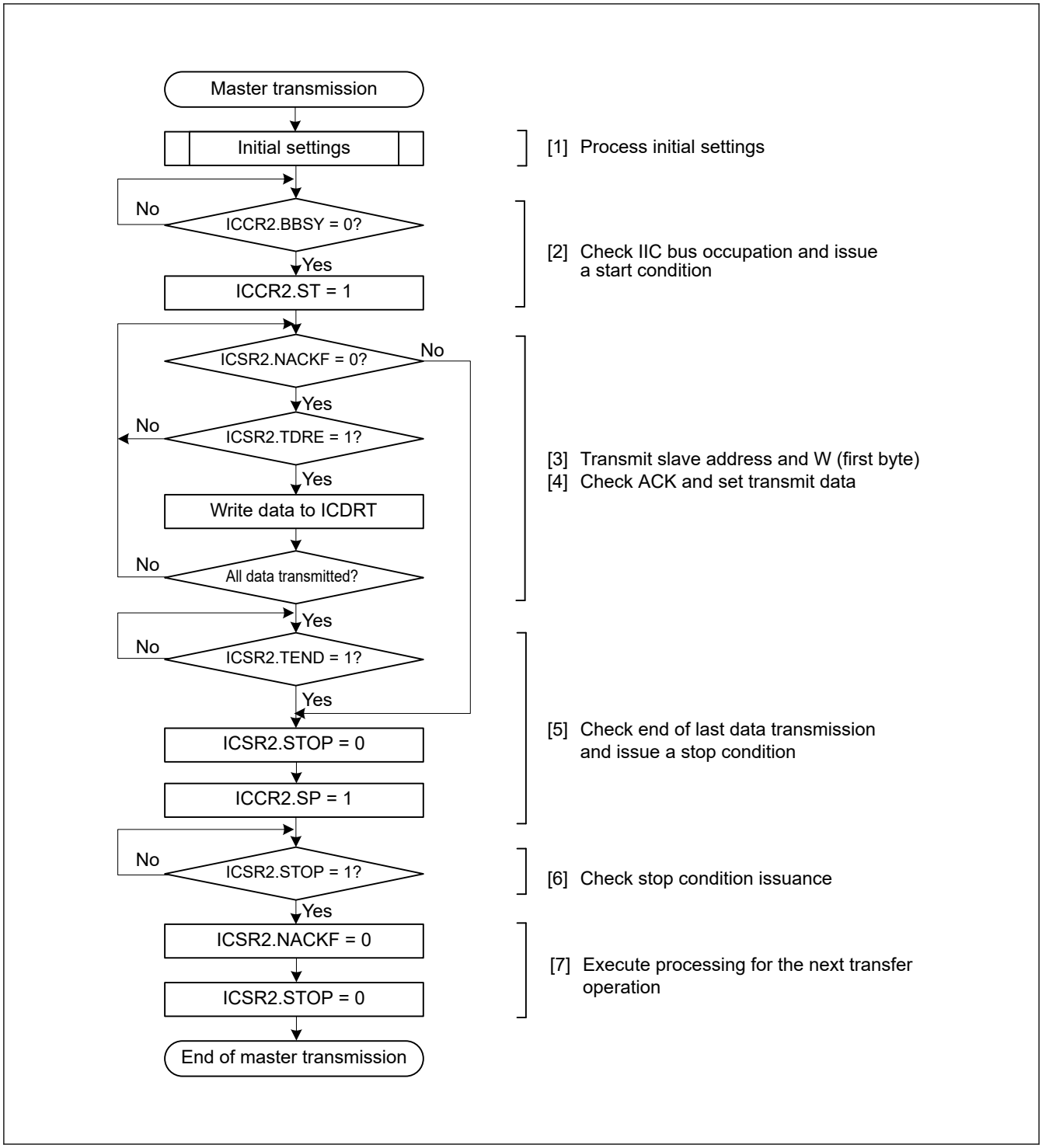


Figure 25.6 Example master transmission flow

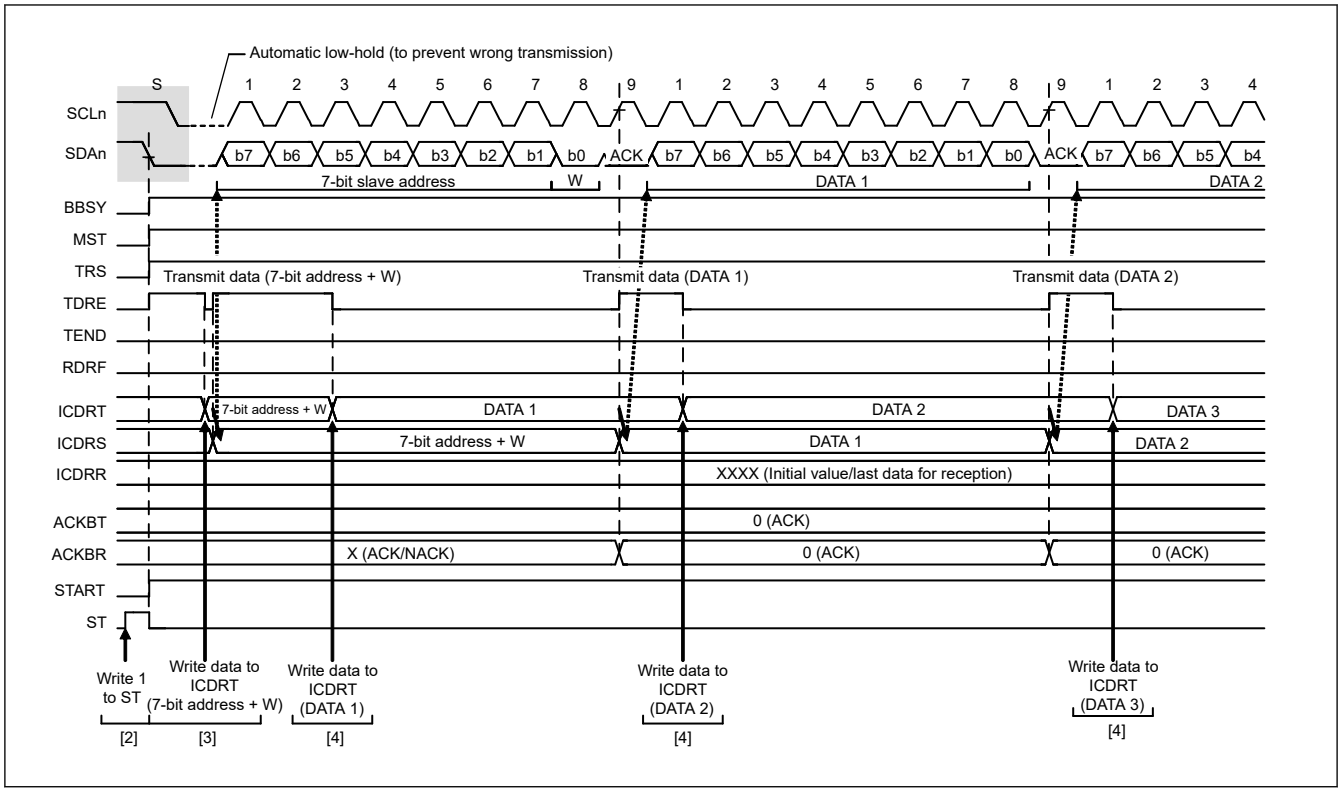


Figure 25.7 Master transmit operation timing (1) with 7-bit address format

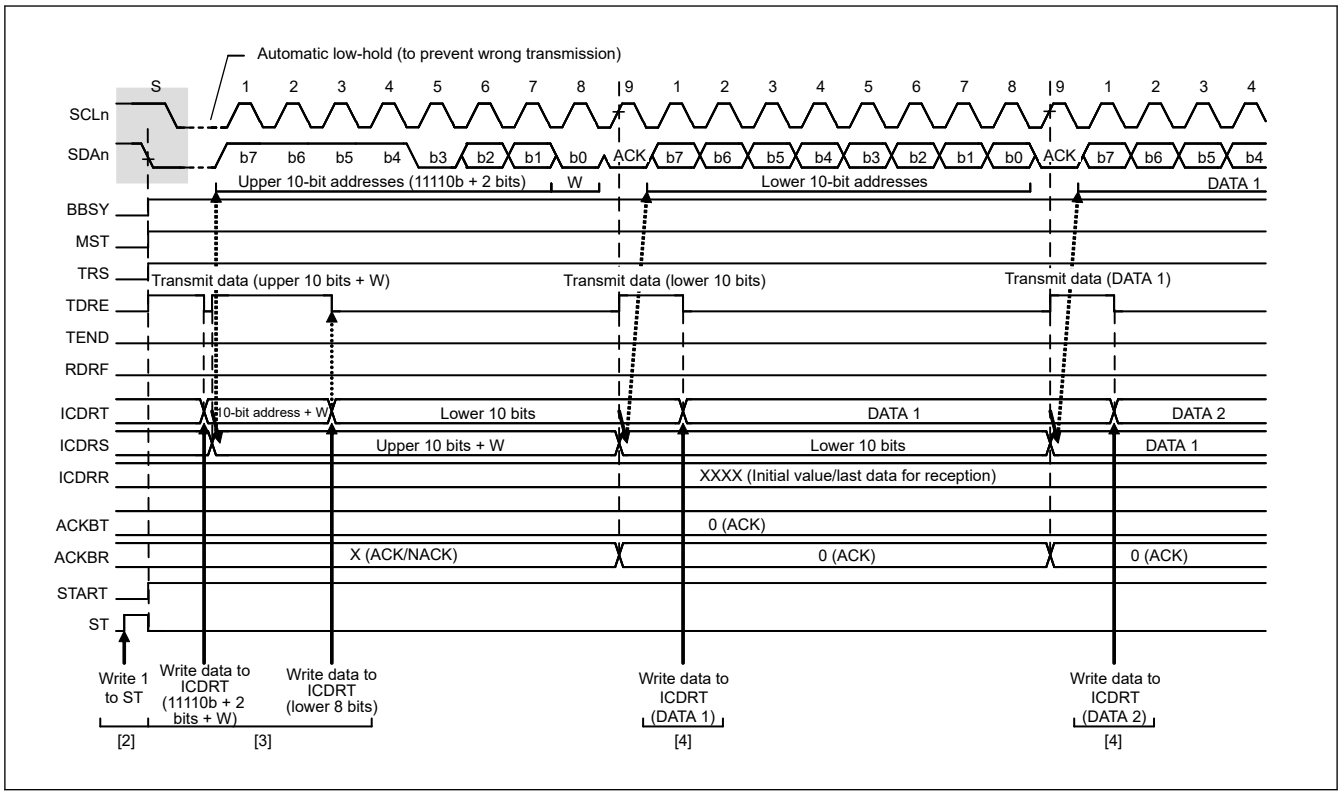


Figure 25.8 Master transmit operation timing (2) with 10-bit address format

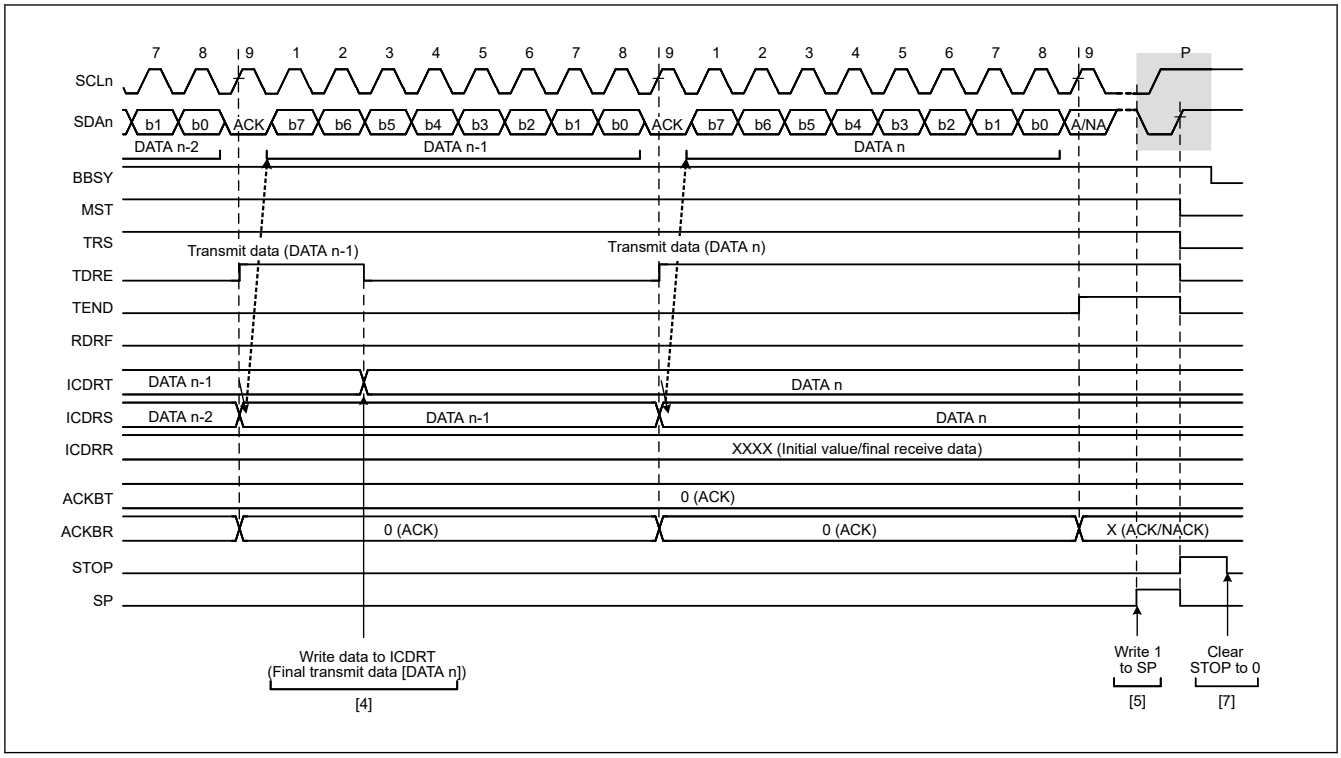


Figure 25.9 Master transmit operation timing (3)

25.3.4 Master Receive Operation

In master receive operation, the IIC as a master device outputs the SCL clock, receives data from the slave device, and returns acknowledgments. Because the IIC must start by sending a slave address to the associated slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 25.10 and Figure 25.11 show examples of master reception (7-bit address format), and Figure 25.12 to Figure 25.14 show the operation timing in master reception.

To set up and perform master reception:

1. Process initial settings. For details, see section 25.3.2. Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition request). On receiving the request, the IIC issues a start condition. When the IIC detects the start condition, the BBSY and START flags in ICSR2 automatically set to 1, and the ST bit automatically is set to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that issuance of the start condition as requested by the ST bit is successfully complete, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 also automatically is set to 1 in response to the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag automatically is set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again is set to 1. When the byte containing the slave address and R/W# bit is transmitted, the value of the ICCR2.TRS bit automatically updates to select transmit or receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit is 1, the TRS bit is set to 0 on the rising edge of the ninth cycle of the SCL clock, placing the IIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag automatically is set to 1.
   Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
   For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a restart condition. After that, transmitting 11110b, the two higher-order bits of the slave address, and the R bit places the IIC in master receive mode.
4. Dummy read ICDRR after confirming that the RDRF flag in ICSR2 is 1. This makes the IIC start output of the SCL clock and start data reception.

5. After 1 byte of data is received, the RDRF flag in ICSR2 is set to 1 on the rising edge of the 8th or 9th cycle of the SCL clock, as selected in the RDRFS bit in ICMR3. Reading ICDRR at this time produces the received data, and the RDRF flag is automatically set to 0 at the same time. Additionally, the value of the acknowledgment field received during the ninth cycle of the SCL clock is returned as the value set in the ICMR3.ACKBT bit. If the next byte to be received is the second-to-last byte, set the ICMR3.WAIT bit to 1 for wait insertion before reading ICDRR, containing the second-to-last byte. In addition to enabling NACK output, even when interrupts or other operations result in delays in setting the ICMR3.ACKBT bit to 1 (NACK) in step (6), this fixes the SCLn line to the low level on the rising edge of the ninth clock cycle in reception of the last byte, which enables the issuing of a stop condition.
6. When the ICMR3.RDRFS bit is 0, and the slave device must be notified that it is to end transfer for data reception after transfer of the next and final byte, set the ICMR3.ACKBT bit to 1 (NACK).
7. After reading the second-to-last byte from the ICDRR register, if the value of the ICSR2.RDRF flag is 1, write 1 to the SP bit in ICCR2 (stop condition requested), and then read the last byte from ICDRR. When ICDRR is read, the IIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is complete or the SCLn line is released from the low-hold state.
8. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Additionally, detection of the stop condition sets the ICSR2.STOP flag to 1.
9. Check that the ICSR2.STOP flag is 1, then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

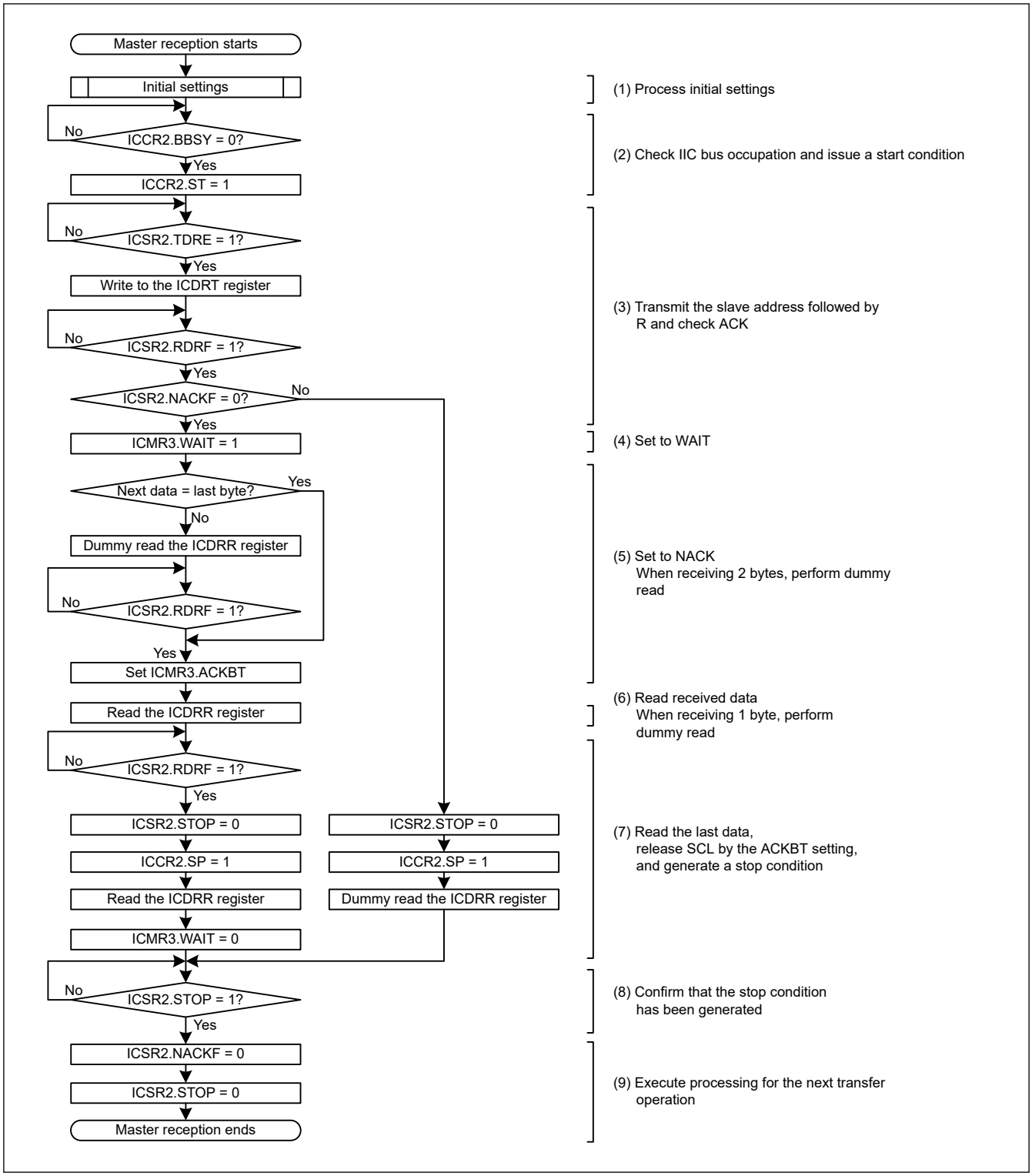


Figure 25.10 Example master reception flow with 7-bit address format of 1 byte or 2 bytes

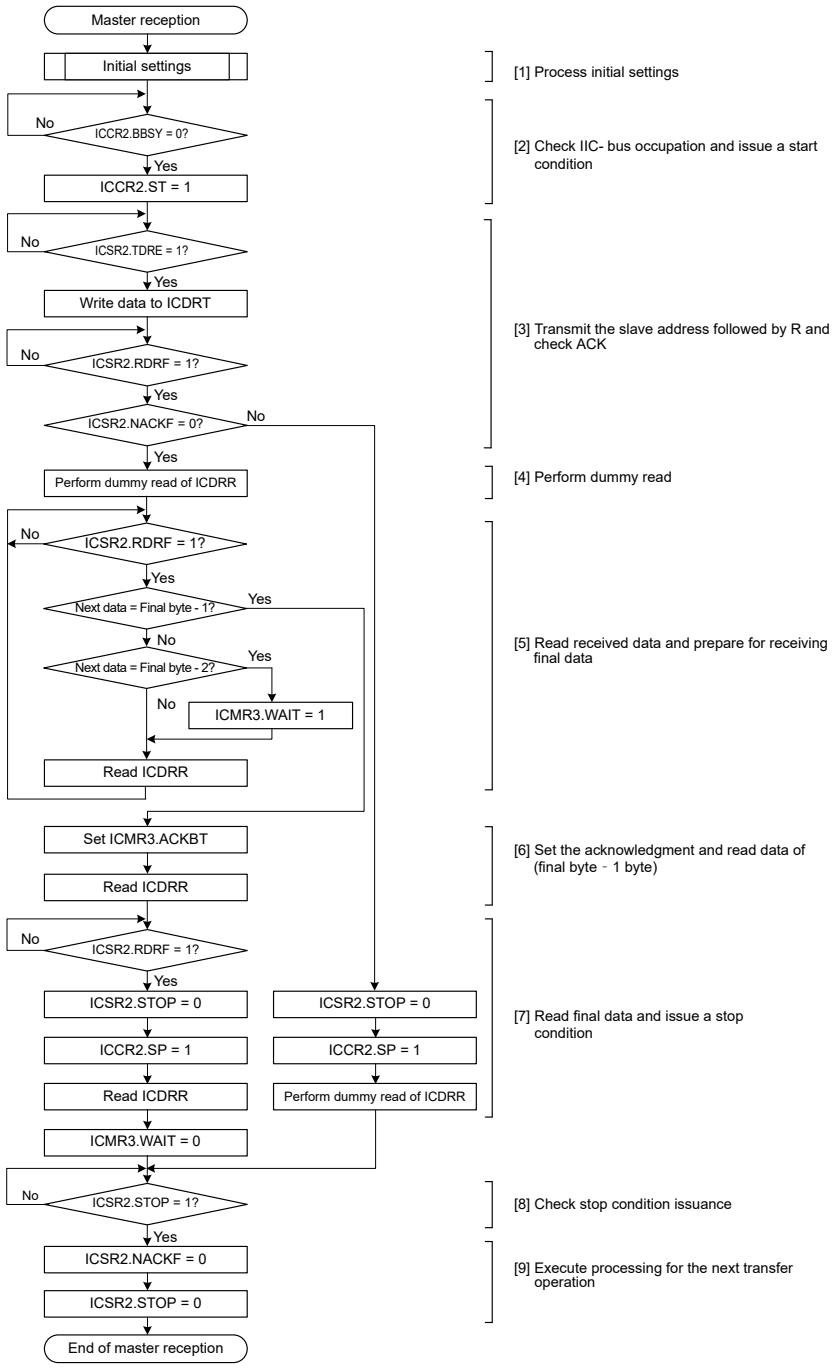


Figure 25.11 Example master reception flow with 7-bit address format of 3 or more bytes

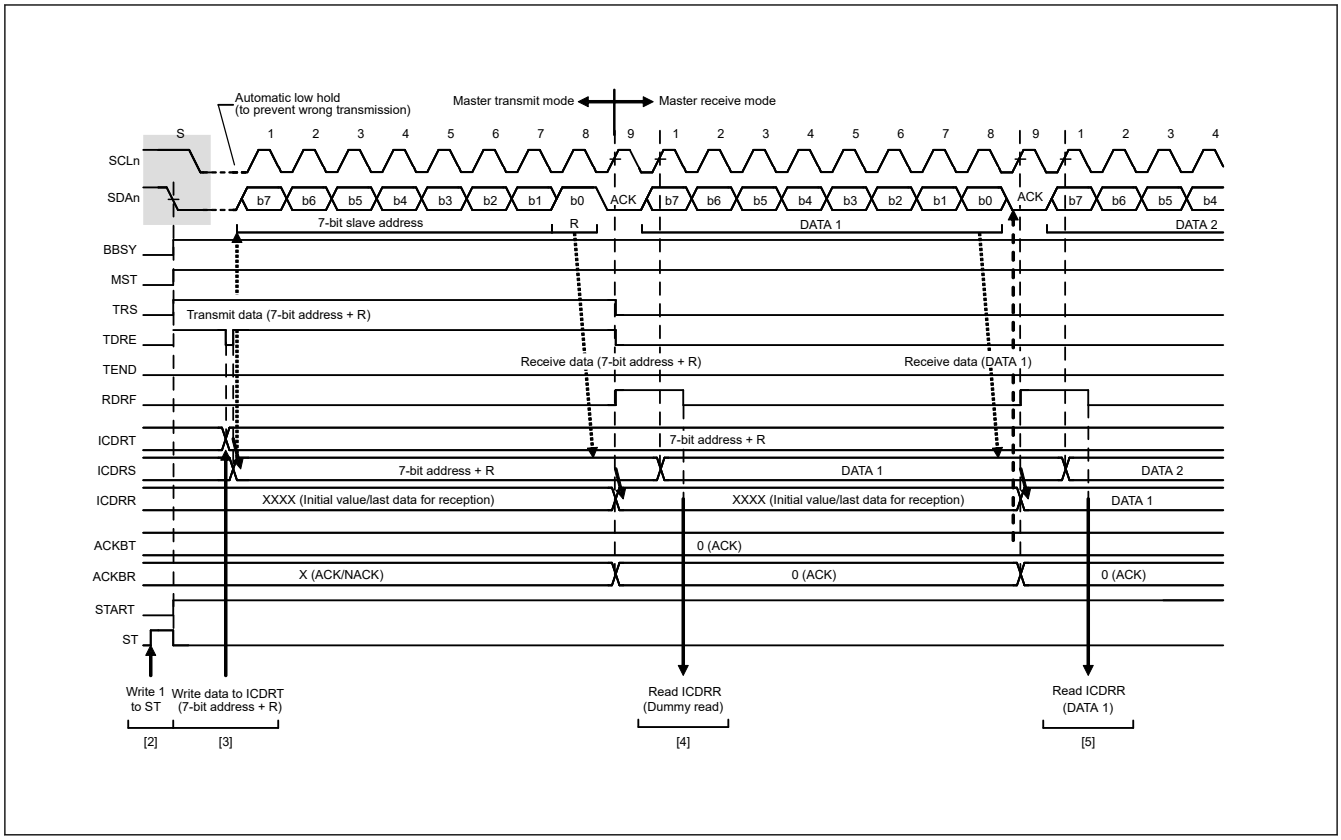


Figure 25.12 Master receive operation timing (1) with 7-bit address format when RDRFS = 0

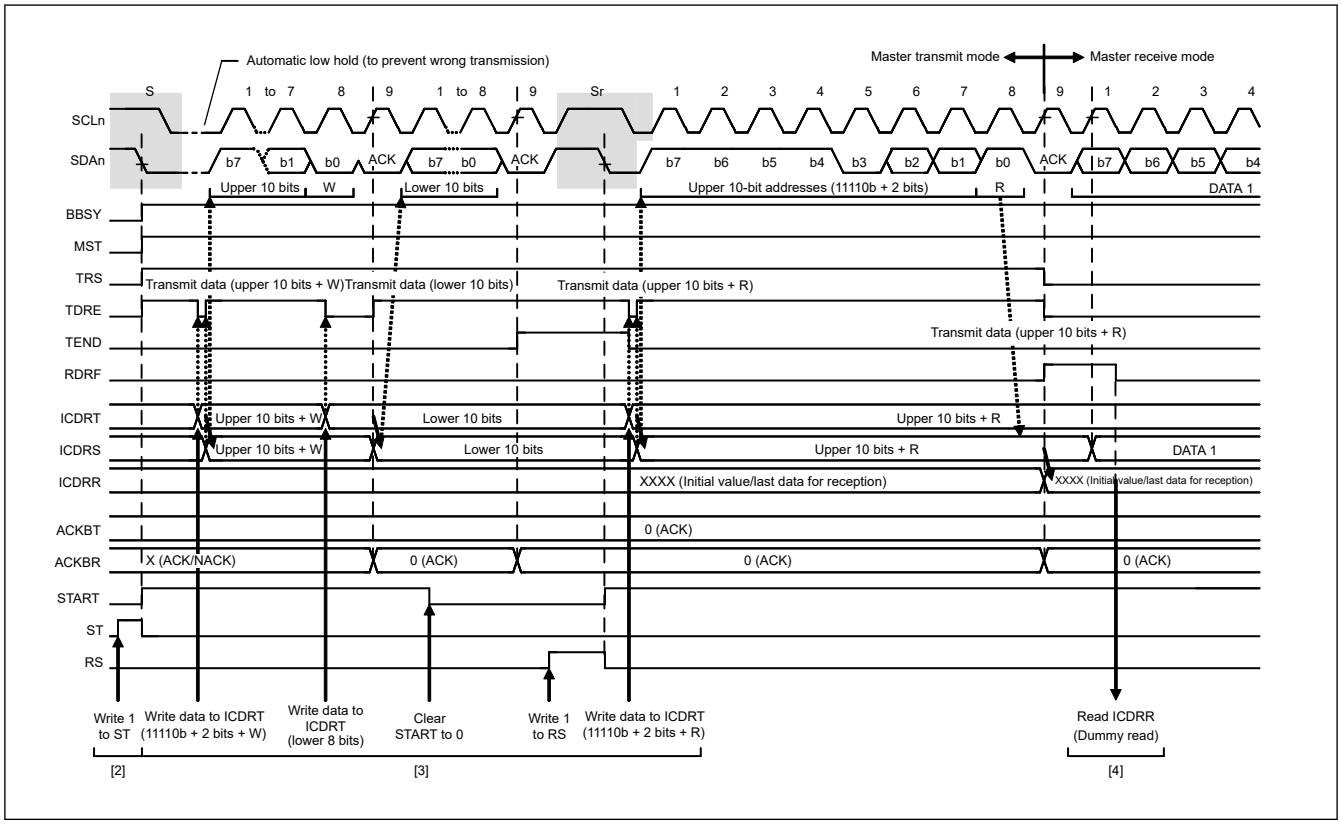


Figure 25.13 Master receive operation timing (2) with 10-bit address format when RDRFS = 0

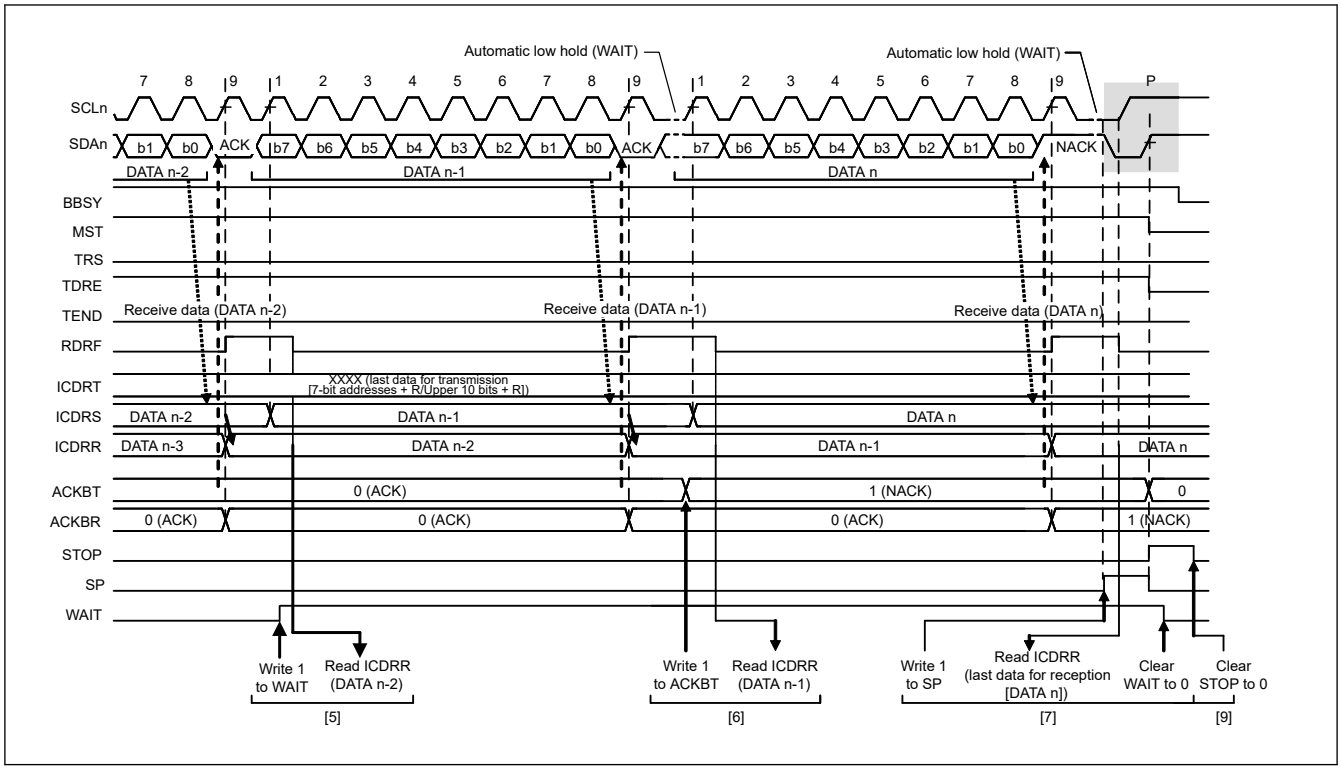


Figure 25.14 Master receive operation timing (3) when RDRFS = 0

25.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL clock, the IIC transmits data as a slave device, and the master device returns acknowledgments.

Figure 25.15 shows an example of slave transmission, and Figure 25.16 and Figure 25.17 show the operation timing in slave transmission.

To set up and perform slave transmission:

1. Initialize the IIC using the procedure in section 25.3.2. Initial Settings.
   After initialization, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 1 on the rising edge of the ninth cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of the SCL clock. If the value of the R/W# bit is 1, the IIC automatically places itself in slave transmit mode by setting both the ICCR2.TRS bit and the ICSR2.TDRE flag to 1.
3. Check that the ICSR2.TDRE flag is 1, then write the transmit data to the ICDRT register. If the IIC receives no acknowledge from the master device (receives an NACK signal) while the ICFER.NACKE bit is 1, the IIC suspends transfer of the next data.
4. Wait until the ICSR2.TEND flag is set to 1 while the ICSR2.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDRT register. When the ICSR2.NACKF flag or the TEND flag is 1, the IIC drives the SCLn line low on the ninth falling edge of the SCL clock.
5. When the ICSR2.NACKF or ICSR2.TEND flag is 1, dummy read ICDRR to complete the processing. This releases the SCLn line.
6. On detecting the stop condition, the IIC automatically sets the ICSR1.HOA, GCA, and AASy flags (y = 0 to 2), the ICSR2.TDRE and TEND flags, and the ICCR2.TRS bit to 0, and enters slave receive mode.
7. Check that the ICSR2.STOP flag is 1, then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

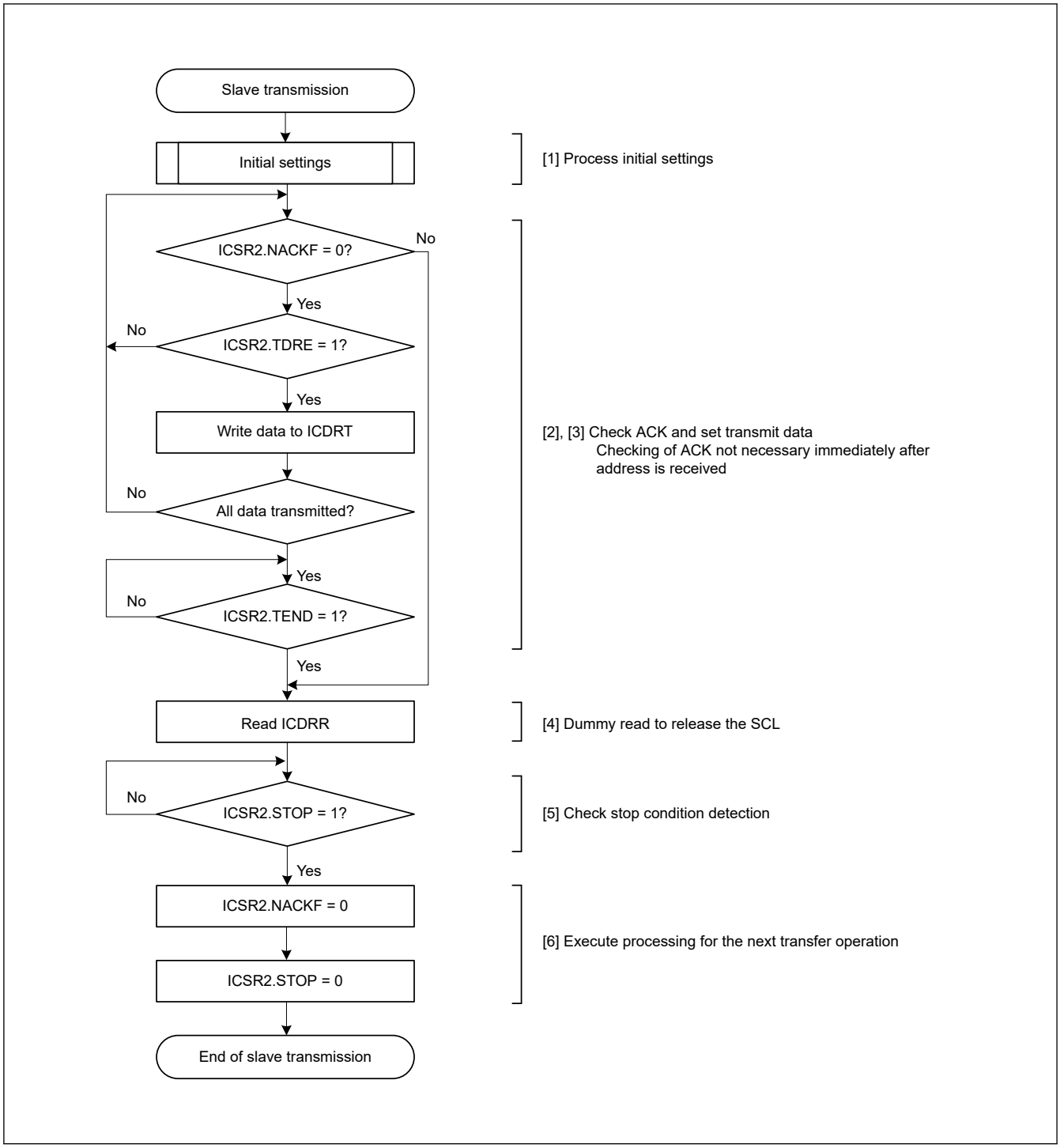


Figure 25.15 Example slave transmission flow

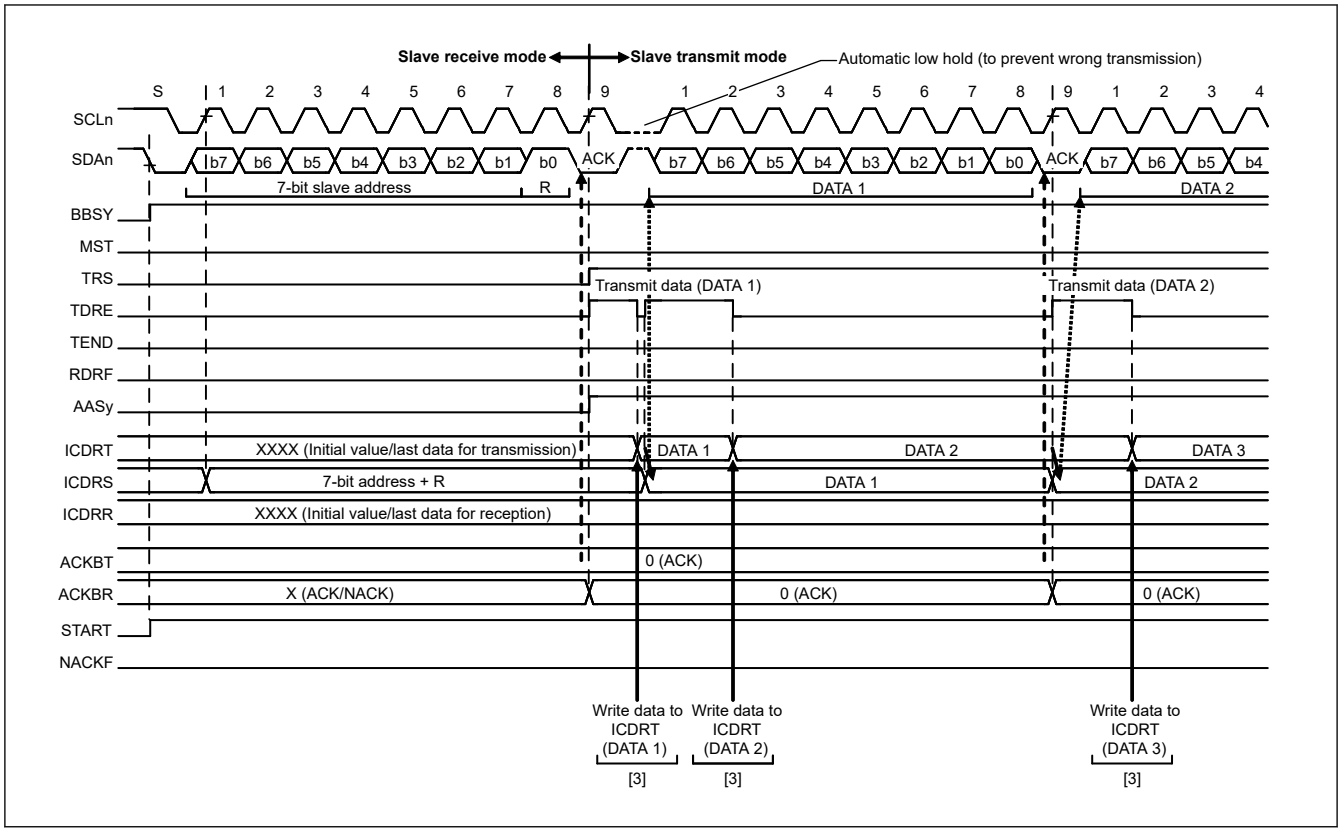


Figure 25.16 Slave transmit operation timing (1) with 7-bit address format

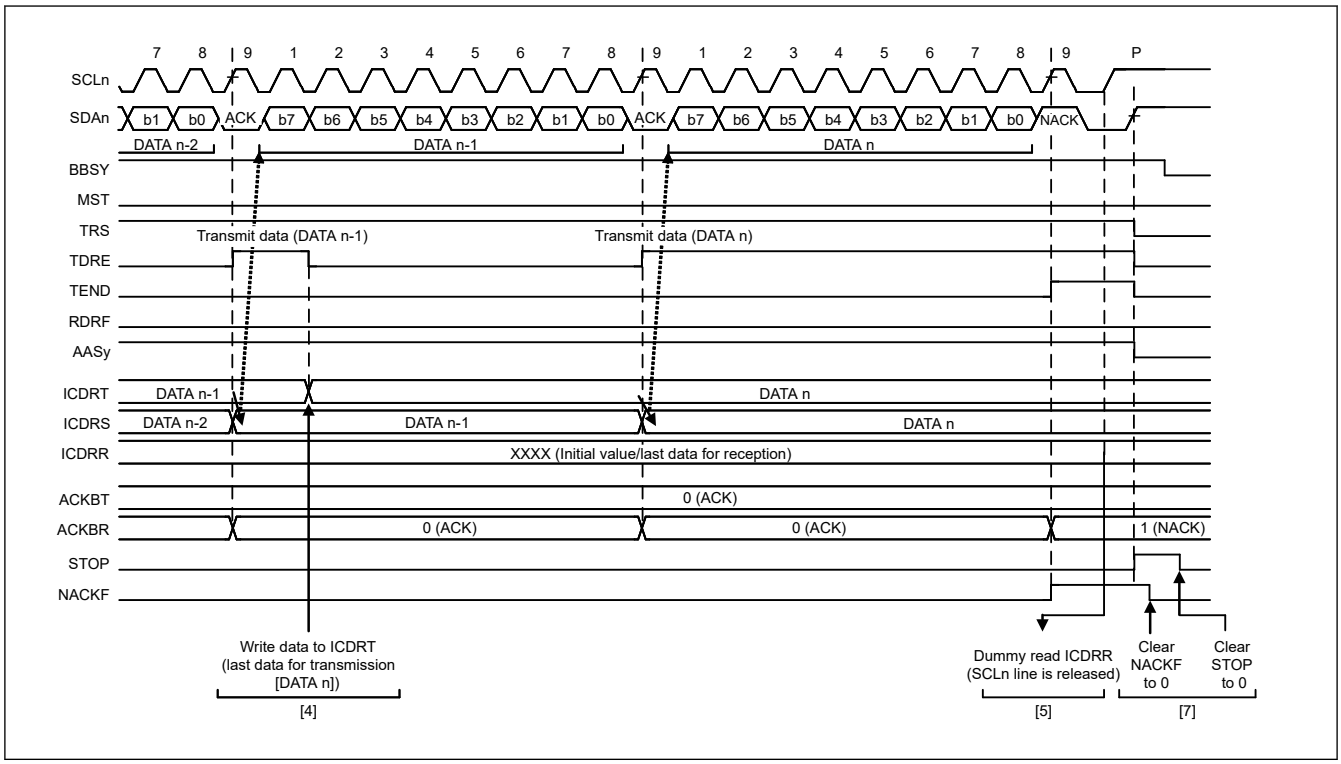


Figure 25.17 Slave transmit operation timing (2)

25.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the IIC returns acknowledgments as a slave device.

Figure 25.18 shows an example of slave reception, and Figure 25.19 and Figure 25.20 show the operation timing in slave reception.

To set up and perform slave reception:

1. Initialize the IIC using the procedure in section 25.3.2. Initial Settings.
   After initialization, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 1 on the rising edge of the ninth cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of the SCL clock. If the value of the R/W# bit is 0, the IIC continues to place itself in slave receive mode and sets the RDRF flag in ICSR2 to 1.
3. Check that the ICSR2.STOP flag is 0 and the ICSR2.RDRF flag is 1, then dummy read ICDRR. The dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected.
4. When ICDRR is read, the IIC automatically sets the ICSR2.RDRF flag to 0. If reading of ICDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the IIC holds the SCLn line low until 1 SCL cycle before the point where RDRF must be set. In this case, reading ICDRR releases the SCLn line from being held low.
   When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1, read ICDRR until all the data is completely received.
5. On detecting the stop condition, the IIC automatically clears the ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 0.
6. Check that the ICSR2.STOP flag is 1, then set the ICSR2.STOP flag to 0 for the next transfer operation.

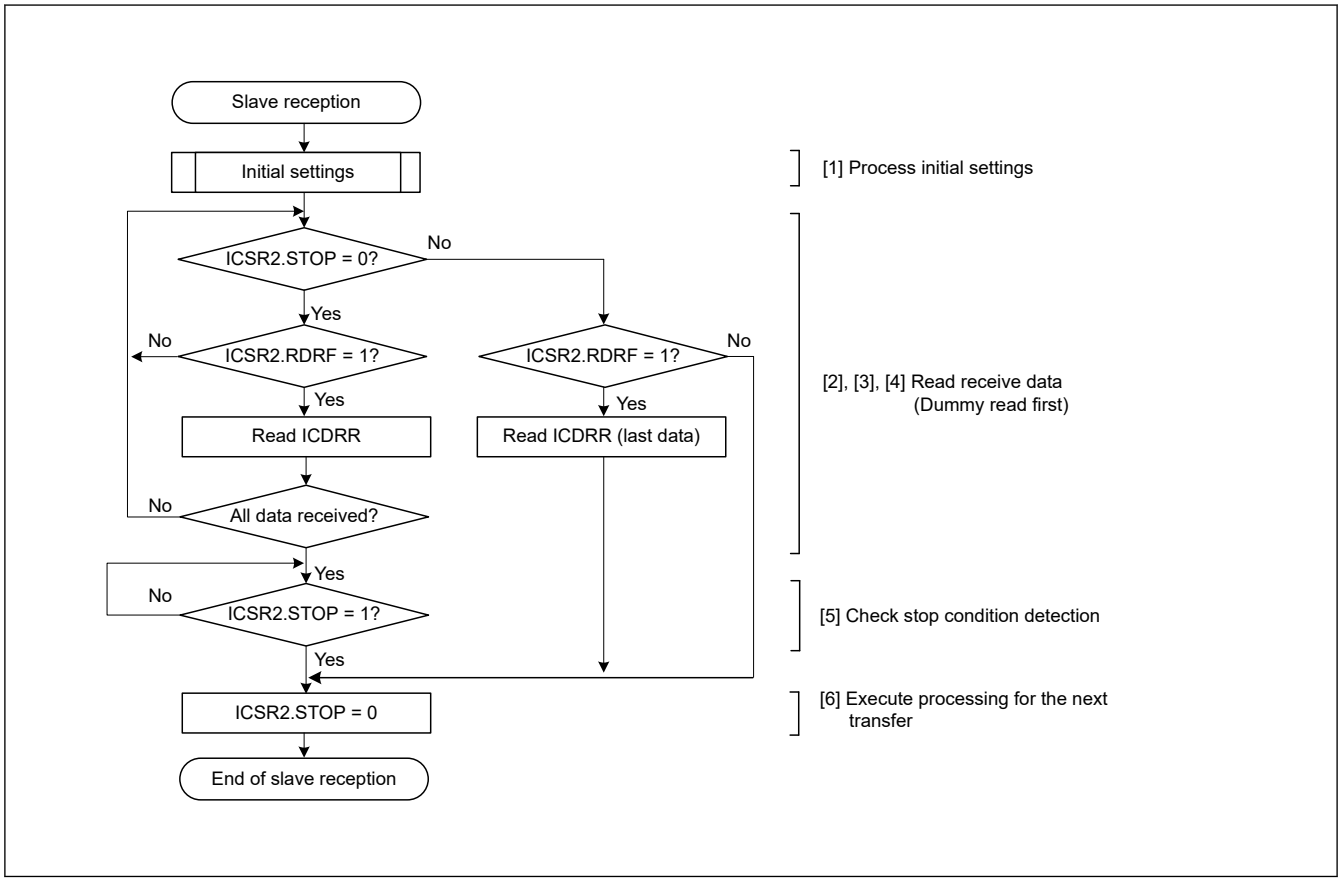


Figure 25.18 Example slave reception flow

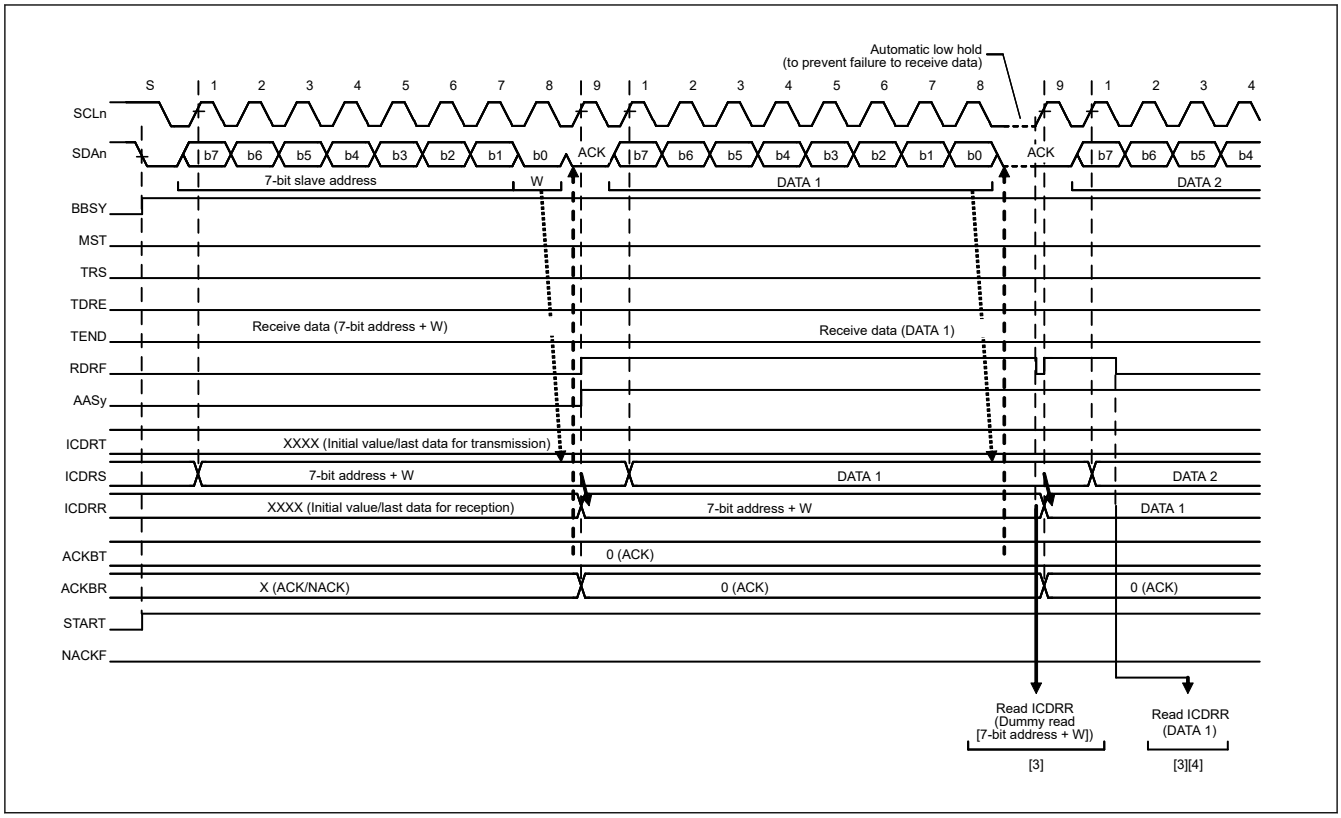


Figure 25.19 Slave receive operation timing (1) with 7-bit address format when RDRFS = 0

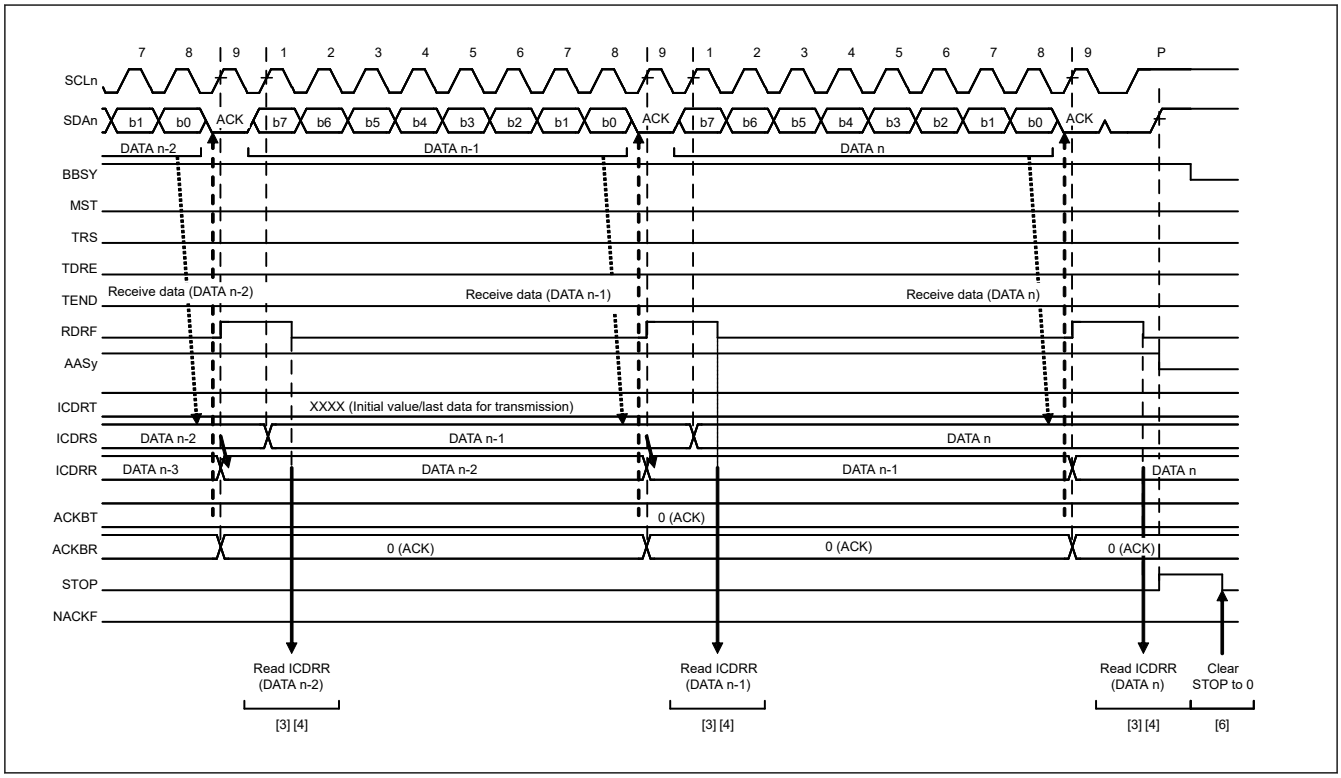


Figure 25.20 Slave receive operation timing (2) when RDRFS = 0

25.4 SCL Synchronization Circuit

For generation of the SCL clock, the IIC starts counting the value for the high-level period specified in ICBRH when it detects a rising edge on the SCLn line, and it drives the SCLn line low when it completes counting. When the IIC detects

the falling edge of the SCLn line, it starts counting the value for the low-level period specified in ICBRL, and then it stops driving the SCLn line, releasing the line, when it completes counting. The IIC repeats this process to generate the SCL clock.

If multiple master devices are connected to the I^2C bus, a collision of SCL signals might arise because of contention with another master device. In such cases, the master devices must synchronize their SCL signals. Because this synchronization of SCL signals must be bit-by-bit, the IIC is equipped with an SCL synchronization circuit to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCLn line while in master mode.

When the IIC detects a rising edge on the SCLn line and starts counting the high-level period specified in ICBRH.BRH[4:0], and the level on the SCLn line falls because an SCL signal is being generated by another master device, the IIC performs the following:

1. Stops counting when it detects the falling edge.
2. Drives the level on the SCLn line low.
3. Starts counting the low-level period specified in ICBRL.BRL[4:0].

When the IIC finishes counting the low-level period, it stops driving the SCLn line low to release the line. If the low-level period of the SCL clock signal from the other master device is longer than the low-level period set in the IIC, the low-level period of the SCL signal is extended. When the low-level period for the other master device ends, the SCL signal rises because the SCLn line is released.

When the IIC finishes outputting the low-level period of the SCL clock, the SCLn line is released and the SCL clock rises. That is, when SCL signals from more than one master are contending, the high-level period of the SCL signal is synchronized with that of the clock with the narrower period, and the low-level period of the SCL signal is synchronized with that of the clock with the broader period. However, such synchronization of the SCL signal is only enabled when the SCLE bit in ICFER is set to 1.

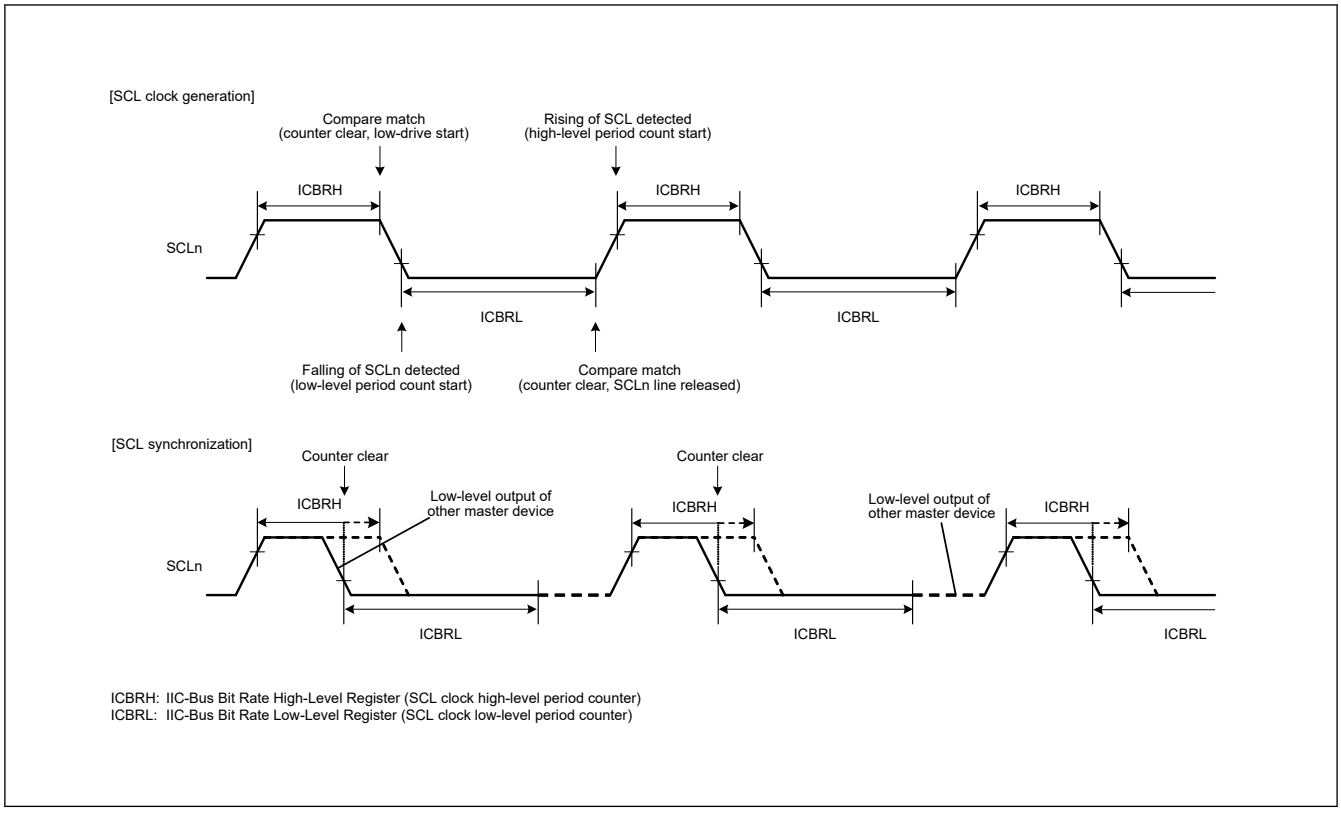


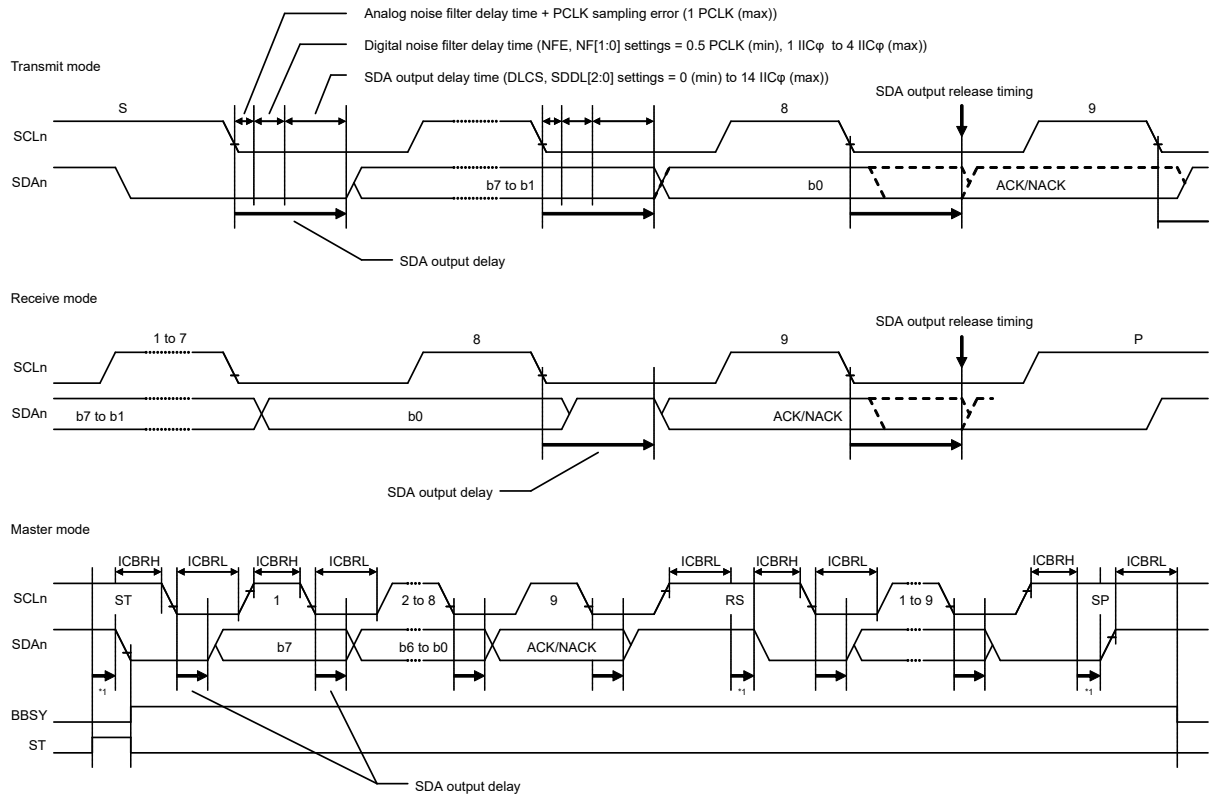
Figure 25.21 Generation and synchronization of SCL signal from IIC

25.5 SDA Output Delay Function

The IIC module incorporates a function for delaying output on the SDA line. The delay can be applied to all output on the SDA line, including issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals.

With this function, SDA output is delayed from the detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval during which the SCL clock is low. This approach helps prevent erroneous operation of communications devices, with the aim of satisfying the 300-ns minimum data-hold time requirement of the SMBus specification. The output delay function is enabled by setting the SDDL[2:0] bits in ICMR2 to a value other than 000b, and disabled by setting the same bits to 000b.

When the SDA output delay function is enabled, for example, the DLCS bit in ICMR2 selects the clock source for the SDA output delay counter, either as the internal base clock (IICφ) for the IIC module or as the internal base clock divided by 2 (IICφ/2). The counter counts the number of cycles set in the SDDL[2:0] bits in ICMR2. When the delay cycles count is reached, the IIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.



Note: PCLK = PCLKB
Note 1. The output delay function is set by the DLCS and SDDL[2:0] bits when a start (ST), restart (RS), or stop (SP) condition is issued.

Figure 25.22 SDA output delay function

25.6 Digital Noise Filter Circuits

The internal circuitry sees the states of the SCLn and SDAn pins through analog and digital noise-filter circuits. Figure 25.23 shows a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the IIC consists of four flip-flop circuit stages connected in series and a match-detection circuit. The number of valid stages in the digital noise filter is selected in the NF[1:0] bits in ICMR3. The selected number of valid stages determines the noise-filtering capability as a period from 1 to 4 IICφ cycles.

The input signal to the SCLn pin (or SDAn pin) is sampled on falling edges of the IICφ signal. When the input signal level matches the output level of the number of valid flip-flop circuit stages as selected in the NF[1:0] bits in ICMR3, the signal level is seen in the subsequent stage. If the signal levels do not match, the previous value is saved.

If the ratio between the frequency of the internal operating clock (PCLKB) and the transfer rate is small, for example, if data transfer is 400 kbps with PCLKB = 4 MHz, the characteristics of the digital noise filter might lead to the elimination of

required signals as noise. In such cases, it is possible to disable the digital noise-filter circuit by setting the ICFER.NFE bit to 0, and use only the analog noise filter circuit.

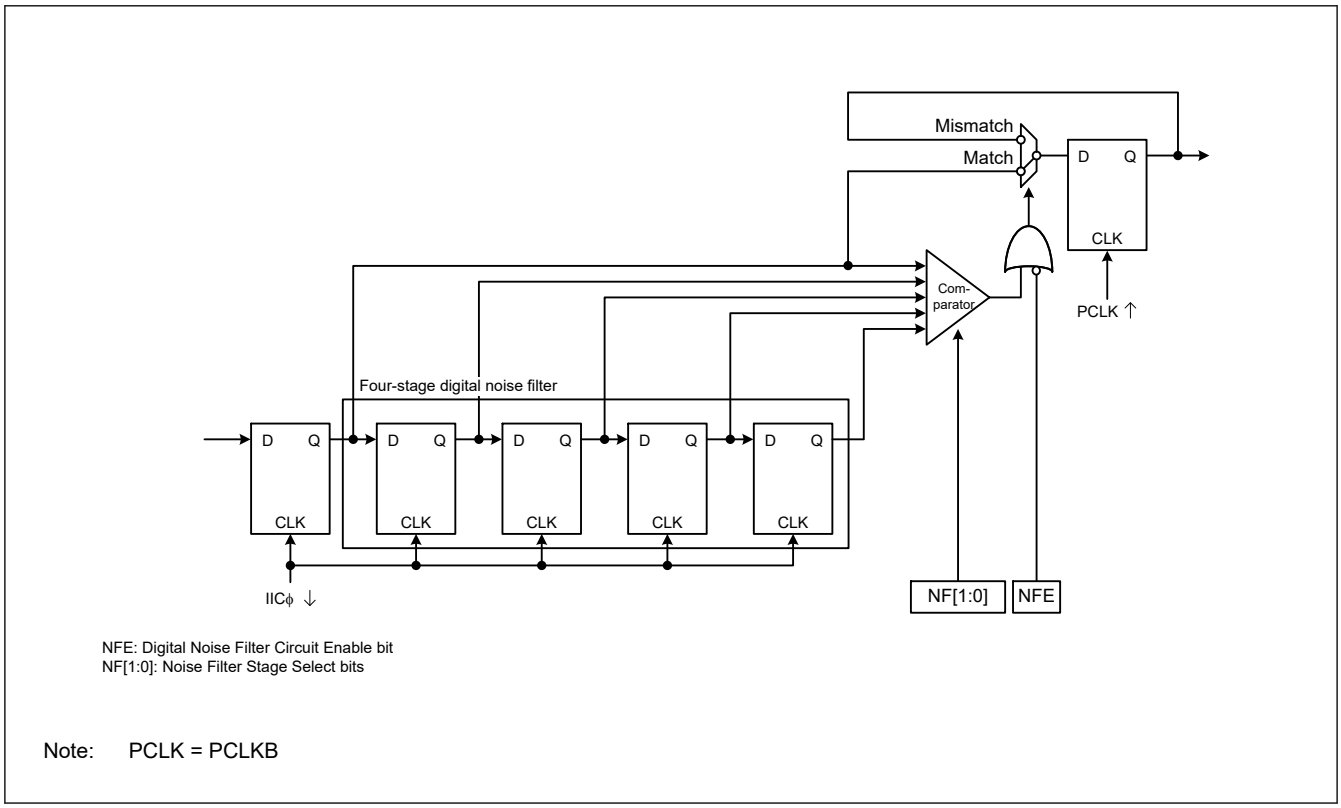


Figure 25.23 Digital noise filter circuit block diagram

25.7 Address Match Detection

The IIC can set three unique slave addresses in addition to the general call address and host address. The slave addresses can be 7-bit or 10-bit slave addresses.

25.7.1 Slave-Address Match Detection

The IIC can set three unique slave addresses and has a slave address detection function for each unique slave address. When the SARyE bit (y = 0 to 2) in ICSER is set to 1, the slave addresses set in SARUy and SARLy (y = 0 to 2) can be detected.

When the IIC detects a match of the set slave address, the associated AASy flag (y = 0 to 2) in ICSR1 is set to 1 on the rising edge of the ninth SCL clock cycle, and the RDRF flag in ICSR2 or the TDRE flag in ICSR2 is set to 1 by the subsequent R/W# bit. This causes a receive data full interrupt (IICn_RXI) or transmit data empty interrupt (IICn_TXI) to be generated. The AASy flag identifies which slave address is specified.

Figure 25.24 to Figure 25.26 show the AASy flag set timing in three cases.

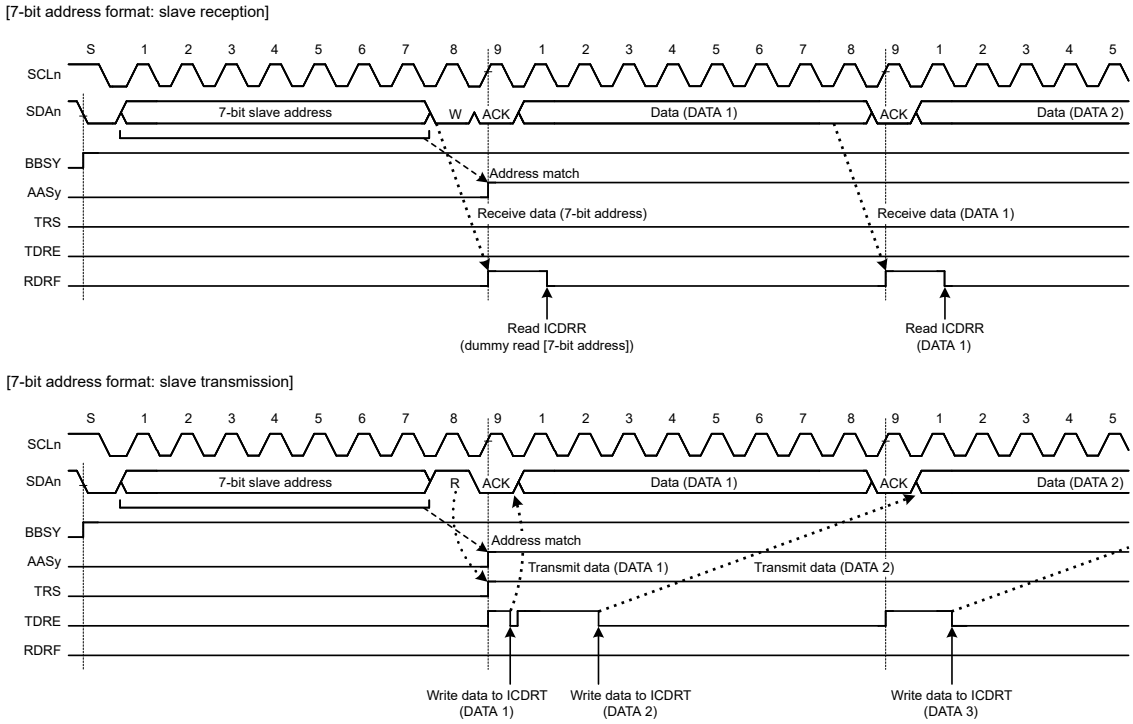


Figure 25.24 AASy flag set timing with 7-bit address format

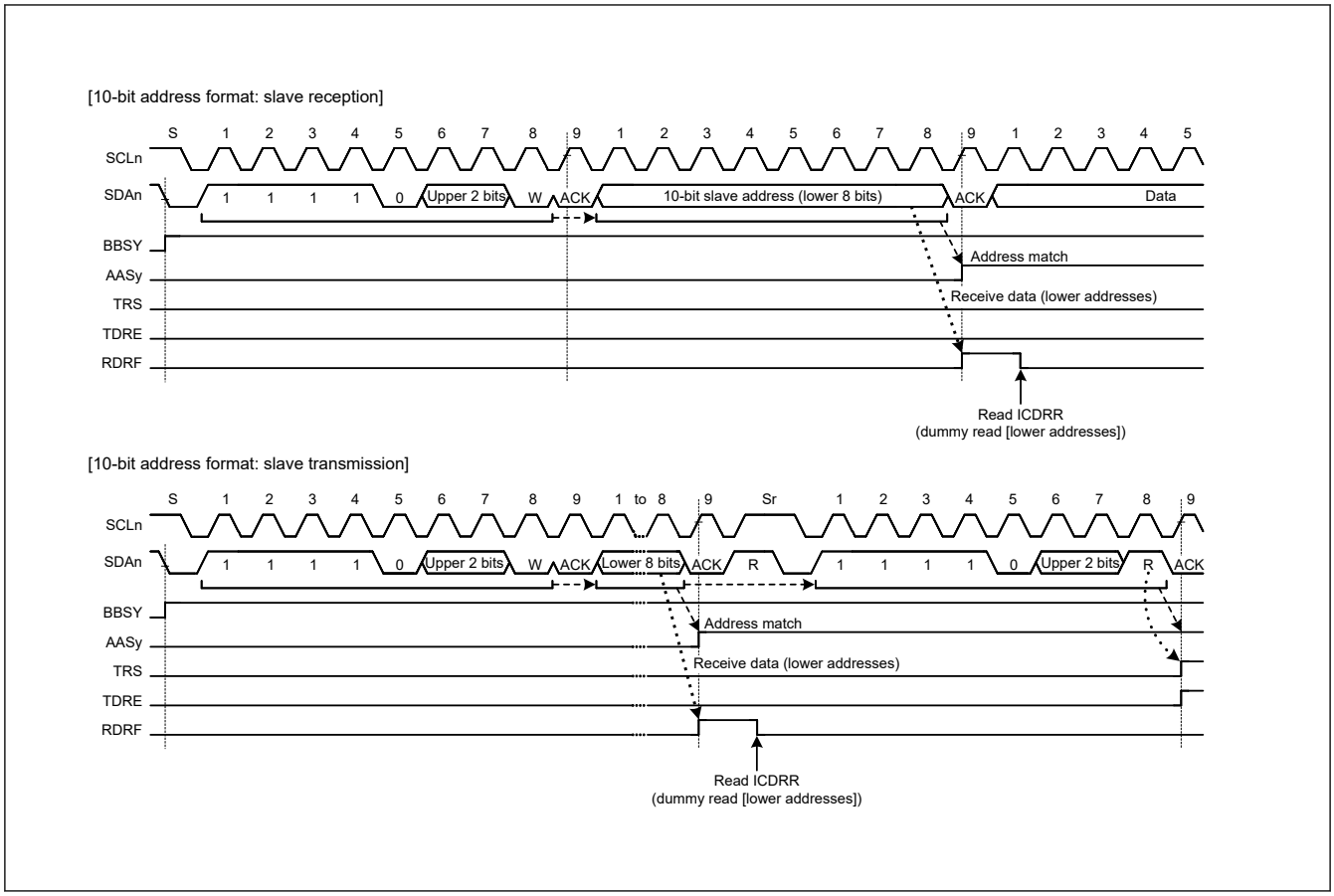


Figure 25.25 AASy flag set timing with 10-bit address format

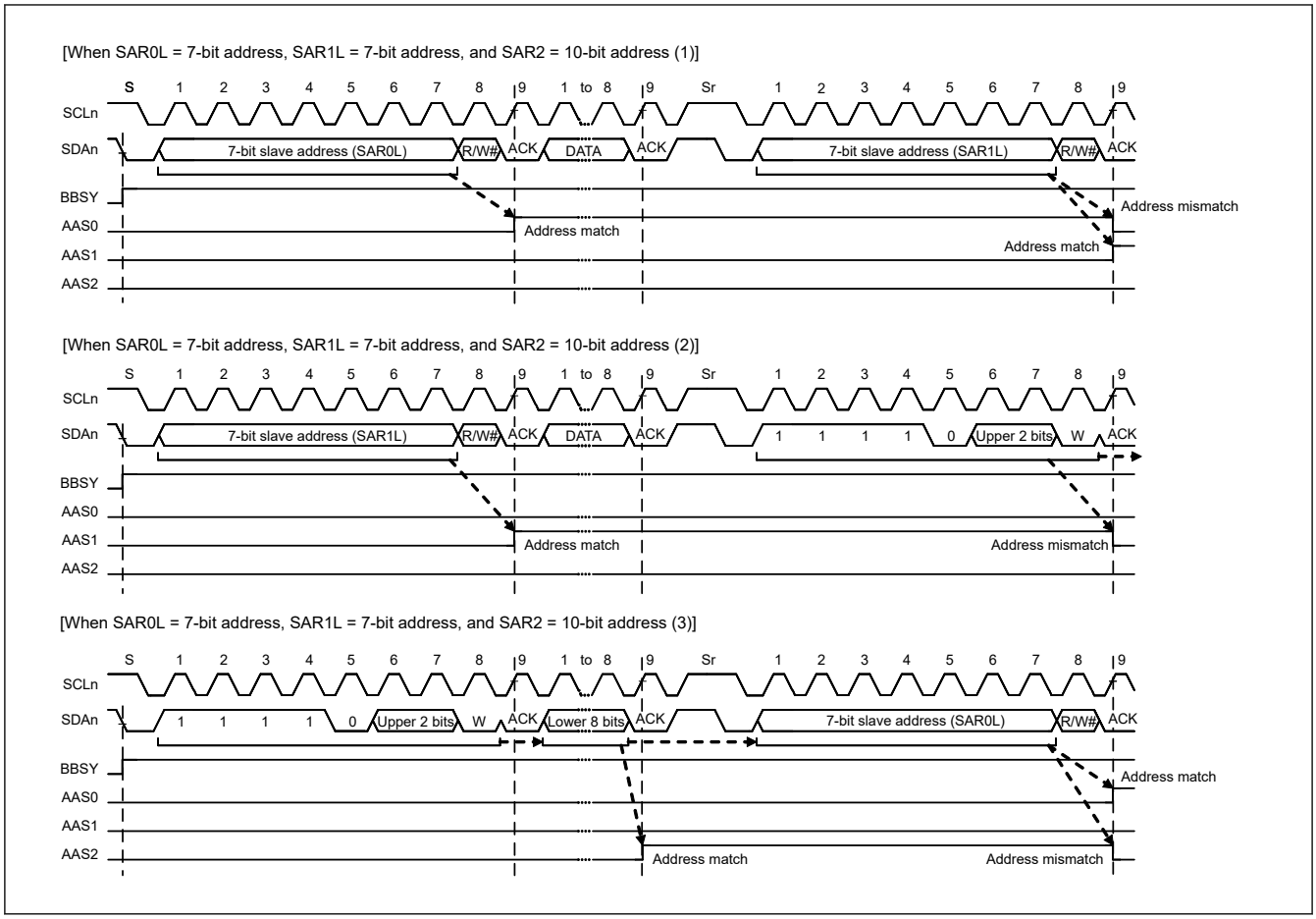


Figure 25.26 AASy flag set and clear timing with mixed 7-bit and 10-bit address formats

25.7.2 Detection of General Call Address

The IIC provides detection of the general call address (0000 000b + 0 [W]). This is enabled by setting the GCAE bit in ICSER to 1.

If the address received after a start or restart condition is issued is 0000 000b + 1[R] (start byte), the IIC recognizes this as the address of a slave device with an all-zero address, but not as the general call address.

When the IIC detects the general call address, both the GCA flag in ICSR1 and the RDRF flag in ICSR2 set to 1 on the rising edge of the ninth cycle of the SCL clock. This leads to the generation of a receive data full interrupt (IICn_RXI). The value of the GCA flag can be checked to confirm that the general call address was transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

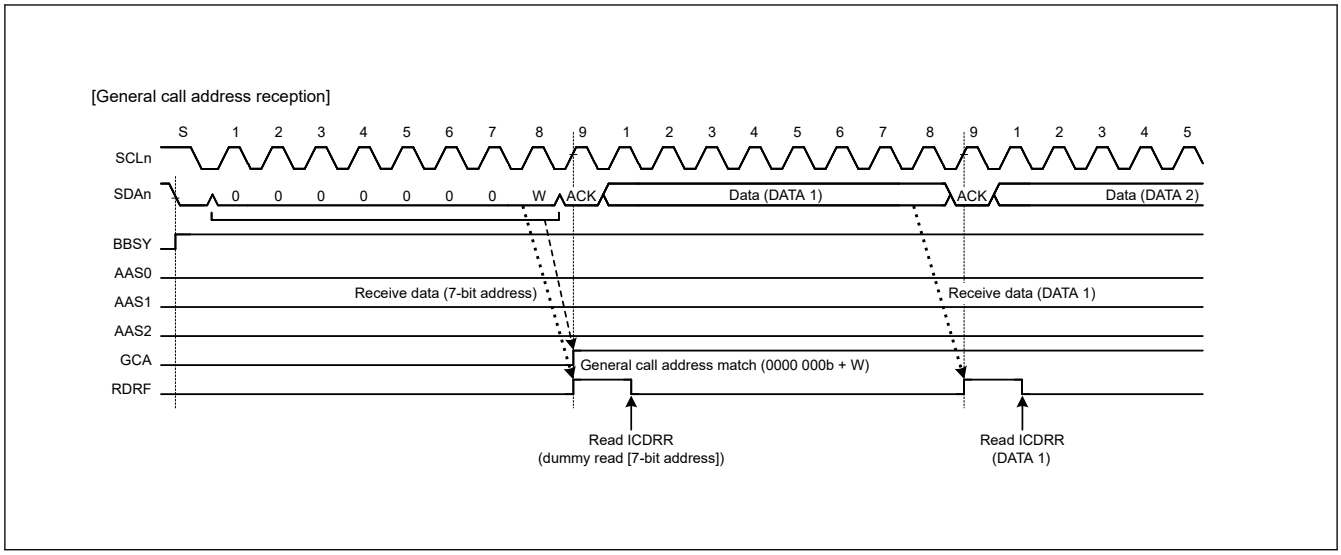


Figure 25.27 Timing of GCA flag setting during reception of general call address

25.7.3 Device-ID Address Detection

The IIC module provides detection of device-ID address compliant with the I^2C bus specification (revision 03). When the IIC receives 1111 100b as the first byte after a start or restart condition is issued with the DIDE bit in ICSER set to 1, it recognizes the address as a device ID, sets the DID flag in ICSR1 to 1 on the rising edge of the ninth SCL clock cycle when the subsequent R/W# bit is 0, then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the IIC sets the associated AASy flag (y = 0 to 2) in ICSR1 to 1.

When the first byte received after the issue of a start or restart condition matches the device ID address (1111 100b) again and the subsequent R/W# bit is 1, the IIC does not compare the second and subsequent bytes and sets the ICSR2.TDRE flag to 1.

In the device ID address detection function, the IIC sets the DID flag to 0 if a match with the IIC slave address is not obtained or a match with the device ID address is not obtained after a match with the IIC slave address and a restart condition is not detected. If the first byte after detection of a start or restart condition matches the device ID address (1111 100b), and the R/W# bit is 0, the IIC sets the DID flag to 1 and compares the second and subsequent bytes with the slave address of the IIC. If the R/W# bit is 1, the DID flag holds the previous value and the IIC does not compare the second and subsequent bytes. Therefore, the reception of a device ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Additionally, prepare the device ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal transmit data. For details on the information that must be included in device ID fields, contact NXP Semiconductors.

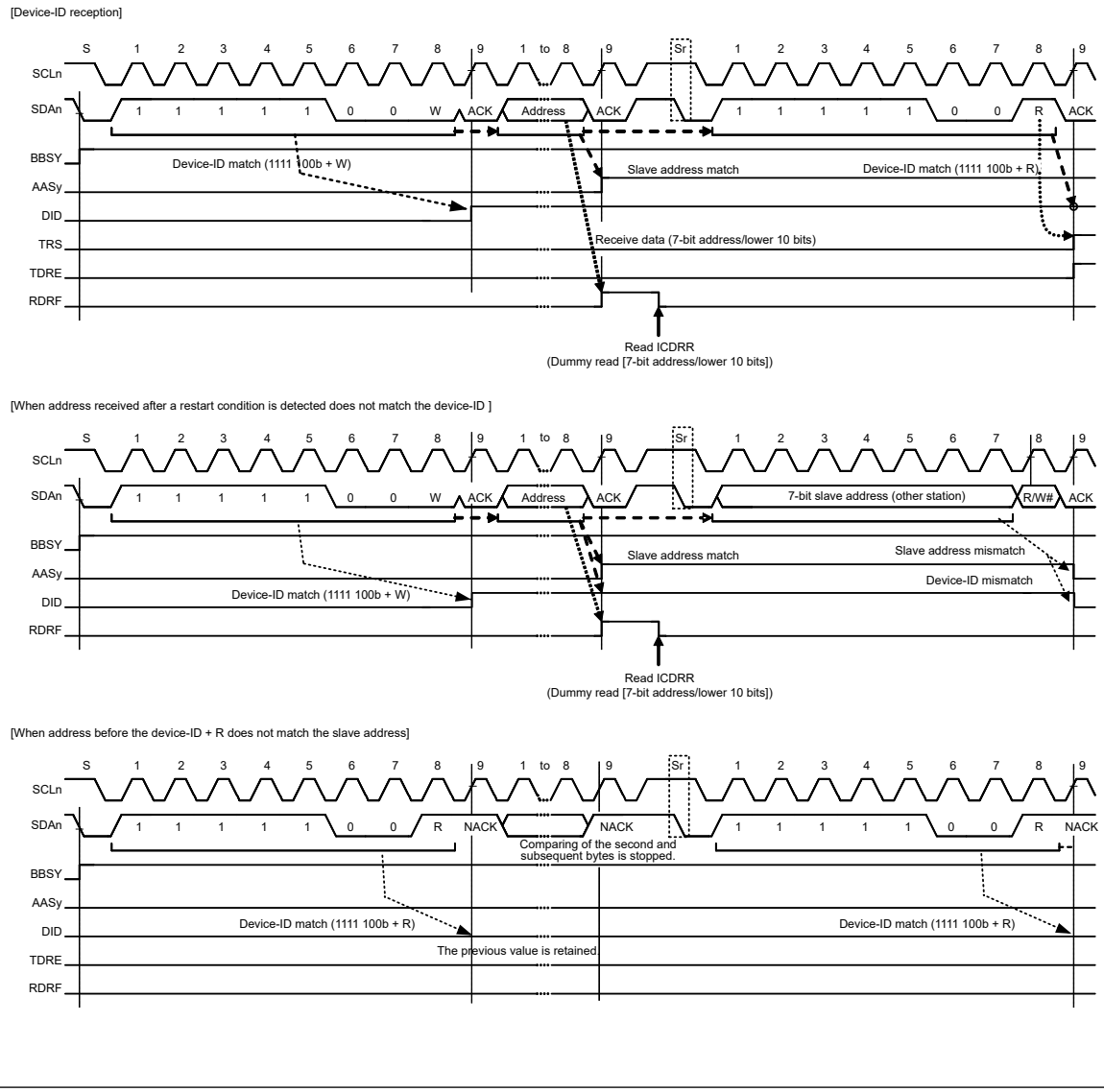


Figure 25.28 AASy and DID flag set and clear timing during reception of device ID

25.7.4 Host Address Detection

The IIC provides host address detection when operating in SMBus. When the HOAE bit in ICSER is set to 1 while the SMBS bit in ICMR3 is 1, the IIC can detect the host address (0001 000b) in slave receive mode (MST and TRS bits = 00b in ICCR2).

When the IIC detects the host address, the HOA flag in ICSR1 is set to 1 on the rising edge of the 9th SCL clock cycle, and at the same time, the RDRF flag in ICSR2 is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (IICn_RXI) to be generated. The HOA flag indicates that the host address was sent from another device.

If the bit following the host address (0001 000b) is an Rd bit (R/W# bit = 1), the IIC can also detect the host address. After the host address is detected, the IIC operates in the same manner as in normal slave operation.

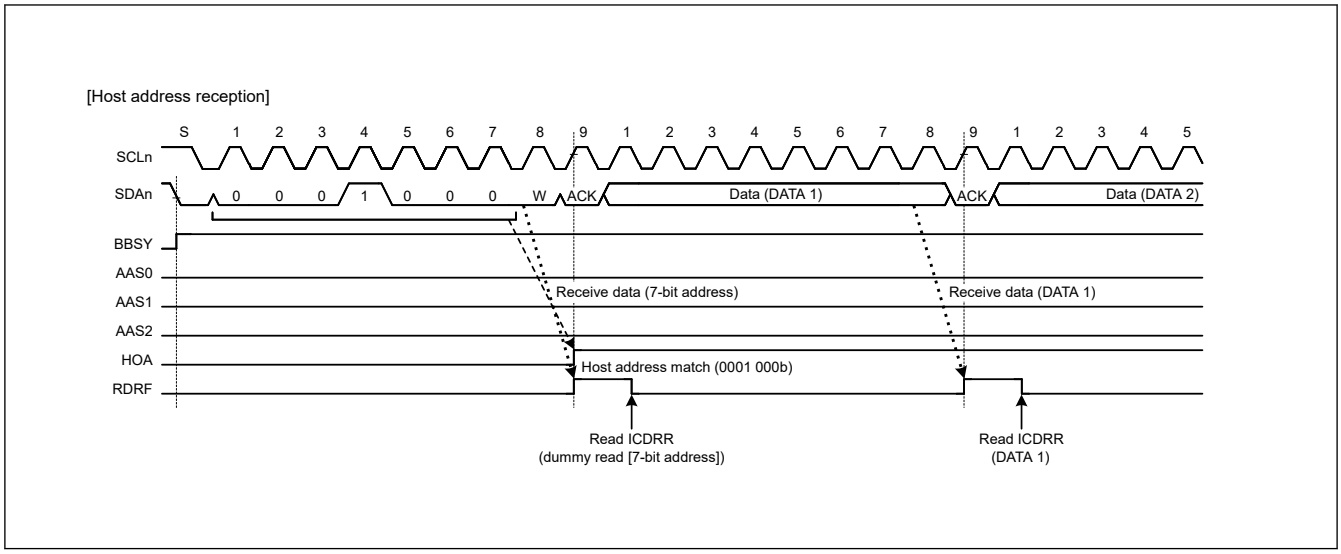


Figure 25.29 HOA flag set timing during reception of host address

25.8 Wakeup Function

The IIC provides a wakeup function that causes the MCU to transition from Software Standby mode or Snooze mode to normal operation. The wakeup function enables the reception of data when the system clock (PCLKB) is stopped, and generates a wakeup interrupt signal on a match of the slave address of the received data. This wakeup interrupt signal triggers the return to normal operation. After the wakeup interrupt occurs, switch the IIC to PCLKB synchronous operation so that communication can continue.

The wakeup function has four operation modes:

- Normal wakeup mode 1
- Normal wakeup mode 2
- Command recovery mode
- EEP response mode

Table 25.9 describes the behavior in these modes.

Table 25.9 Wakeup operation modes

Operation mode	ACK response timing	ACK response before wakeup to PCLKB synchronous operation	SCL state during wakeup to PCLKB synchronous operation
Normal wakeup mode 1	Before wakeup to PCLKB synchronous operation*1	ACK	Fixed low
Normal wakeup mode 2	After wakeup to PCLKB synchronous operation*2	Before wakeup: no response (NACK level retained) After wakeup: ACK response	Fixed low
Command recovery mode	Before wakeup to PCLKB synchronous operation*1	ACK	Open
EEP response mode	Before recovery to PCLKB synchronous operation*1	NACK	Open

Note 1. Switching timing from PCLKB asynchronous operation to PCLKB synchronous operation is the falling edge of the 9th clock of the SCL.
Note 2. Switching timing from PCLKB asynchronous operation to PCLKB synchronous operation is the falling edge of the 8th clock of the SCL.

The following can be selected as wakeup interrupt sources:

- Host address detection (valid when ICSER.HOAE = 1)
- General call address detection (valid when ICSER.GCAE = 1)
- Slave address 0*1 detection (valid when ICSER.SAR0E = 1)

- Slave address 1[*1] detection (valid when ICSER.SAR1E = 1)
- Slave address 2[*1] detection (valid when ICSER.SAR2E = 1)

Note 1. Only 7-bit address can be set. Set the FS bit in SARUy (y = 0 to 2) to 0.

Precautions on the use of the wakeup function

- Do not change the content of the IIC registers except the ICIER register and WUSEN bit in ICWUR2 while the WUASYF flag in ICWUR2 is 1 (during PCLKB asynchronous operation).
- Set ICWUR.WUE and ICWUR.WUIE to 1, and ICCR2.MST and ICCR2.TRS to 0 (slave reception mode) before switching to PCLKB asynchronous mode.
- The device ID and the 10-bit slave address cannot be selected for the wakeup interrupt source. Set the DIDE bit in ICSER and FS bit in SARUy (y = 0 to 2) to 0.
- Set bits TIE, TEIE, RIE, NAKIE, SPIE, STIE, ALIE, and TMOIE in the ICIER register to 0 (interrupt disabled) after switching to PCLKB asynchronous operation (ICWUR2.WUASYF = 1).
- When the wakeup function is enabled, do not use the timeout function (ICWUR.WUE = 1)
- Even when a wakeup interrupt is generated during PCLKB asynchronous operation (when ICWUR2.WUASYF = 1), if the slave addresses match in PCLKB synchronous mode (ICWUR2.WUASYF = 0), the wakeup interrupt does not occur and the WUF flag is not set.
- If the timing of writing 0 to the ICWUR2.WUSEN bit and the timing of detecting a start condition conflict, the IIC might start the next reception in PCLKB synchronous operation mode. In this case, ICWUR2.WUASYF flag becomes 1 (switch to PCLKB asynchronous mode) when data communication is complete, a stop condition is detected, and detection of a wakeup event starts.
- After writing 0 to the WUSEN bit in ICWUR2, do not change registers relate to the IIC operation mode setting (ICMR3, ICSER, and SARLy) until the mode is switched to PCLKB asynchronous operation from PCLKB synchronous operation (while the ICWUR2.WUASYF flag is 1). If the register value changes during this period by an interrupt handling or another factor, the IIC might malfunction before switching to the asynchronous operation.

25.8.1 Normal Wakeup Mode 1

This section describes the behavior, the timing, and an example operation of normal wakeup mode 1.

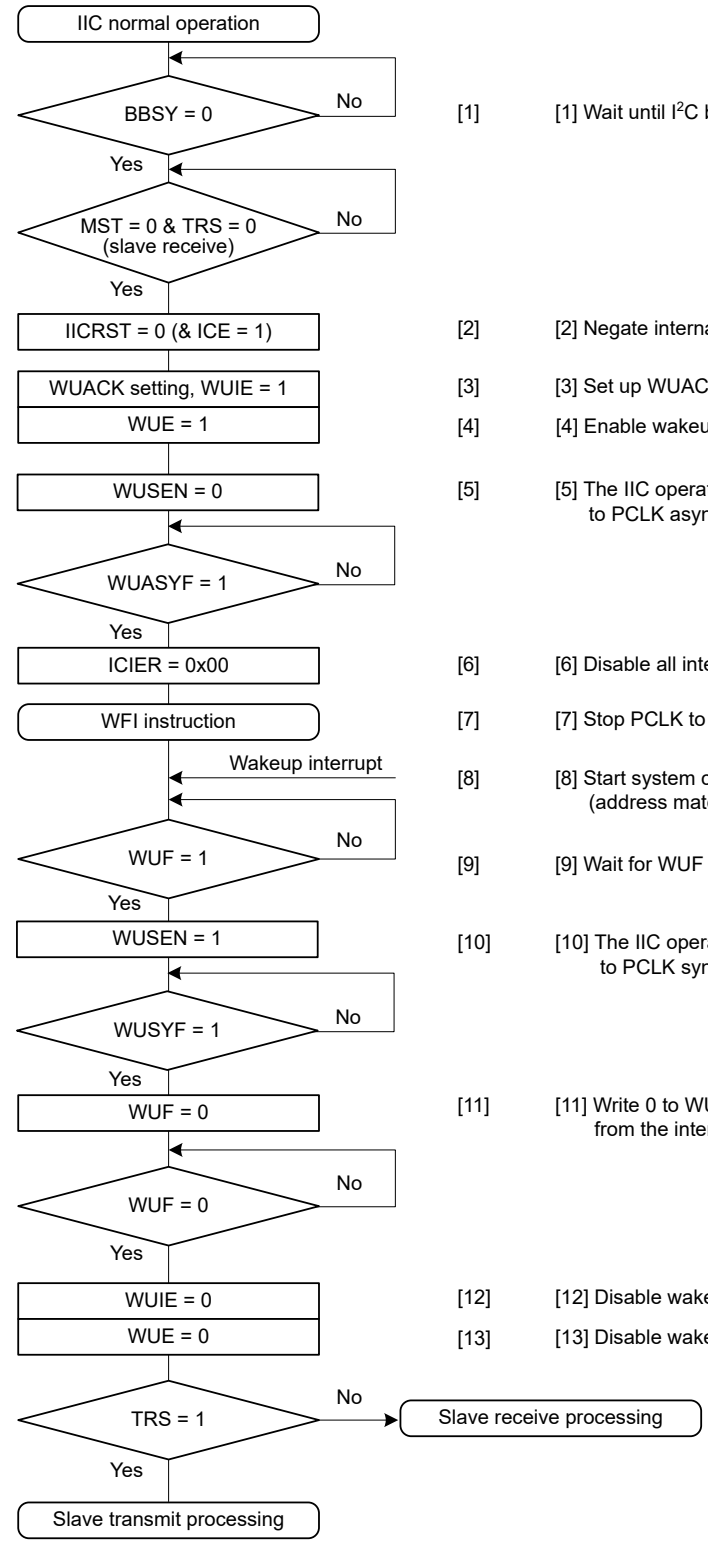
In normal wakeup mode 1, a wakeup interrupt triggered by the match of the slave address initiates the transition to normal operation as follows:

Before wakeup:	ACK is sent in response to the data received with its own slave address of the IIC.
During wakeup:	ACK response is made on the 9th clock cycle of SCL, after which SCL is held low[*1].
After wakeup:	Normal operation continues.

Note 1. Between the 9th clock cycle and 1st clock cycle during wakeup, WAIT = 1 is invalid.

If the slave address does not match, the SCL line is not held low after the 9th clock cycle of SCL, and the slave operation continues. Figure 25.30 shows an operation example, and Figure 25.32 shows the detailed timing.

If the transition from Software Standby mode or Snooze mode is triggered by an interrupt other than a wakeup interrupt, for example the IRQn, the WUF flag is not set to 1. Figure 25.31 shows an operation example.



[1] [1] Wait until I^2C bus is open and stay in standby state.

[2] [2] Negate internal reset (if asserted).

[3] [3] Set up WUACK for the desired wakeup mode. Enable wakeup interrupt.

[4] [4] Enable wakeup function.

[5] [5] The IIC operating state is changed from PCLK synchronous to PCLK asynchronous.

[6] [6] Disable all interrupt requests except WUI.

[7] [7] Stop PCLK to IIC. IIC continues to receive.

[8] [8] Start system clock and PCLK supply to IIC on wakeup interrupt (address match).

[9] [9] Wait for WUF = 1.

[10] [10] The IIC operating state is changed from PCLK asynchronous to PCLK synchronous.

[11] [11] Write 0 to WUF. Read and check that WUF = 0 before returning from the interrupt handling.

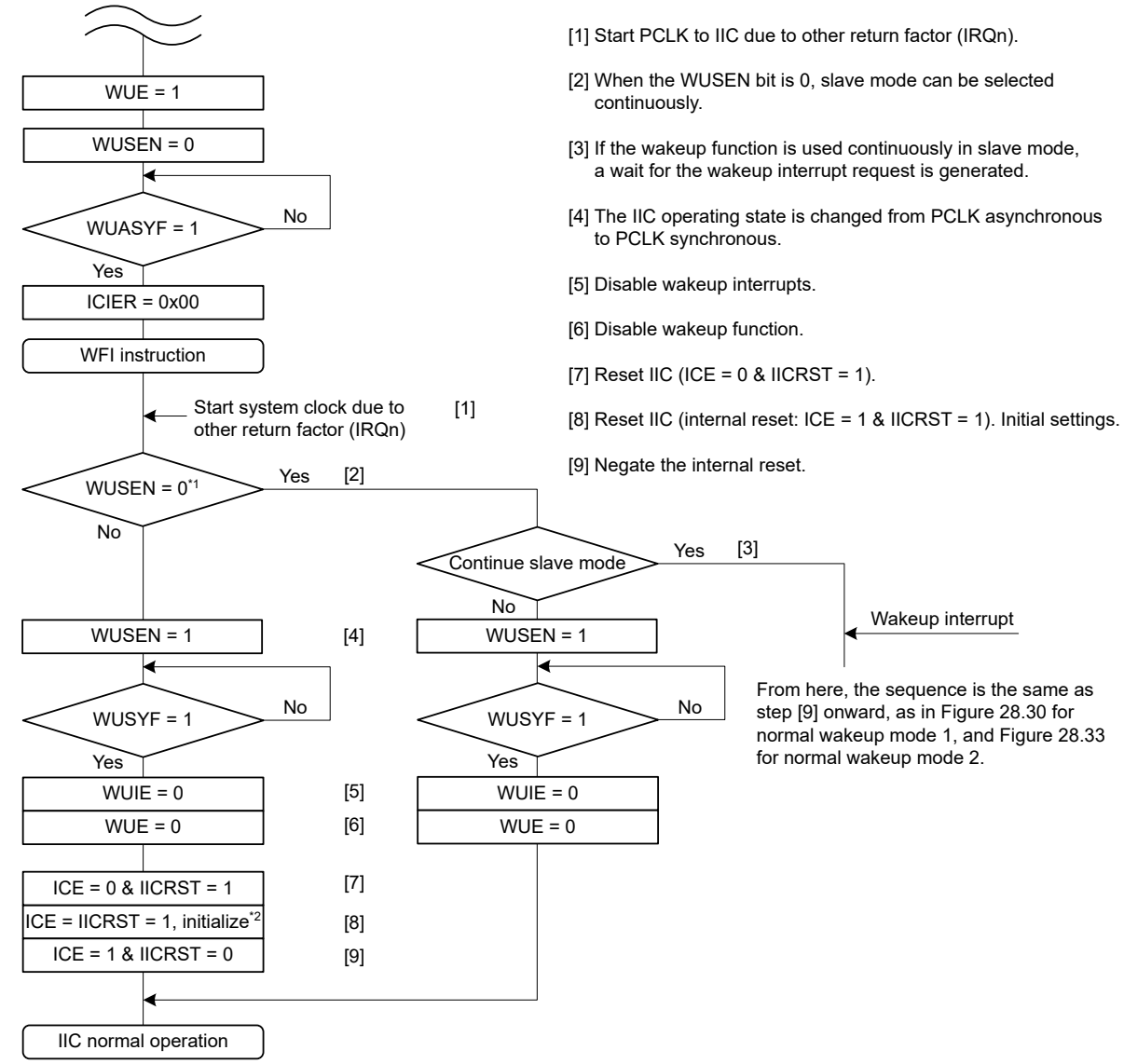
[12] [12] Disable wakeup Interrupt.

[13] [13] Disable wakeup function.

Note: PCLK = PCLKB

Figure 25.30 Example operation of normal wakeup mode 1 when wakeup is triggered by a wakeup interrupt on match of the slave address

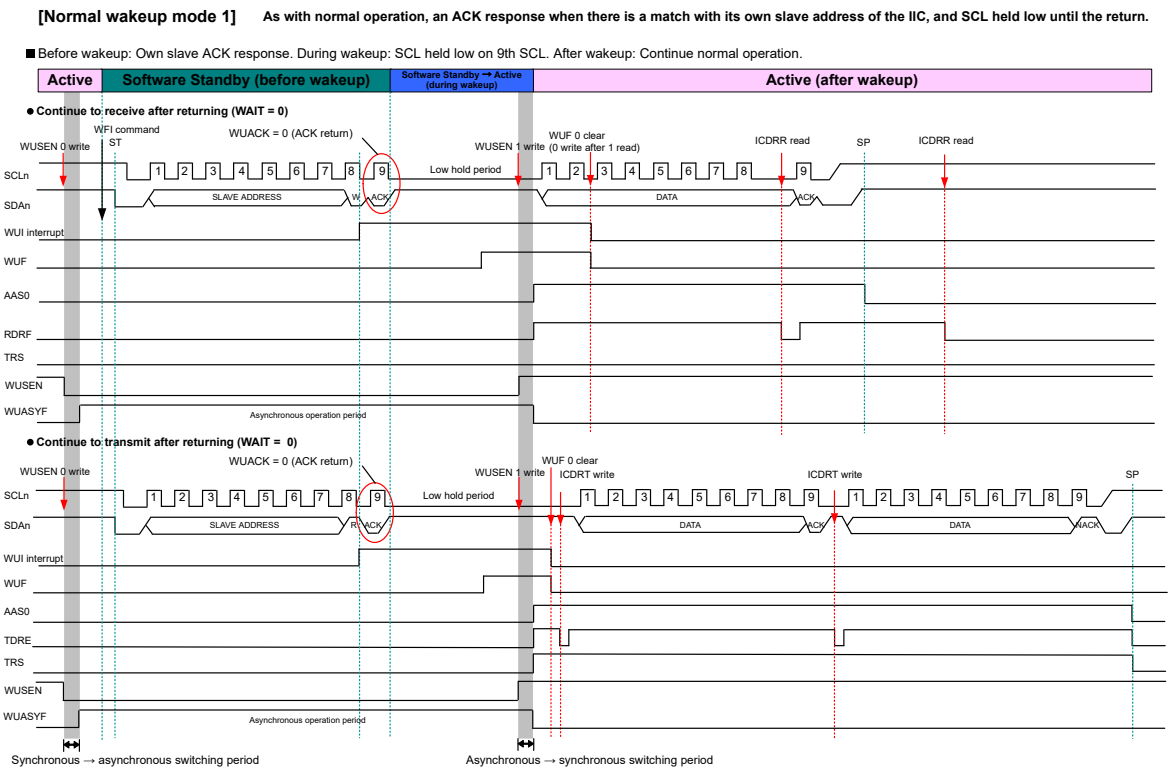
Note: See Precautions on the use of the wakeup function.



Note: PCLK = PCLKB
Note 1. In the MCU, WUSEN bit is always 0 in the IIC.
Note 2. For details on the IIC initial settings, see section 25.3.2. Initial Settings.

Figure 25.31 Example operation of normal wakeup modes 1 and 2 when wakeup is triggered by an interrupt other than IIC wakeup interrupt, for example, the IRQn

Note: For details on the IIC initial settings, see section 25.3.2. Initial Settings.



Note: n = 0

Figure 25.32 Timing of normal wakeup mode 1

25.8.2 Normal Wakeup Mode 2

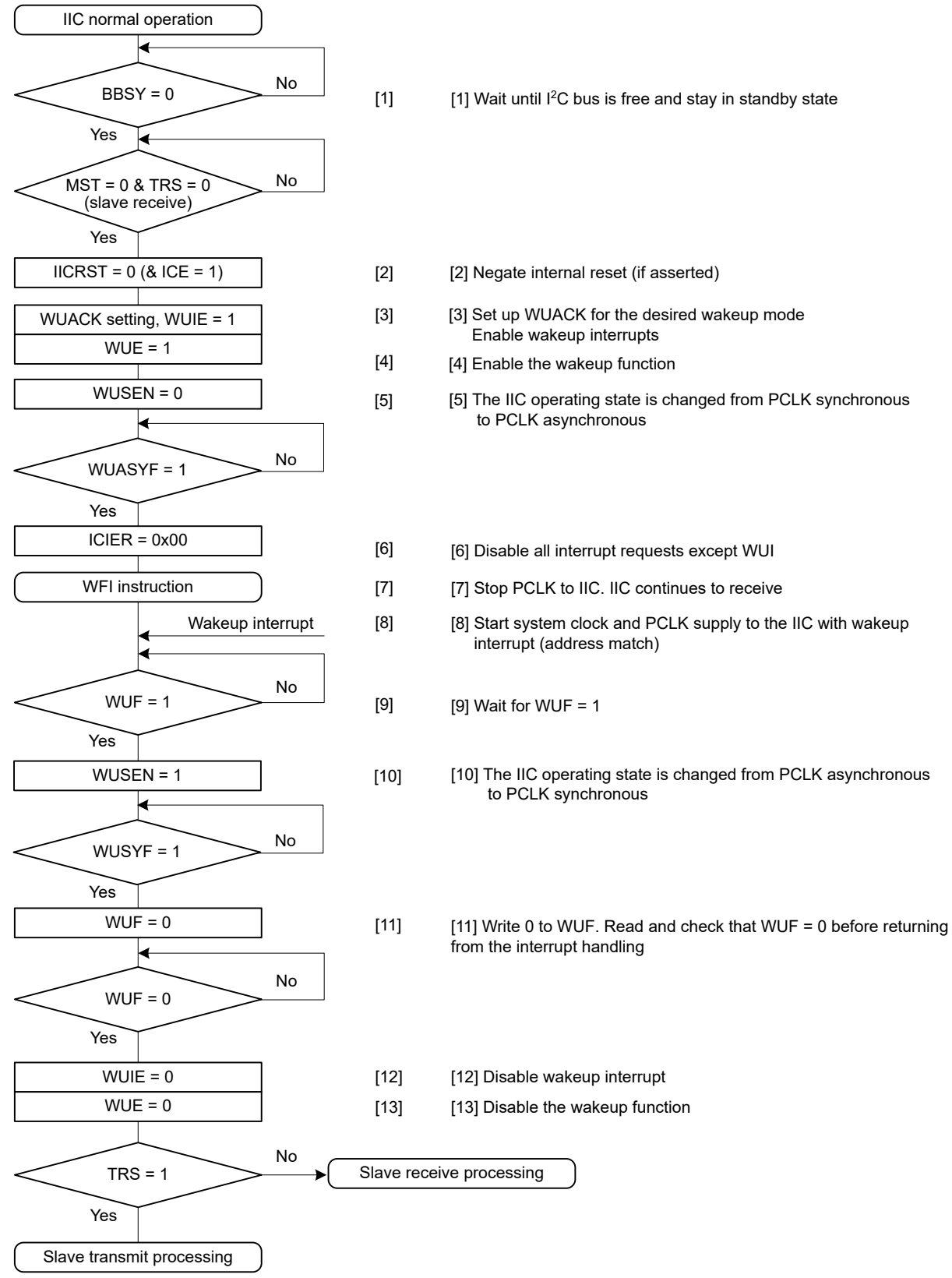
This section describes the behavior, the timing, and an example operation of normal wakeup mode 2.

In normal wakeup mode 2, a wakeup interrupt triggered by a match of the slave address initiates the transition to normal operation as follows:

Before wakeup:	No response to data received with its own slave address until the end of the 8th SCL cycle.
During wakeup:	SCL line held low during the 8th and 9th clock cycles.
After wakeup:	ACK returns on the 9th clock cycle of SCL, and normal operation continues.

If the slave address does not match, the SCL line is not held low after the 8 th SCL clock cycle, and the slave operation continues. Figure 25.33 shows an example operation, and Figure 25.34 shows the detailed timing.

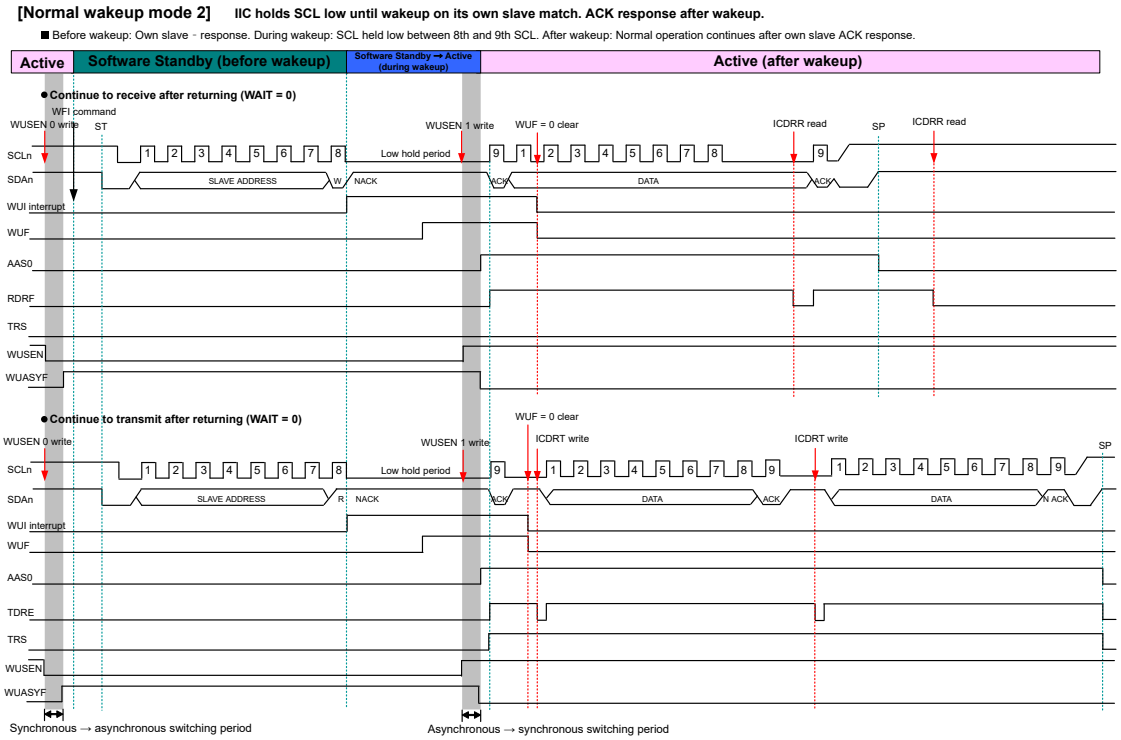
If the transition from Software Standby mode or Snooze mode is triggered by an interrupt other than a wakeup interrupt, such as the IRQn, for example, the WUF flag is not set to 1. Figure 25.31 shows an operation example.



Note: PCLK = PCLKB

Figure 25.33 Example operation of normal wakeup mode 2 when wakeup is triggered by a wakeup interrupt on match of the slave address

Note: See Precautions on the use of the wakeup function.



Note: n = 0

Figure 25.34 Timing of normal wakeup mode 2

25.8.3 Command Recovery Mode and EEP Response Mode (Special Wakeup Modes)

This section describes the behavior, the timing, and example operations of the command recovery and EEP response modes.

In the command recovery and EEP response modes, the SCL line is not held low during the wakeup period (after the rise of the 9th clock cycle of SCL). Therefore, other IIC devices can use the I^2C bus during this period.

A wakeup interrupt triggered by a match of the slave address initiates the transition to normal operation as follows:

Before wakeup:	In response to the data received with its own slave address, the IIC returns ACK (command recovery mode) or NACK (EEP response mode).
During wakeup:	The SCL line is not held low.
After wakeup:	Normal operation continues after IIC initialization.

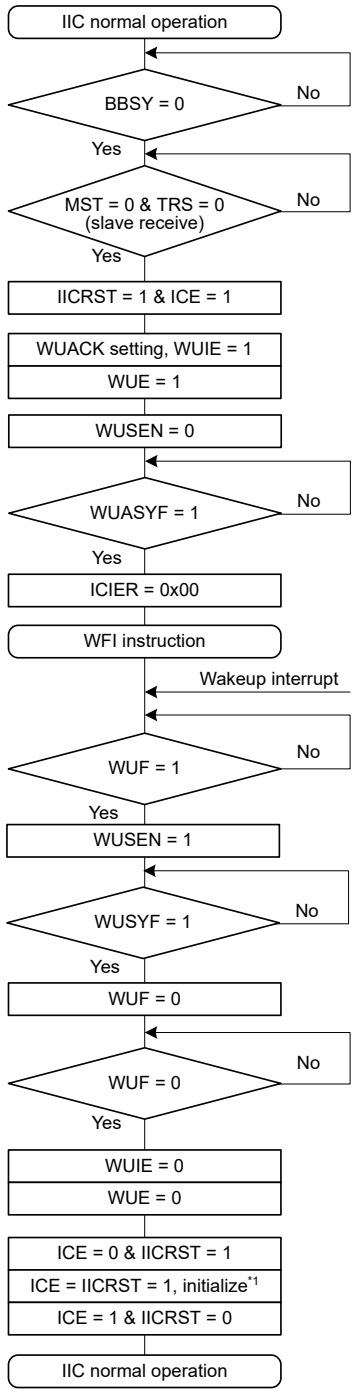
If the slave address does not match, the slave operation continues.

Note: Because the SCL line is not held low during wakeup, transmission or reception of the data that follows the slave address is not possible.

Note: The command recovery and EEP response modes are internal reset states (ICE = IICRST = 1). Therefore, the match of the slave address does not set the flags, HOA, GCA, ASS0, ASS1, and ASS2 in the ICSR1 register.

Figure 25.35 shows an example operation in recovery and EEP response modes. Figure 25.37 shows the detailed timing.

If the transition from Software Standby mode or Snooze mode is triggered by an interrupt other than a wakeup interrupt, such as IRQn for example, the WUF flag is not set to 1. Follow the processing shown in Figure 25.36.



[1] [1] Wait until I^2C bus is open and stay in standby state.

[2] [2] Internal reset is asserted.

[3] [3] Set up WUACK for the desired wakeup mode. Enable wakeup interrupt.

[4] [4] Enable wakeup function.

[5] [5] The IIC operating state is changed from PCLK synchronous to PCLK asynchronous.

[6] [6] Disable all interrupt requests except WUI.

[7] [7] Stop PCLK to IIC. IIC continues to receive.

[8] [8] Start system clock and PCLK supply to the IIC with wakeup interrupt (address match).

[9] [9] Wait for WUF = 1.

[10] [10] The IIC operating state is changed from PCLK asynchronous to PCLK synchronous.

[11] [11] Write 0 to WUF. Read and check that WUF = 0 before returning from the interrupt handling.

[12] [12] Disable wakeup interrupt.

[13] [13] Disable wakeup function.

[14] [14] Reset IIC (ICE = 0 & IICRST = 1).

[15] [15] Reset IIC (internal reset: ICE = 1 & IICRST = 1). Initial settings.

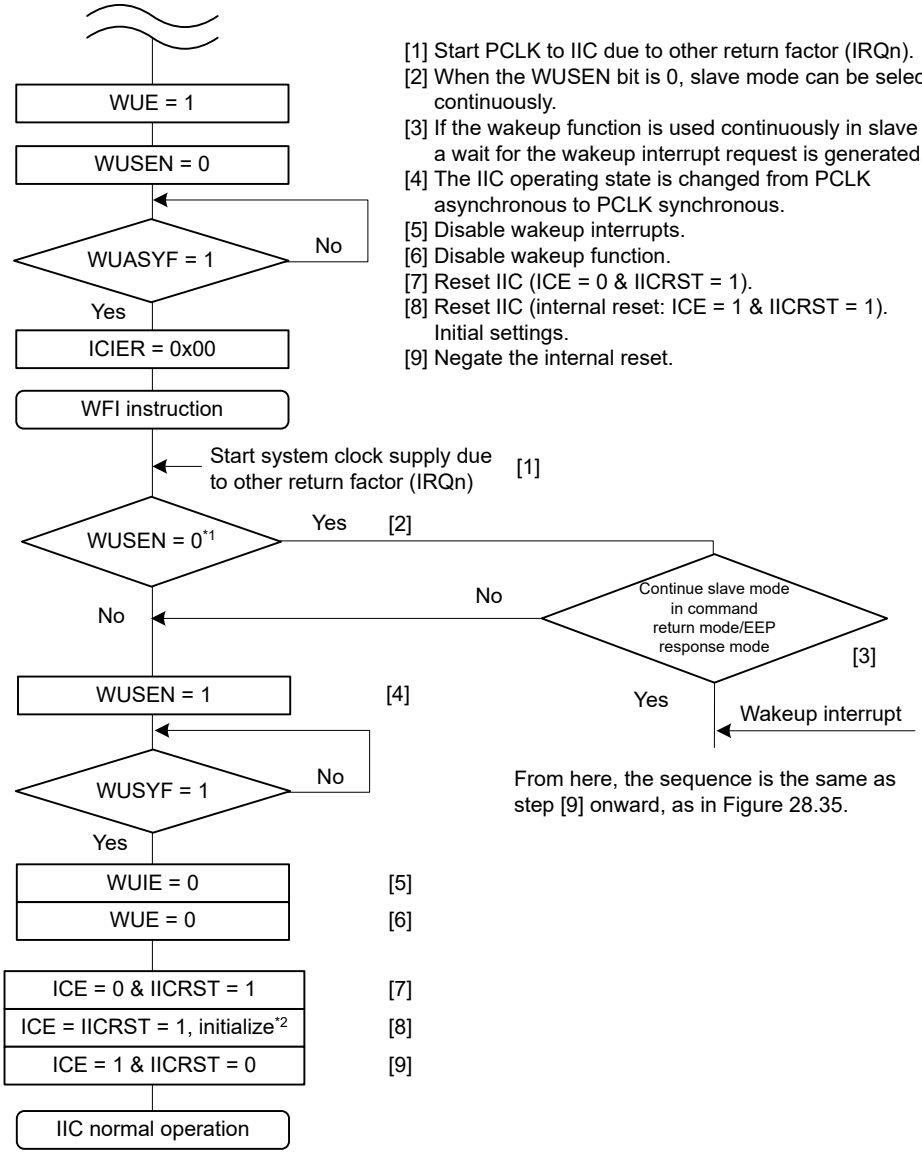
[16] [16] Negate the internal reset.

Note: PCLK = PCLKB

Note 1. For details on IIC initialization, see section 25.3.2. Initial Settings.

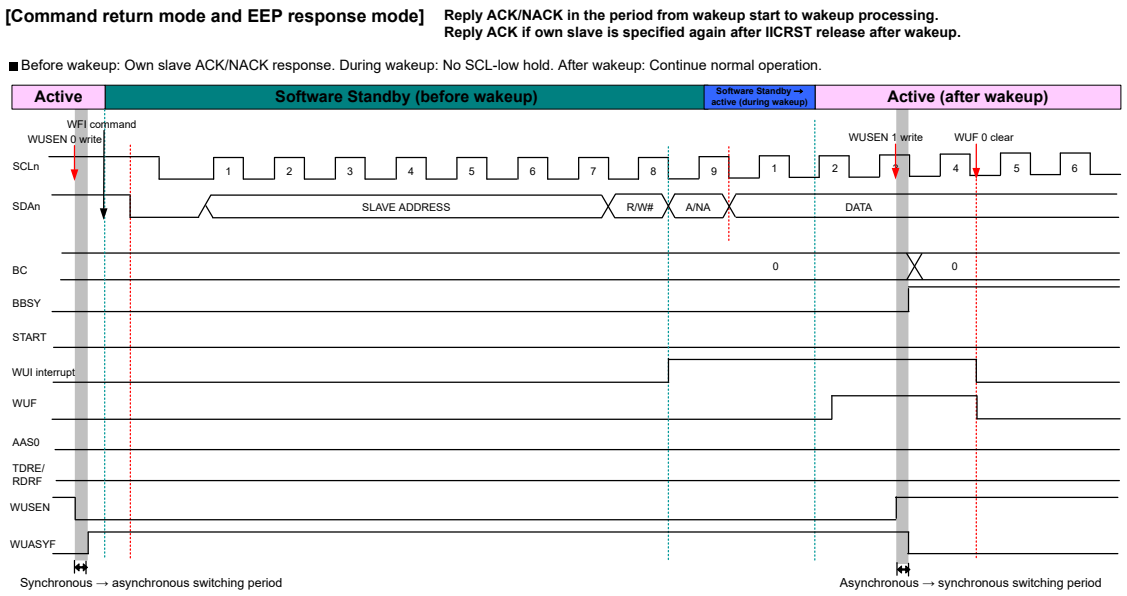
Figure 25.35 Example operation of command recovery mode and EEP response mode when wakeup is triggered by a wakeup interrupt on match of the slave address

Note: See Precautions on the use of the wakeup function.



Note: PCLK = PCLKB
Note 1. In the MCU, WUSEN bit is always 0 in the IIC.
Note 2. For details on IIC initialization, see section 25.3.2. Initial Settings.

Figure 25.36 Example operation of command recovery and EEP response modes when wakeup is triggered by an interrupt other than IIC wakeup interrupt, for example, the IRQn



Note: n = 0

Figure 25.37 Timing of command recovery and EEP response modes

25.9 Automatic Low-Hold Function for SCL

25.9.1 Function to Prevent Wrong Transmission of Transmit Data

If the I^2C Bus Shift Register (ICDRS) is empty and data has not been written to the I^2C Bus Transmit Data Register (ICDRT) with the IIC in transmission mode (TRS bit = 1 in ICCR2), the SCLn line is automatically held low over the subsequent intervals. This low-hold period is extended until the transmit data is written, which prevents the unintended transmission of erroneous data.

Master transmit mode:

- Low-level interval after a start or restart condition is issued
- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

Slave transmit mode:

- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

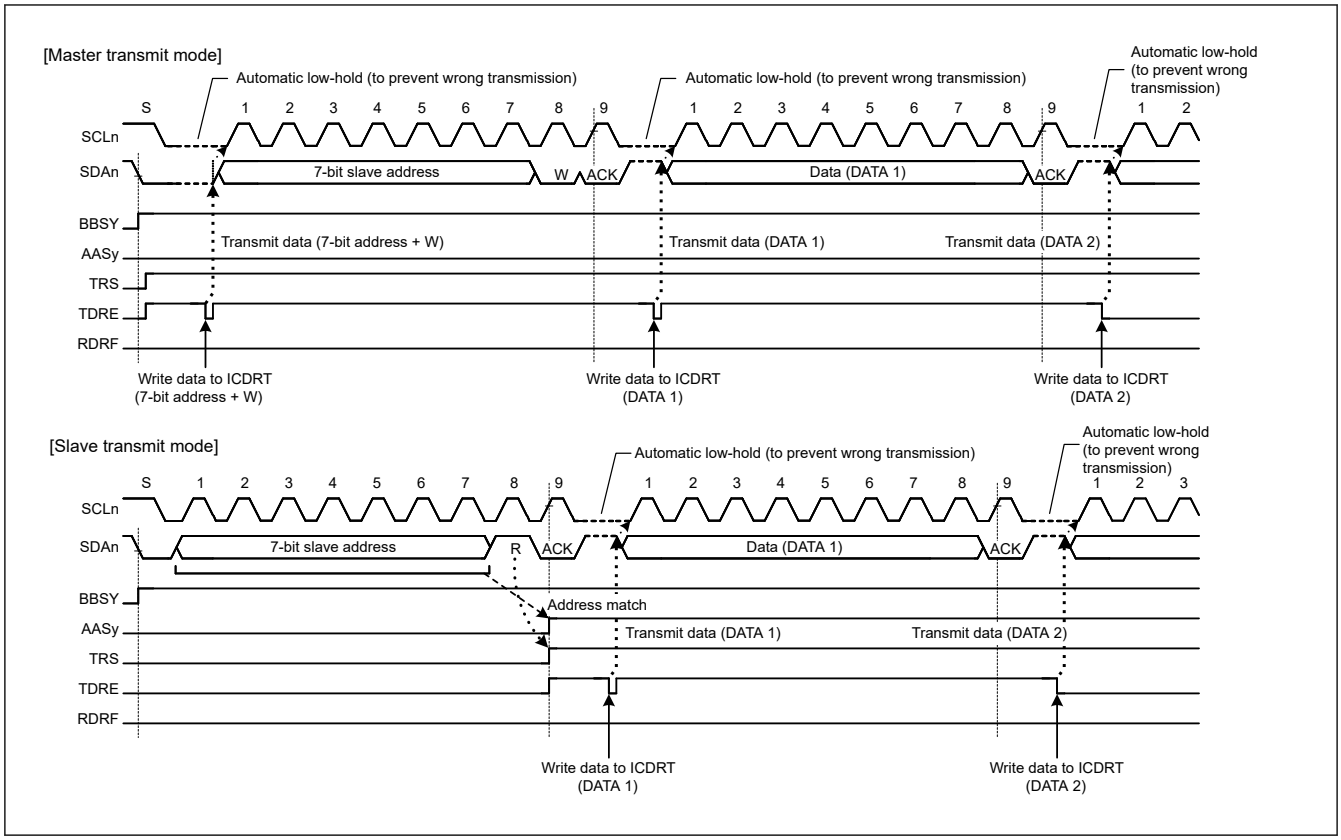


Figure 25.38 Automatic low-hold operation in transmit mode

25.9.2 NACK Reception Transfer Suspension Function

This function suspends transfer operation when NACK is received in transmit mode (TRS bit = 1 in ICCR2). This function is enabled when the NACKE bit in ICFER is set to 1. If the next transmit data is already written (TDRE flag = 0 in ICSR2) when NACK is received, the next data transmission on the falling edge of the 9th SCL clock cycle is automatically suspended. This prevents the SDAn line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (NACKF flag = 1 in ICSR2), transmit and receive operations are discontinued. To restore transmit or receive operation, after issuing the restart condition, you need to set the NACKF flag to 0 and try again, or set the NACKF flag to 0 after issuing the stop condition and then start again from issuing the start condition.

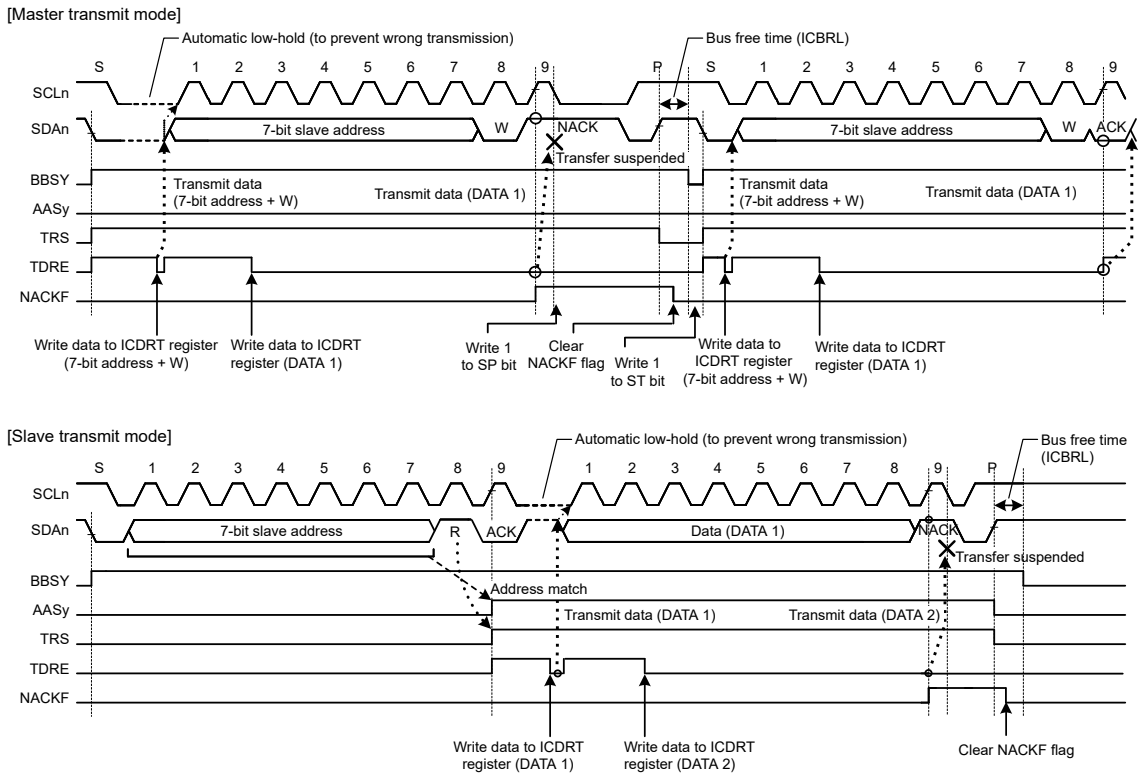


Figure 25.39 Suspension of data transfer when NACK is received, when NACKE = 1

25.9.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDRR) read is delayed for a period of one transfer frame or more with receive data full (RDRF flag = 1 in ICSR2) in receive mode (TRS = 0 in ICCR2), the IIC holds the SCLn line low automatically immediately before the next data is received to prevent a failure to receive data.

This function is enabled even if the read processing of the final receive data is delayed and, in the meantime, the IIC slave address is designated after a stop condition is issued. This function does not interfere with other communication because the IIC does not hold the SCLn line low when a mismatch with its own slave address occurs after a stop condition is issued.

Periods in which the SCLn line is held low can be selected with a combination of the WAIT and RDRFS bits in ICMR3.

(1) 1-byte receive operation and automatic low-hold function using the WAIT bit

When the WAIT bit in ICMR3 is set to 1, the IIC performs a 1-byte receive operation using the WAIT bit function. Additionally, when the ICMR3.RDRFS bit is 0, the IIC automatically sends the ACKBT bit value in ICMR3 for the acknowledge bit in the period from the falling edge of the 8th SCL clock cycle to the falling edge of the 9th SCL clock cycle, and automatically holds the SCLn line low on the falling edge of the 9th SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from ICDRR, which enables byte-wise receive operation.

The WAIT bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master or slave receive mode.

(2) 1-byte receive operation (ACK/NACK transmission control) and automatic low-hold function using the RDRFS bit

When the RDRFS bit in ICMR3 is set to 1, the IIC performs a 1-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the RDRF flag in ICSR2 is set to 1 (receive data full) on the rising edge of the eighth SCL clock cycle, and the SCLn line is automatically held low on the falling edge of the eighth SCL clock cycle. This low-hold is released by writing a value to the ACKBT bit in ICMR3, but cannot be released by reading data from ICDRR, which enables receive operation through the ACK or NACK transmission control based on the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master or slave receive mode.

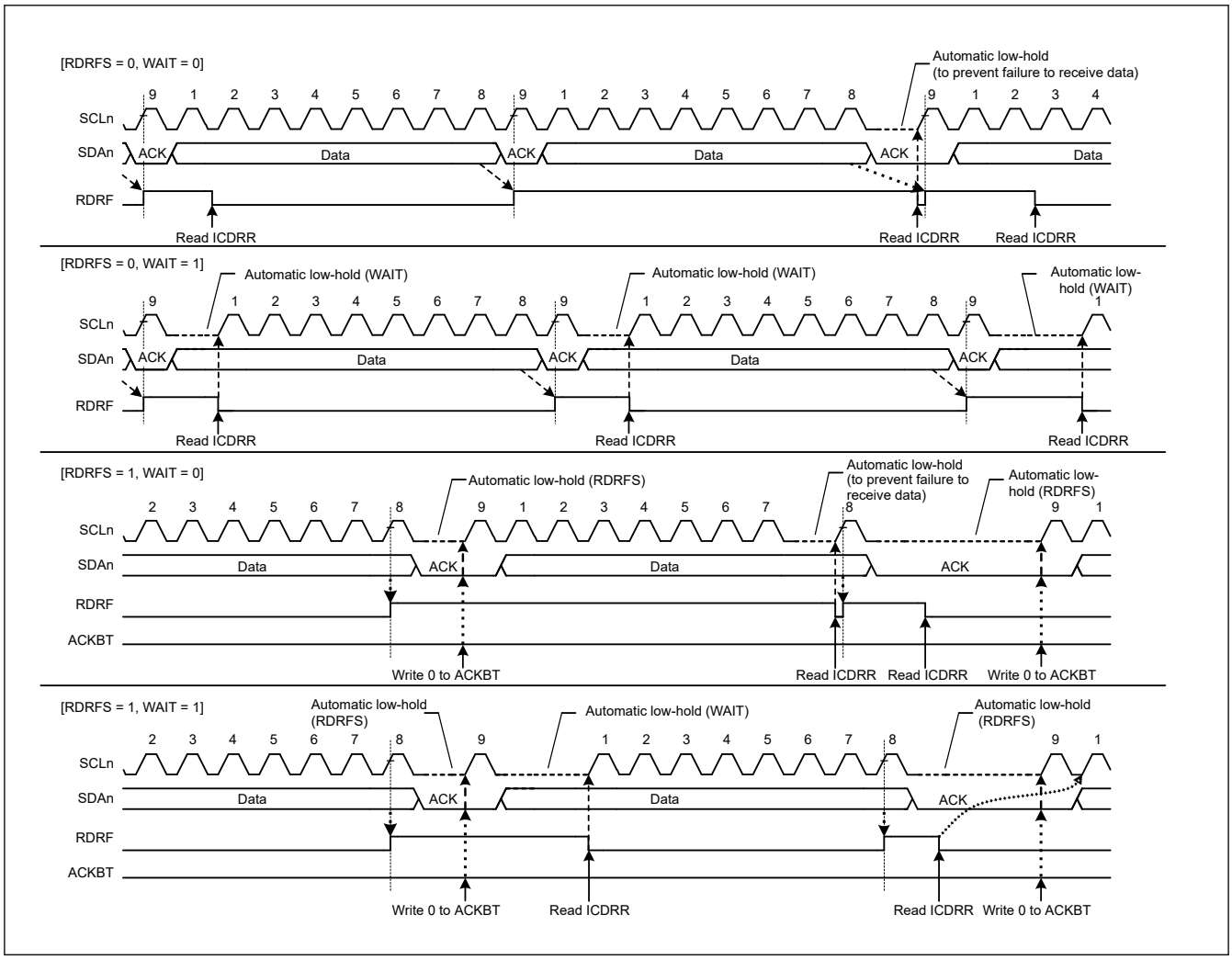


Figure 25.40 Automatic low-hold operation in receive mode using the RDRFS and WAIT bits

25.10 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I^2C bus standard, the IIC provides functions to prevent double-issue of a start condition, detect arbitration-lost during transmission of NACK, and detect arbitration-lost in slave transmit mode.

25.10.1 Master Arbitration-Lost Detection (MALE Bit)

The IIC drives the SDAn line low to issue a start condition. However, if the SDAn line was already driven low by another master device issuing a start condition, the IIC regards its own start condition as an error and considers this a loss in arbitration. Priority is given to transfer by the other master device. Similarly, if a request to issue a start condition is made by setting the ST bit in ICCR2 to 1 while the bus is busy (BBSY flag = 1 in ICCR2), the IIC regards this as a double-issuing-of-start-condition error and considers itself to have lost in arbitration. This prevents a failure of transfer resulting from a start condition being issued while transfer is in progress.

When a start condition is issued successfully, if the transmit data including the address bits (internal SDA output level) and the level on the SDAn line do not match (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state and a low level is detected on the SDAn line), the IIC loses in arbitration.

After a loss in arbitration of mastership, the IIC immediately enters slave receive mode. If a slave address, including the general call address, matches its own address at this time, the IIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the MALE bit in ICFER is 1 (master arbitration-lost detection enabled).

[Master arbitration-lost conditions]

- Mismatching of the internal level for output on SDA and the level on the SDAn line after a start condition was issued by setting the ST bit in ICCR2 to 1 while the BBSY flag in ICCR2 is set to 0 (erroneous issuing of a start condition)
- Setting of the ST bit in ICCR2 to 1 (start condition double-issue error) while the BBSY flag is 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDAn line in master transmit mode (MST and TRS bits = 11b in ICCR2).

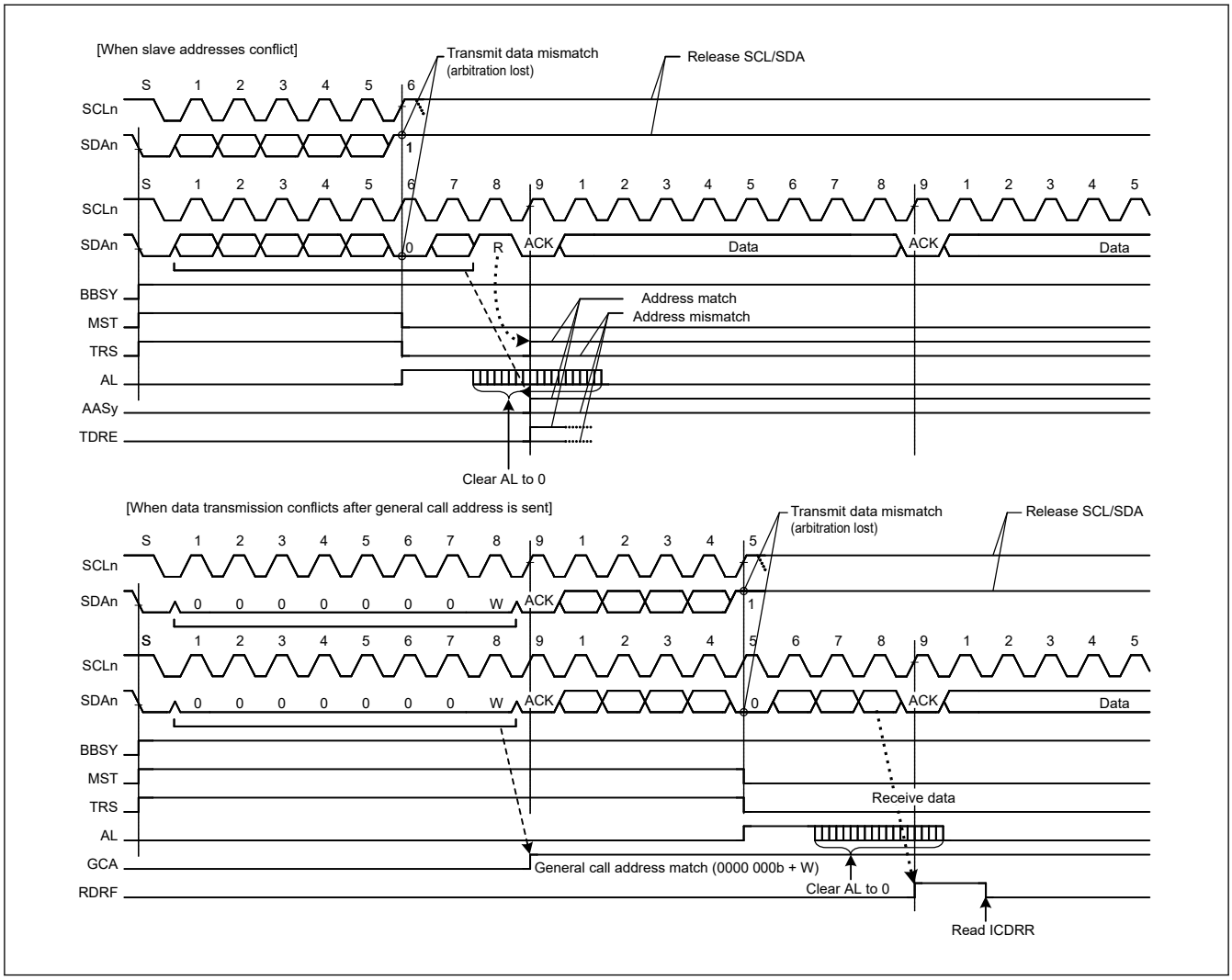


Figure 25.41 Examples of master arbitration-lost detection when MALE = 1

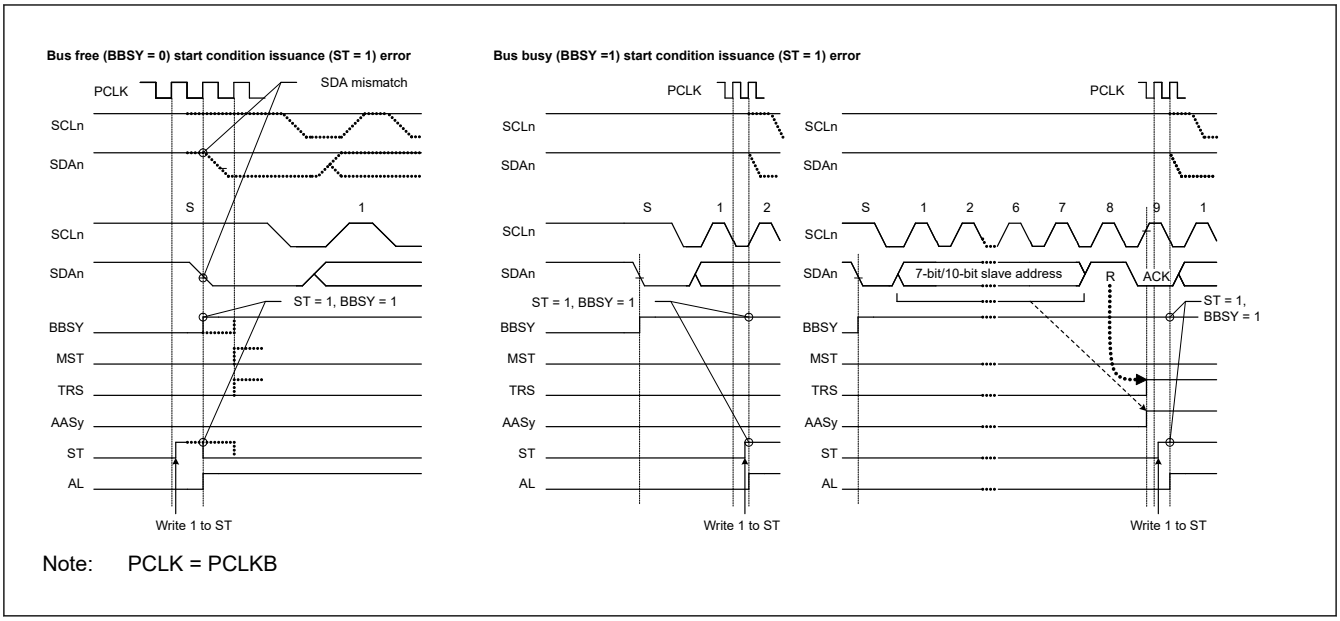


Figure 25.42 Arbitration-lost when start condition is issued when MALE = 1

25.10.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

This function causes arbitration to be lost if the internal SDA output level does not match the level on the SDAn line (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state) and the low level is detected on the SDAn line during transmission of NACK in receive mode. Arbitration is lost because of a conflict between NACK and ACK transmissions when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such a conflict occurs when multiple master devices send or receive the same information through a single slave device. Figure 25.43 shows an example of arbitration-lost detection during transmission of NACK.

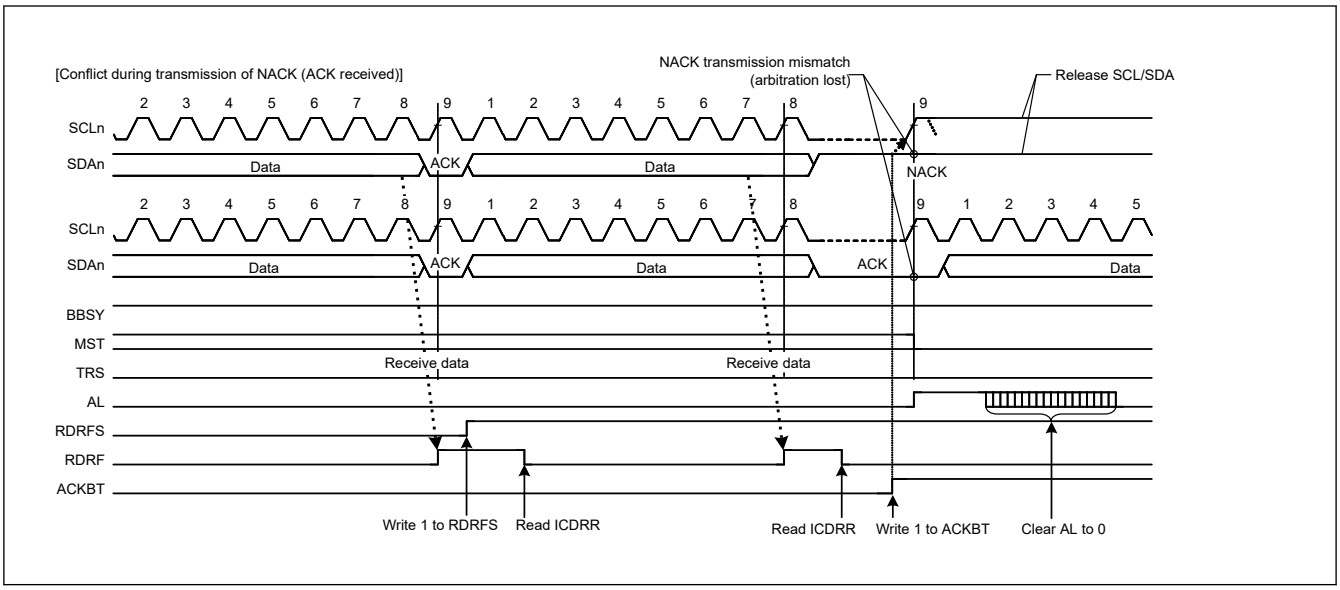


Figure 25.43 Example of arbitration-lost detection during transmission of NACK when NALE = 1

The following explains arbitration-lost detection using an example in which two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives 2 bytes of data from the slave device, and master B receives 4 bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in either master A or master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Master A sends NACK when it has received 2 final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received the required 4 bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like

this occurs, master A cannot detect the ACK transmitted by master B and issues a stop condition. The stop condition issue conflicts with the SCL clock output of master B, which disrupts communication.

When the IIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost. If arbitration is lost during transmission of NACK, the IIC immediately cancels the slave match condition and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing, such as 0xFF transmission processing, which is required if the UDID (Unique Device Identifier) of the assigned address does not match in the Get UDID general processing after the Assign Address command.

The IIC detects arbitration-lost during transmission of NACK when the following condition is met with the NALE bit in ICFER set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

- When the internal SDA output level does not match the SDAn line (ACK is received) during transmission of NACK (ACKBT bit = 1 in ICMR3).

25.10.3 Slave Arbitration-Lost Detection (SALE Bit)

This function causes arbitration to be lost if the transmit data (internal SDA output level) and the level on the SDAn line do not match (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state), and the low level is detected on the SDAn line in slave transmit mode. This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.

When the IIC loses slave arbitration, the IIC is immediately released from the slave-matched state and enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminates subsequent redundant processing for the transmission of 0xFF.

The IIC detects slave arbitration-lost when the following condition is met with the SALE bit in ICFER set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

- When transmit data excluding acknowledge (internal SDA output level) does not match the SDAn line in slave transmit mode (MST and TRS bits = 01b in ICCR2).

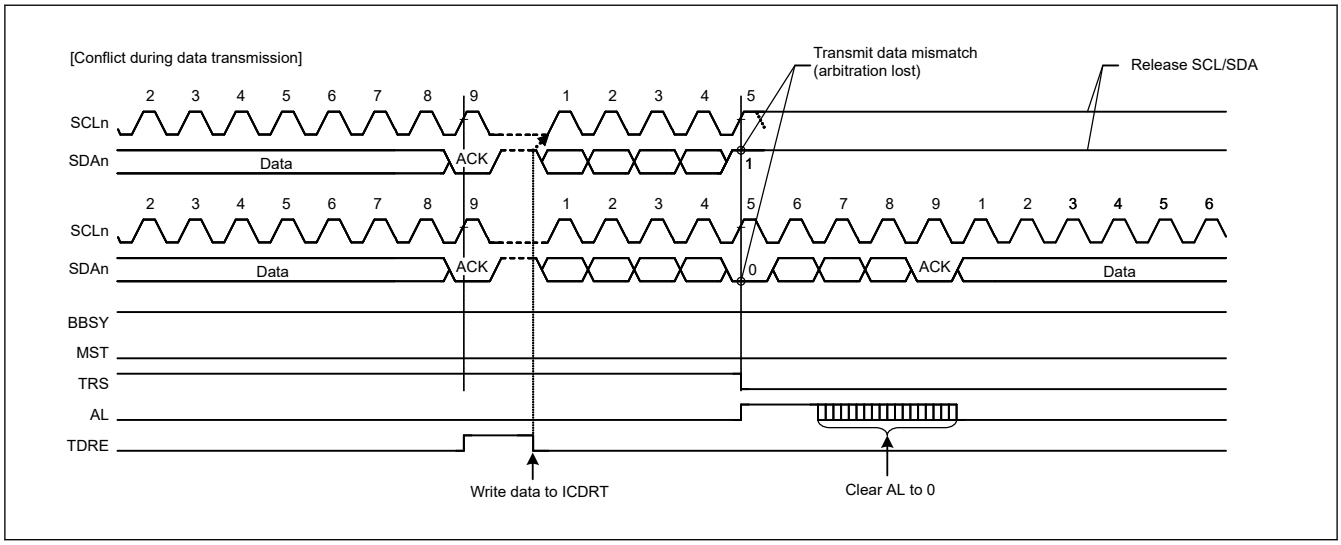


Figure 25.44 Example of slave arbitration-lost detection when SALE = 1

25.11 Start, Restart, and Stop Condition Issuing Function

25.11.1 Issuing a Start Condition

The IIC issues a start condition when the ST bit in ICCR2 is set to 1. When the ST bit is set to 1, a start condition request is made, and the IIC issues a start condition when the BBSY flag in ICCR2 is 0 (bus free state). When a start condition is issued normally, the IIC automatically shifts to the master transmit mode.

To issue a start condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure that the time set in ICBRH and the start condition hold time elapse.
3. Drive the SCLn line low (high level to low level).
4. Detect low level on the SCLn line and ensure the low-level period of the SCLn line set in ICBRL elapses.

25.11.2 Issuing a Restart Condition

The IIC issues a restart condition when the RS bit in ICCR2 is set to 1. When the RS bit is set to 1, a restart condition request is made, and the IIC issues a restart condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a restart condition:

1. Release the SDAn line.
2. Ensure the low-level period of the SCLn line set in ICBRL elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level on the SCLn line and ensure the time set in ICBRL and the restart condition setup time elapse.
5. Drive the SDAn line low (high level to low level).
6. Ensure the time set in ICBRH and the restart condition hold time elapse.
7. Drive the SCLn line low (high level to low level).
8. Detect a low level on the SCLn line and ensure the low-level period of the SCLn line set in ICBRL elapses.

Note: When issuing restart condition requests, write the slave address to ICDRT after confirming that ICCR2.RS = 0. Data written while ICCR2.RS = 1 is not forwarded because of the retransmission condition before the occurrence.

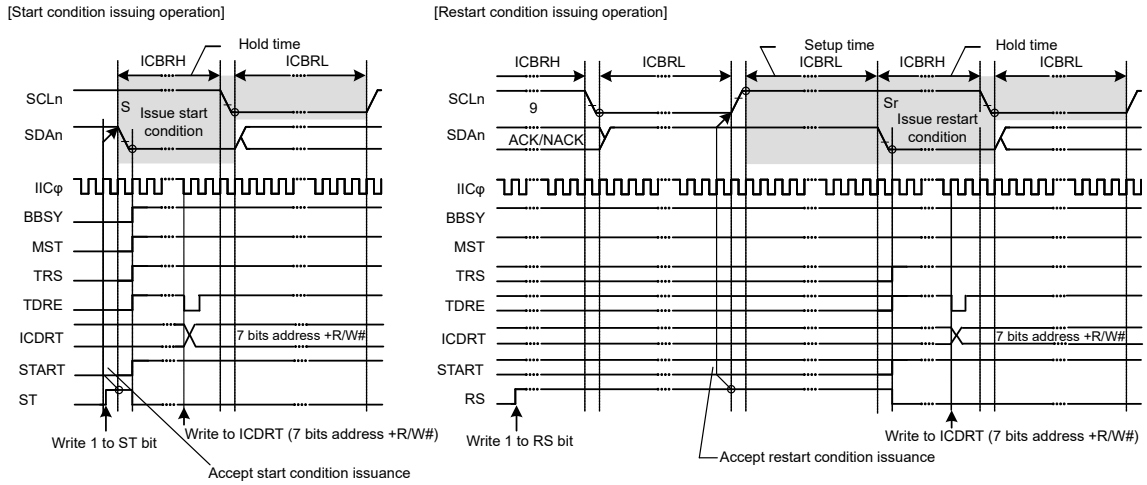


Figure 25.45 Start and restart condition issue timing using the ST and RS bits

Figure 25.46 shows the operation timing when a restart condition is issued after the master transmission.

[To issue a restart condition after the master transmission:]

1. Initialize the IIC using the procedure in section 25.3.2. Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is open, then set the ST bit in ICCR2 to 1 (start condition issuance request). On receiving the request, the IIC issues a start condition. At the same time, the BBSY and the START flags in ICSR2 are automatically set to 1 and the ST bit is automatically set to 0. If the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that a start condition is successfully issued as requested by the ST bit has been successfully completed. The MST and TRS bits in ICCR2 are automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 when the TRS bit is set to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. After the data for transmission is written to ICDRT, the TDRE flag is automatically set to 0, data is transferred from ICDRT to ICDRS, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode according to the value of the transmitted R/W# bit. If the value of the R/W# bit is 0, the IIC continues in master transmit mode. If the ICSR2.NACKF flag is 1 at this time, indicating that no slave device recognized the address or that there was an error in communications, write 1 to ICCR2.SP bit to issue a stop condition. To transmit data with an address in the 10-bit format, start by writing 1111 0b, the 2 upper bits of the slave address, and W to ICDRT as the first address transmission. Then, as the second address transmission, write the 8 lower bits of the slave address to ICDRT.
4. After confirming that the TDRE flag in ICSR2 is 1, write data for transmission to the ICDRT register. The IIC automatically holds the SCLn line low until data for transmission is ready, a restart condition is issued or a stop condition is issued.
5. After all bytes of data for transmission are written to the ICDRT register, wait until the value of the TEND flag in ICSR2 returns to 1. Then after checking that the START flag in ICSR2 is 1, set the START flag in ICSR2 to 0.
6. Set the RS bit in ICCR2 to 1 (restart condition issue request). On receiving the request, the IIC issues a restart condition.
7. After checking that the START flag in ICSR2 is 1, write the value for transmission (the slave address and the R/W# bit) to ICDRT.

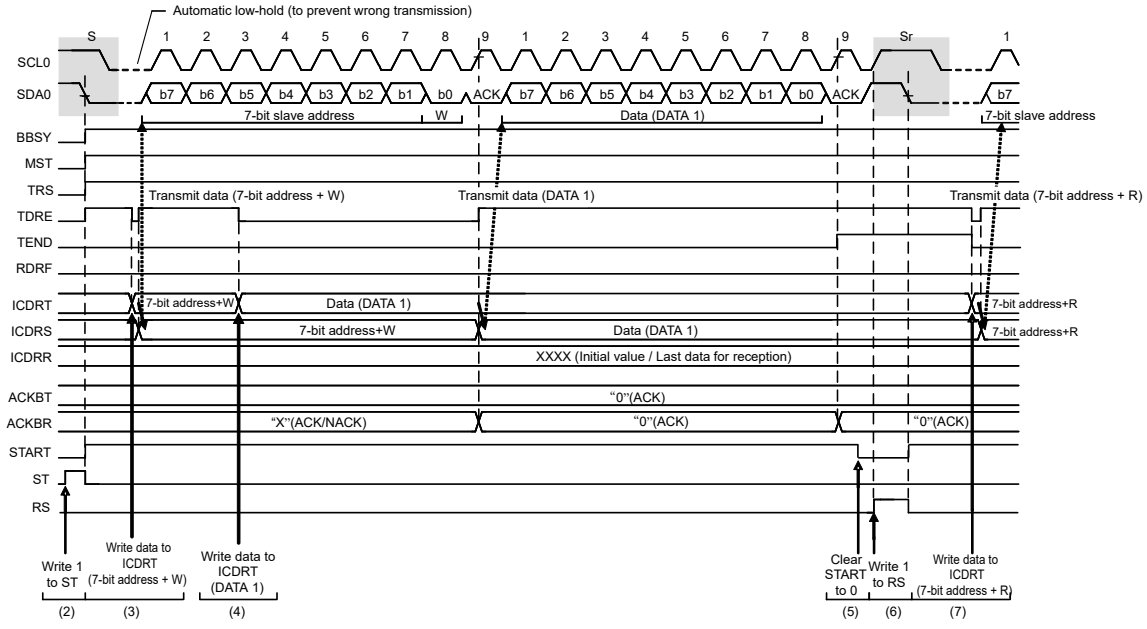


Figure 25.46 Restart condition issue timing after master transmission.

25.11.3 Issuing a Stop Condition

The IIC issues a stop condition when the SP bit in ICCR2 is set to 1. When the SP bit is set to 1, a stop condition request is made, and the IIC issues a stop condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a stop condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure the low-level period of the SCLn line set in ICBRL elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level on the SCLn line and ensure the time set in ICBRH and the stop condition setup time elapse.
5. Release the SDAn line (low level to high level).
6. Ensure the time set in ICBRL and the bus free time elapse.
7. Clear the BBSY flag to 0 to release the bus mastership.

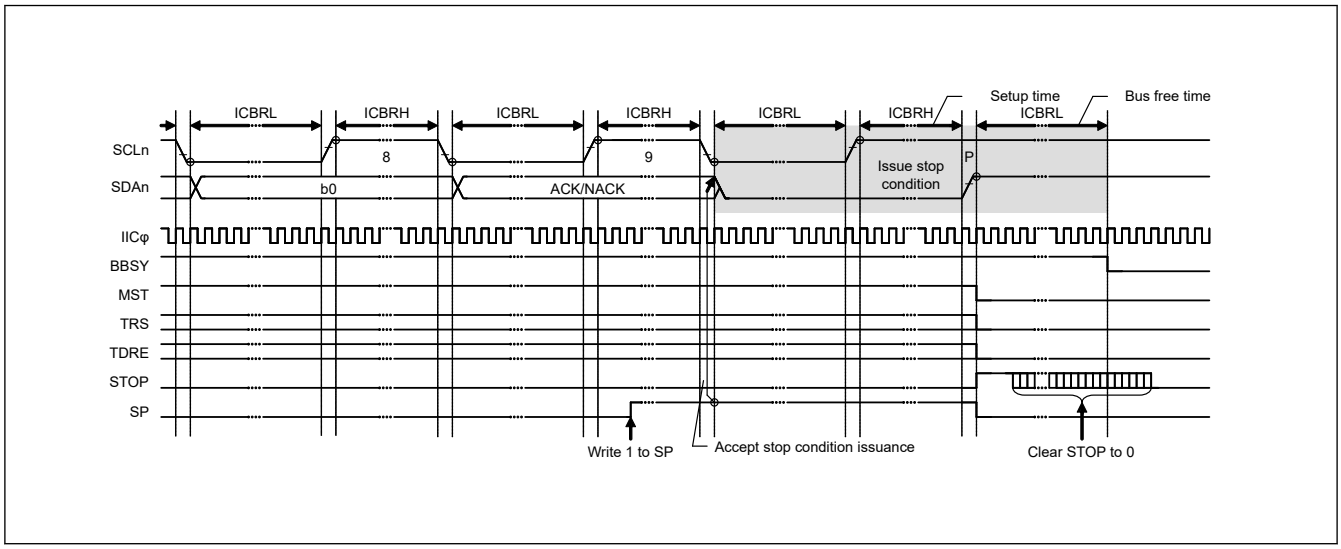


Figure 25.47 Stop condition issue timing using the SP bit

25.12 Bus Hanging

If the clock signals from the master and slave devices are out of synchronization because of noise or other factors, the I^2C bus might hang with a fixed level on the SCLn line or SDAn line.

To manage bus hanging, the IIC has a timeout function to detect hanging by monitoring the SCLn line, and a function for outputting an extra SCL clock cycle to release the bus from:

- A timeout function to detect hanging by monitoring the SCLn line
- The IIC reset function
- An internal reset function.

By checking the SCLO, SDAO, SCLI, and SDAI bits in ICCR1, it is possible to see whether the IIC or its communicating partner is placing the low level on the SCLn or SDAn line.

25.12.1 Timeout Function

The timeout function can detect when the SCLn line is stuck longer than the predetermined time. The IIC can detect an abnormal bus state by monitoring that the SCLn line is stuck low or high for a predetermined time.

The timeout function monitors the SCLn line state and counts the low- or high-level period using the internal counter. The timeout function resets the internal counter each time the SCLn line changes (rises or falls), but continues to count unless the SCLn line changes. If the internal counter overflows because no SCLn line changes, the IIC can detect the timeout and report the bus hung state.

This timeout function is enabled when the ICFER.TMOE bit is 1. It detects a hung state when the SCLn line is stuck low or high during the following conditions:

- The bus is busy (ICCR2.BBSY flag is 1) in master mode (ICCR2.MST bit is 1)
- The IIC slave address is detected (ICSR1 register is not 0x00) and the bus is busy (ICCR2.BBSY flag is 1) in slave mode (ICCR2.MST bit is 0)
- The bus is open (ICCR2.BBSY flag is 0) while a start condition is requested (ICCR2.ST bit is 1).

The internal counter of the timeout function uses the internal reference clock (IICφ) set in the CKS[2:0] bits in ICMR1 as a count source. It functions as a 16-bit counter when long mode is selected (TMOS bit = 0 in ICMR2) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCLn line level (low, high, or both levels) during which this counter is activated can be selected in the TMOH and TMOL bits in ICMR2. If both TMOL and TMOH bits are set to 0, the internal counter is disabled.

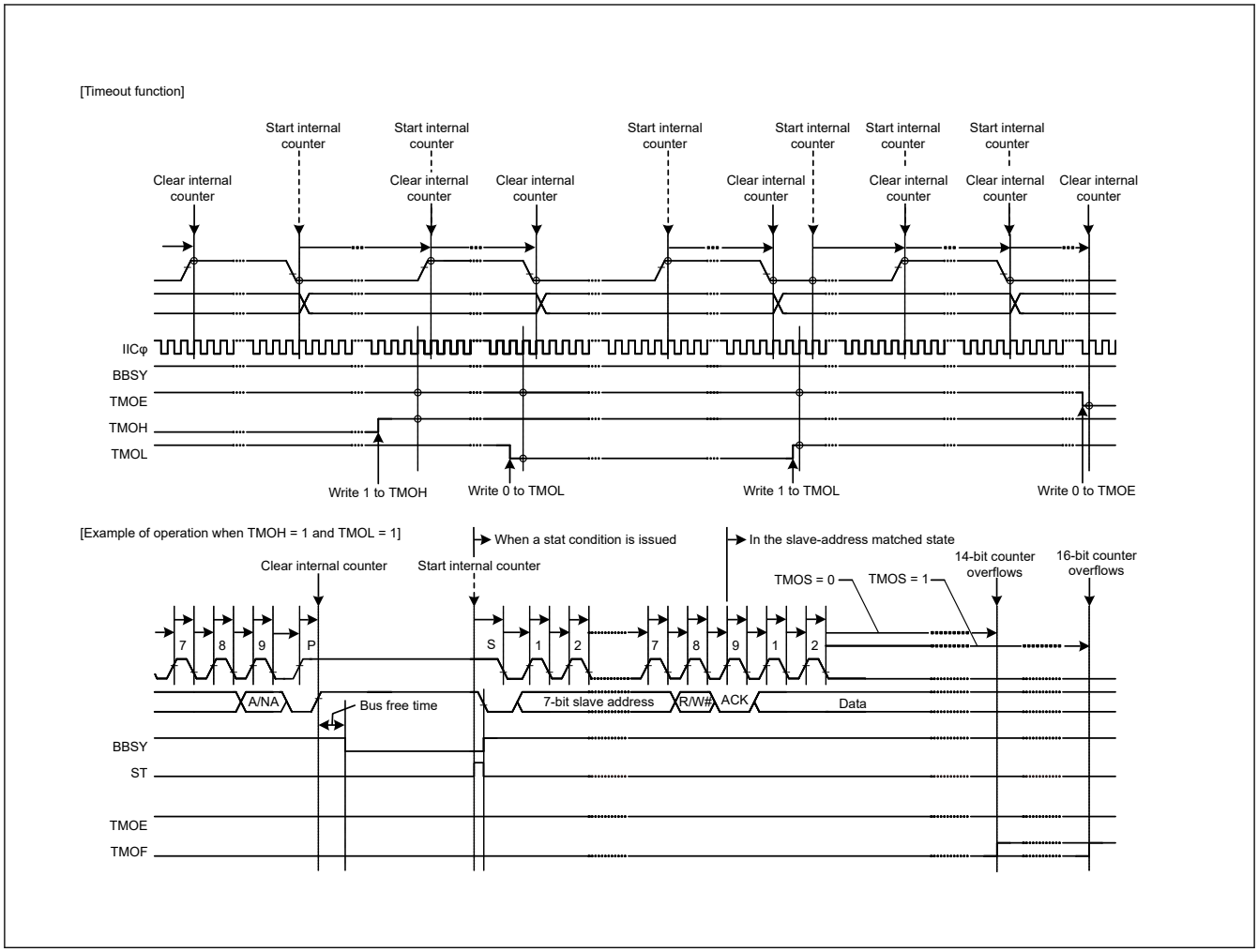


Figure 25.48 Timeout function using the TMOE, TMOS, TMOH, and TMOL bits

25.12.2 Extra SCL Clock Cycle Output Function

In master mode, this function outputs extra SCL clock cycles to release the SDAn line of the slave device from being held low because the master is out of synchronization with the slave device. This function is mainly used in master mode to release the SDAn line of the slave device from being fixed low by including extra cycles of SCL output from the IIC. It uses single cycles of the SCL clock for a bus error where the IIC cannot issue a stop condition because the slave device is holding the SDAn line at the low level. Do not use this function in normal situations. Using it when communications are proceeding correctly leads to malfunctions.

When the CLO bit in ICCR1 is set to 1 in master mode, a single cycle of the SCL clock at the frequency specified in the CKS[2:0] bits in ICMR1, and in the ICBRH and ICBRL registers, is output as an extra clock cycle. After output of this

single cycle of the SCL clock, the CLO bit is automatically set to 0. At this time, if ICCR2.BBSY = 1, the SCL pin goes low, and when ICCR2.BBSY = 0, the SCL pin goes high. After confirming that the CLO bit is 0 by software, write 1 to the CLO bit to output the additional clock continuously.

When the IIC module is in master mode and the slave device is holding the SDAn line low because synchronization with the slave device is lost because of effects like noise, the output of a stop condition is not possible. This function can be used to output extra cycles of SCL one by one to make the slave device release the SDAn line from being held low, and so recover the bus from an unusable state. Release of the SDAn line by the slave device can be monitored by reading the SDAI bit in ICCR1. After confirming the release of the SDAn line by the slave device, complete communications by reissuing the stop condition.

[Output conditions for using the CLO bit in ICCR1]

- When the bus is open (BBSY flag in ICCR2 = 0) or in master mode (MST bit = 1 and BBSY flag = 1 in ICCR2)
- When the communication device does not hold the SCLn line low.

Figure 25.49 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

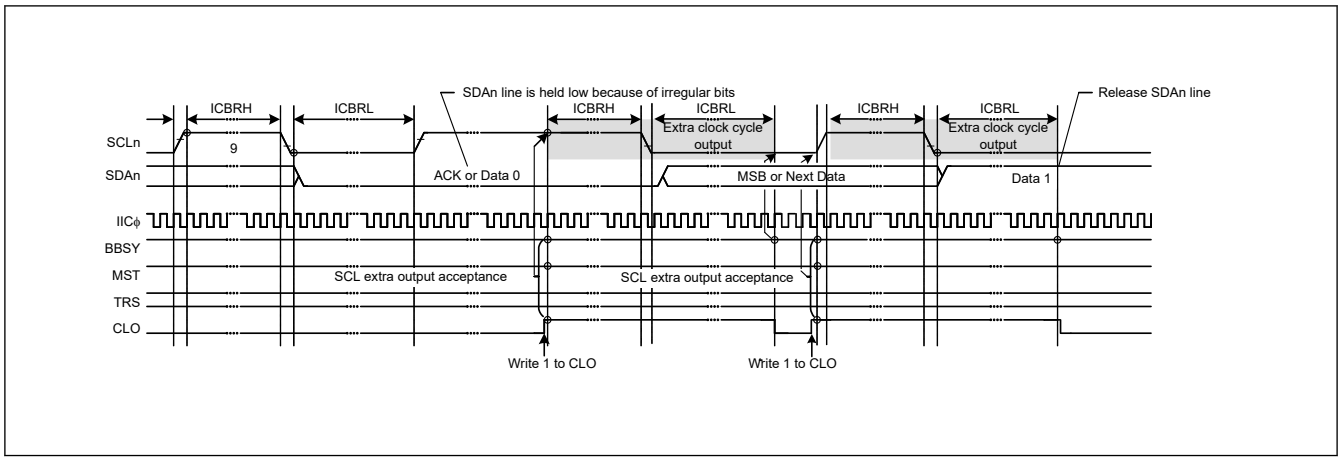


Figure 25.49 Extra SCL clock cycle output function using the CLO bit

25.12.3 IIC Reset and Internal Reset

The IIC module incorporates a function for resetting itself. It uses two types of resets:

- An IIC reset, which initializes all registers, including the BBSY flag in ICCR2.
- An internal reset, which releases the IIC from the slave-address matched state and initializes the internal counter while saving other settings.

After issuing a reset, always set the IICRST bit in ICCR1 to 0. Both types of resets are valid for release from bus-hung states, because both restore the output state of the SCLn and SDAn pins to the high-impedance state.

Issuing a reset during slave operation might lead to a loss of synchronization between the master device clock and the slave device clock, so avoid this when possible. In addition, monitoring of the bus state, such as for the presence of a start condition, is not possible during an IIC reset (ICE and IICRST bits = 01b in ICCR1).

For a detailed description of the IIC and internal resets, see section 25.15. State of Registers When Issuing Each Condition.

25.13 SMBus Operation

The IIC supports data communication conforming to the SMBus Specification (version 2.0). To perform SMBus communication, set the SMBS bit in ICMR3 to 1. To use the transfer rate within a range of 10 to 100 kbps of the SMBus standard, set the CKS[2:0] bits in ICMR1, the ICBRH, and ICBRL registers. In addition, specify the values in the DLCS bit in ICMR2 and the SDDL[2:0] bits in ICMR2 to meet the data hold time specification of 300 ns or more. When the IIC is used only as a slave device, the transfer rate setting is not required, but ICBRL must be set to a value longer than the data setup time (250 ns).

For the SMBus device default address (1100 001b), use one of the slave address registers L0 to L2 (SARL0, SARL1, and SARL2), and set the associated FS bit (7- or 10-bit address format select) in SARUy (y = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the SALE bit in ICFER to 1 to enable the slave arbitration-lost detection function.

25.13.1 SMBus Timeout Measurement

(1) Measuring slave device timeout

The following period (timeout interval: $T_{LOW:\ SEXT}$) must be measured for slave devices in SMBus communication:

- From start condition to stop condition.

To measure timeout for slave devices, measure the period from start condition detection to stop condition detection with the GPT using the IIC start condition detection interrupt (STIn) and stop condition detection interrupt (SPIn). The measured timeout period must be within the total clock low-level period [slave device] $T_{LOW:\ SEXT}$: 25 ms (maximum) of the SMBus standard.

If the time measured with the GPT exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (minimum) of the SMBus standard, the slave device must release the bus by writing 1 to the IICRST bit in ICCR1 to issue an internal reset of the IIC. When an internal reset is issued, the IIC stops driving the bus for the SCLn and SDAn pins, making them output high-impedance, which releases the bus.

(2) Measuring master device timeout

The following periods (timeout interval: $T_{LOW:\ MEXT}$) must be measured for master devices in SMBus communication:

- From start condition to acknowledge bit
- Between acknowledge bits
- From acknowledge bit to stop condition.

To measure timeout for master devices, measure these periods with the GPT using the IIC start condition detection interrupt (STIn), stop condition detection interrupt (SPIn), transmit end interrupt (IICn_TEI), or receive data full interrupt (IICn_RXI). The measured timeout period must be within the total clock low-level extended period (master device) $T_{LOW:\ MEXT}$: 10 ms (maximum) of the SMBus standard, and the total of all $T_{LOW:\ MEXT}$ values from start condition to stop condition must be within $T_{LOW:\ SEXT}$: 25 ms (maximum).

For the ACK receive timing (rising edge of the 9th SCL clock cycle), monitor the TEND flag in ICSR2 in master transmit mode (master transmitter) and the RDRF flag in ICSR2 in master receive mode (master receiver). Perform byte-wise transmit operations in master transmit mode, and hold the RDRFS bit in ICMR3 at 0 until the byte immediately before reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 on the rising edge of the 9th SCL clock cycle.

If the period measured with the GPT exceeds the total clock low-level extended period (master device) $T_{LOW:\ MEXT}$: 10 ms (maximum) of the SMBus standard or the total of measured periods exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (minimum) of the SMBus standard, the master device must stop the transaction by issuing a stop condition. In master transmit mode, immediately stop the transmit operation (stop writing data to ICDRT).

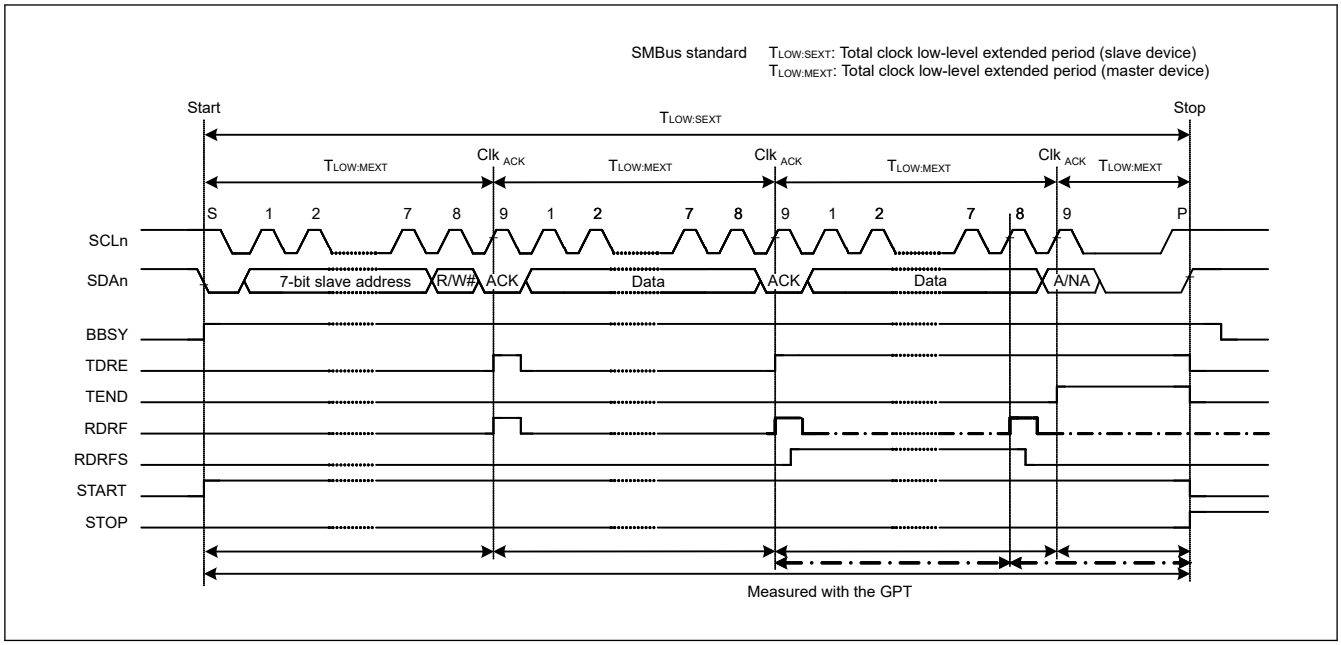


Figure 25.50 SMBus timeout measurement

25.13.2 Packet Error Code (PEC)

The MCU provides a CRC calculator that enables transmission of a Packet Error Code (PEC) or allows checking of the received data in SMBus data communication for the IIC. For the CRC-generating polynomials of the CRC calculator, see section 27, Cyclic Redundancy Check (CRC).

In master transmit mode, the PEC data can be generated by writing all transmit data to the CRC Data Input Register (CRCDIR) in the CRC calculator.

In master receive mode, the PEC data can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC Data Output Register (CRCDOR) with the received PEC data.

To send ACK or NACK based on the match or mismatch result when the final byte is received as a result of the PEC code check, set the RDRFS bit in ICMR3 to 1 before the rising edge of the 8th SCL clock cycle during reception of the final byte, and hold the SCLn line low on the falling edge of the 8th clock cycle.

25.13.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communicating over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address, or to request its own slave address from the SMBus host.

For a product using the MCU to operate as an SMBus host or ARP master, the host address (0001 000b) sent from the slave device must be detected as a slave address, and so the IIC provides a function for detecting the host address. To detect the host address as a slave address, set the SMBS bit in ICMR3 and the HOAE bit in ICSER to 1. Operation after the host address is detected is the same as normal slave operation.

25.14 Interrupt Sources

The IIC issues five types of interrupt requests:

- Transfer error or event generation (arbitration-lost, NACK detection, timeout detection, start condition detection, and stop condition detection)
- Receive data full
- Transmit data empty
- Transmit end
- Address match during wakeup function

Table 25.10 lists details about the interrupt requests. The receive data full and transmit data empty interrupts can activate data transfer by the DTC.

Table 25.10 Interrupt sources

Symbol	Interrupt source	Interrupt flag	DTC activation	Interrupt condition
IICn_EEI*5	Transfer error or event occurrence	AL	Not possible	AL = 1, ALIE = 1
		NACKF		NACKF = 1, NAKIE = 1
		TMOF		TMOF = 1, TMOIE = 1
		START		START = 1, STIE = 1
		STOP		STOP = 1, SPIE = 1
IICn_RXI*2 *5	Receive data full	RDRF	Possible	RDRF = 1, RIE = 1
IICn_TXI*1 *5	Transmit data empty	RDRF	Possible	TDRE = 1, TIE = 1
IICn_TEI*3 *5	Transmit end	TEND	Not possible	TEND = 1, TEIE = 1
IIC0_WUI*4	Slave address match during wakeup function	WUF	Not possible	Slave address match Slave receive complete RWAK operation ASY0 = 1 WUIE = 1

Note: There is a delay between the execution of a write instruction for a peripheral module by the CPU and the actual writing to the module. When an interrupt flag is cleared or masked, read the relevant flag again to check whether clearing or masking is complete, then return from interrupt handling. Not doing so creates the possibility of repeated processing of the same interrupt.
Note 1. Because IICn_TXI is edge-detected, it does not require clearing. Additionally, the TDRE flag in ICSR2 (condition for IICn_TXI) is automatically set to 0 when transmit data is written to the ICDRT register or a stop condition is detected (STOP flag = 1 in ICSR2).
Note 2. Because IICn_RXI is edge-detected, it does not require clearing. Additionally, the RDRF flag in ICSR2 (condition for IICn_RXI) is automatically set to 0 when data is read from ICDRR.
Note 3. When using the IICn_TEI interrupt, clear the TEND flag in ICSR2 in the IICn_TEI interrupt handling. The TEND flag in ICSR2 automatically is set to 0 when transmit data is written to the ICDRT register or a stop condition is detected (STOP flag = 1 in ICSR2).
Note 4. Only channel 0 has a wakeup function, so IIC0_WUI is for channel 0 only.
Note 5. Channel number (n = 0).

Clear or mask each flag during interrupt handling.

25.14.1 Buffer Operation for IICn_TXI and IICn_RXI Interrupts

If the conditions for generating an IICn_TXI or IICn_RXI interrupt are satisfied while the associated IR flag is 1, the interrupt request is not output for the ICU but is saved internally. One request per source can be saved internally.

An interrupt request that is saved in the ICU is output when the ICU.IELSRn.IR flag becomes 0. Internally saved interrupt requests are automatically cleared under normal conditions. They can also be cleared by writing 0 to the interrupt enable bit within the associated peripheral module.

25.15 State of Registers When Issuing Each Condition

The IIC has two dedicated resets, IIC reset and Internal reset. Table 25.11 lists the registers states when issuing each condition.

Table 25.11 Register states when issuing each condition (1 of 2)

Registers		Reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
ICCR1	ICE, IICRST	In reset	Saved	Saved	Saved	Saved
	SCLO, SDAO		In reset	In reset		
	Others			Saved		
ICCR2	BBSY	In reset	In reset	Saved	Set	In reset
	ST, RS			In reset	In reset	Saved
	SP					In reset
	TRS				Set or saved	
	MST					

Table 25.11 Register states when issuing each condition (2 of 2)

Registers		Reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
ICMR1	BC[2:0]	In reset	In reset	In reset	In reset	Saved
	Others			Saved	Saved	
ICMR2		In reset	In reset	Saved	Saved	Saved
ICMR3	ACKBT	In reset	In reset	Saved	Saved	In reset
	Others					Saved
ICFER		In reset	In reset	Saved	Saved	Saved
ICSER		In reset	In reset	Saved	Saved	Saved
ICIER		In reset	In reset	Saved	Saved	Saved
ICSR1		In reset	In reset	In reset	Saved	In reset
ICSR2	TEND	In reset	In reset	In reset	Saved	In reset
	TDRE				Set or saved	
	START				Set	
	STOP				Saved	Set
	Others				Saved	Saved
ICWUR		In reset	In reset	Saved	Saved	Saved
SARL0, SARL1, SARL2 SARU0, SARU1, SARU2		In reset	In reset	Saved	Saved	Saved
ICBRH, ICBRL		In reset	In reset	Saved	Saved	Saved
ICDRT		In reset	In reset	Saved	Saved	Saved
ICDRR		In reset	In reset	Saved	Saved	Saved
ICDRS		In reset	In reset	In reset	Saved	Saved
Timeout function		In reset	In reset	In reset	Operating	Operating
Bus free time measurement		In reset	In reset	Operating	Operating	Operating
ICWUR2	WUSEN	In reset	In reset	Saved	Saved	Saved
	Others					Saved or set or reset

25.16 Event Link Output

The IIC0 module handles the event output for the Event Link Controller (ELC) for the following sources:

(1) Transfer error event

When a transfer error event occurs, the associated event signal can be output to another module by the ELC.

(2) Receive data full

When a receive data register becomes full, the associated event signal can be output to another module by the ELC.

(3) Transmit data empty

When a transmit data register becomes empty, the associated event signal can be output to another module by the ELC.

(4) Transmit end

On completion of the transfer, the associated event signal can be output to another module by the ELC.

25.16.1 Interrupt Handling and Event Linking

Each of the IIC interrupt types (see Table 25.10) has an enable bit to control enabling and disabling of the associated interrupt signal. An interrupt request signal is output to the CPU when an interrupt source condition is satisfied while the associated enable bit is set.

The associated event link output signals are sent to other modules as event signals by the ELC when the interrupt source conditions are satisfied, regardless of the interrupt enable bit settings. For details on interrupt sources, see Table 25.10.

25.17 Usage Notes

25.17.1 Settings for the Module-Stop Function

The Module Stop Control Register B (MSTPCRB) can enable or disable IIC operation. The IIC is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

25.17.2 Notes on Starting Transfer

If the IR flag associated with the IIC interrupt is 1 when transfer is started (ICCR1.ICE bit = 1), follow the procedure in this section to clear the interrupts before enabling operations. Starting transfer with the IR flag set to 1 while the ICCR1.ICE bit is 1 leads to an interrupt request being internally saved after transfer starts, and this can lead to unexpected behavior of the IR flag.

To clear interrupts before starting transfer operation:

1. Confirm that the ICCR1.ICE bit is 0.
2. Set the relevant interrupt enable bits, such as ICIER.TIE to 0.
3. Read the relevant interrupt enable bits, such as ICIER.TIE, and confirm that the value is 0.
4. Set the IR flag to 0.

26. Serial Peripheral Interface (SPI)

26.1 Overview

The Serial Peripheral Interface (SPI) has 1 channel. The SPI provides high-speed full-duplex synchronous serial communications with multiple processors and peripheral devices. Table 26.1 lists the SPI specifications, Figure 26.1 shows a block diagram, and Table 26.2 lists the I/O pins.

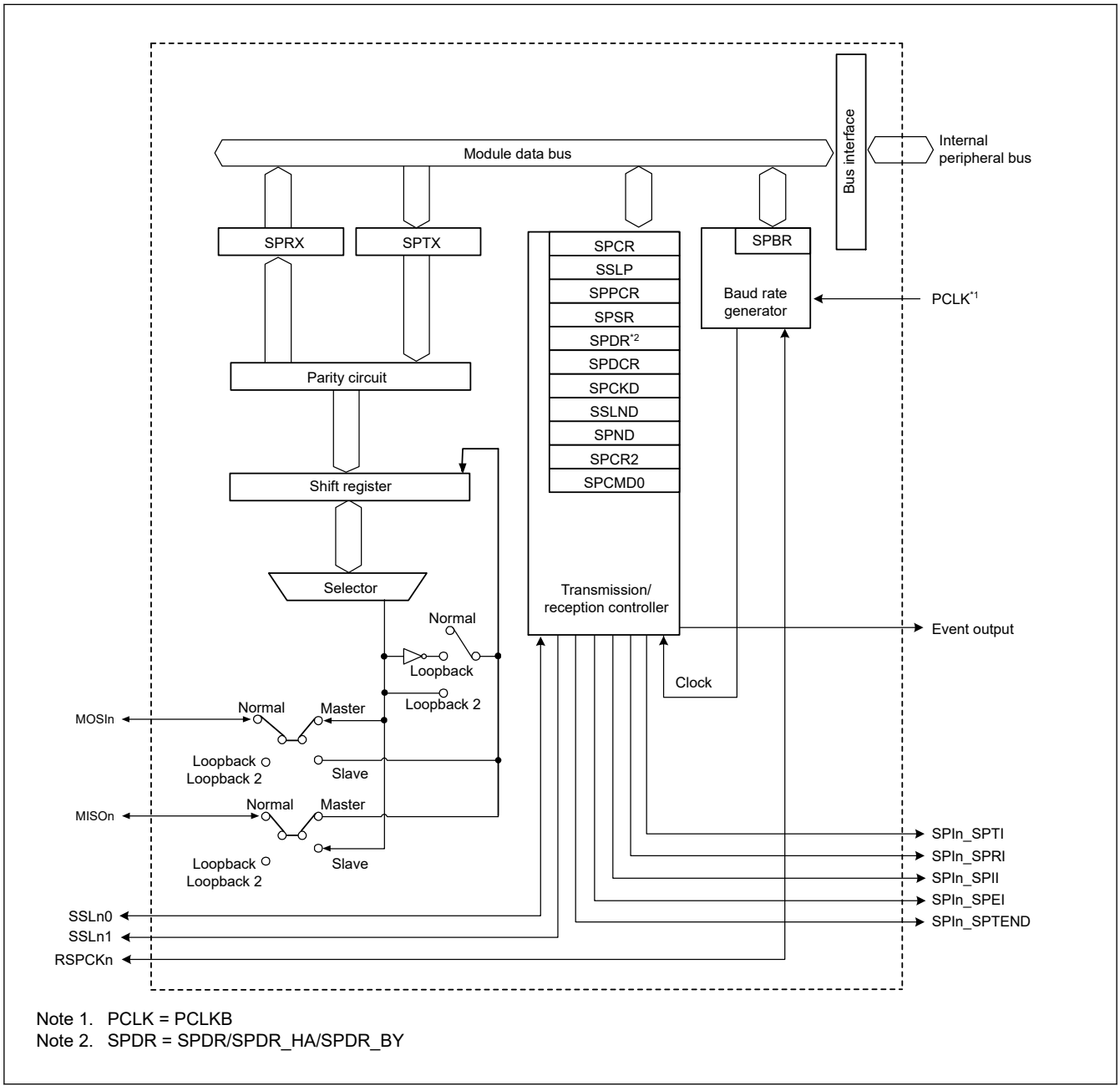
Table 26.1 SPI specifications (1 of 2)

Parameter	Specifications
Number of channels	One channel
SPI transfer functions	• Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (SPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method) • Transmit-only operation available • Communication mode selectable to full-duplex or transmit-only • RSPCK polarity switching • RSPCK phase switching
Data format	• MSB-first or LSB-first selectable • Transfer bit length selectable to 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits • 32-bit transmit and receive buffers
Bit rate	• In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing PCLKB (the division ratio ranges from divided by 2 to divided by 4096) • In slave mode, the minimum PCLKB clock divided by 6 can be input as RSPCK (PCLKB divided by 6 is the maximum RSPCK frequency) Width at high level: 3 PCLKB cycles; width at low level: 3 PCLKB cycles
Buffer configuration	• Double buffer configuration for the transmit and receive buffers • 32 bits for the transmit and receive buffers
Error detection	• Mode fault error detection • Underrun error detection • Overrun error detection*1 • Parity error detection
SSL control function	• Four SSL pins (SSLni: SSLn0, SSLn1) (n = A) for each channel • In single-master mode, SSLn0 and SSLn1 pins are output • In multi-master mode, SSLn0 pin for input and SSLn1 pin for output or unused • In slave mode, SSLn0 pin for input and SSLn1 pin unused • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Function for changing SSL polarity
Control in master transfer	• For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity and phase, transfer data length, MSB- or LSB-first, burst, RSPCK delay, SSL negation delay, and next-access delay • Transfers can be initiated by writing to the transmit buffer • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function
Interrupt sources	Interrupt sources: • Receive buffer full interrupt • Transmit buffer empty interrupt • SPI error interrupt (mode fault error, overrun error, parity error) • SPI idle interrupt (SPI idle) • Transmission-complete interrupt
Event link function	The following events can be output to the Event Link Controller (ELC): • Receive buffer full signal • Transmit buffer empty signal • Mode fault, underrun, overrun, or parity error signal • SPI idle signal • Transmission-complete signal

Table 26.1 SPI specifications (2 of 2)

Parameter	Specifications
Other functions	• Switching between CMOS output and open-drain output • SPI initialization function • Loopback mode
Module-stop function	Module-stop state can be set to reduce power consumption.

Note 1. In master reception and when the RSPCK auto-stop function is enabled, an overrun error does not occur because the transfer clock is stopped on overrun error detection.



Note 1. PCLK = PCLKB
Note 2. SPDR = SPDR/SPDR_HA/SPDR_BY

Figure 26.1 SPI block diagram

The SPI automatically switches the I/O direction of the SSLn0 pin. SSLn0 is set as an output when the SPI is a single master, and as an input when the SPI is a multi-master or a slave. The RSPCKn, MOSIn, and MISOn pins are automatically set as inputs or outputs based on the master or slave setting and the level input on the SSLn0 pin. For details, see section 26.3.2. Controlling the SPI Pins.

Table 26.2 SPI I/O pins

Channel	Pin name	I/O	Description
SPI0	RSPCKA	I/O	Clock input/output pin
	SSLA0	I/O	Slave selection input/output
	SSLA1	Output	Slave selection output
	MOSIA	I/O	Master transmit data input/output
	MISOA	I/O	Slave transmit data input/output

26.2 Register Descriptions

26.2.1 SPCR : SPI Control Register

Base address: SPI0 = 0x4007_2000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPMS	SPI Mode Select 0: Select SPI operation (4-wire method) 1: Select clock synchronous operation (3-wire method)	R/W
1	TXMD	Communications Operating Mode Select 0: Select full-duplex synchronous serial communications 1: Select serial communications with transmit-only	R/W
2	MODFEN	Mode Fault Error Detection Enable 0: Disable detection of mode fault errors 1: Enable detection of mode fault errors	R/W
3	MSTR	SPI Master/Slave Mode Select 0: Select slave mode 1: Select master mode	R/W
4	SPEIE	SPI Error Interrupt Enable 0: Disable SPI error interrupt requests 1: Enable SPI error interrupt requests	R/W
5	SPTIE	Transmit Buffer Empty Interrupt Enable 0: Disable transmit buffer empty interrupt requests 1: Enable transmit buffer empty interrupt requests	R/W
6	SPE	SPI Function Enable 0: Disable SPI function 1: Enable SPI function	R/W
7	SPRIE	SPI Receive Buffer Full Interrupt Enable 0: Disable SPI receive buffer full interrupt requests 1: Enable SPI receive buffer full interrupt requests	R/W

SPMS bit (SPI Mode Select)

The SPMS bit selects SPI operation (4-wire method) or clock synchronous operation (3-wire method).

The SSLn0 and SSLn1 pins are not used in clock synchronous operation. The RSPCKn, MOSIn, and MISOn pins handle communications. For clock synchronous operation in master mode (MSTR = 1), the SPCMD0.CPHA bit can be set to either 0 or 1. For clock synchronous operation in slave mode (MSTR = 0), always set the CPHA bit to 1. Do not perform operations if the CPHA bit is set to 0 for clock synchronous operation in slave mode (MSTR = 0).

TXMD bit (Communications Operating Mode Select)

The TXMD bit selects full-duplex synchronous serial communications or transmit-only operations. When this bit is set to 1, the SPI only performs transmit operations and not receive operations (see section 26.3.6. Data Transfer Modes), and receive buffer full interrupt requests cannot be used.

MODFEN bit (Mode Fault Error Detection Enable)

The MODFEN bit enables or disables the detection of mode fault errors (see section 26.3.8. Error Detection). In addition, the SPI determines the I/O direction of the SSLni pins based on combination of the MODFEN and MSTR bits (see section 26.3.2. Controlling the SPI Pins).

MSTR bit (SPI Master/Slave Mode Select)

The MSTR bit selects master or slave mode for the SPI. Based on the MSTR bit settings, the SPI determines the direction of the RSPCKn, MOSIn, MISOn, and SSLni pins.

SPEIE bit (SPI Error Interrupt Enable)

The SPEIE bit enables or disables the generation of SPI error interrupt requests when one of the following occurs:

- The SPI detects a mode fault error or underrun error and sets the SPSR.MODF flag to 1
- The SPI detects an overrun error and sets the SPSR.OVRF flag to 1
- The SPI detects a parity error and sets the SPSR.PERF flag to 1

For details, see section 26.3.8. Error Detection.

SPTIE bit (Transmit Buffer Empty Interrupt Enable)

The SPTIE bit enables or disables the generation of transmit buffer empty interrupt requests when the SPI detects that the transmit buffer is empty. To generate a transmit buffer empty interrupt request when transmission starts, set the SPE and SPTIE bits to 1 at the same time or set the SPE bit to 1 after setting the SPTIE bit to 1.

When the SPTIE bit is 1, transmit buffer interrupts are generated even when the SPI function is disabled (when the SPE bit is changed to 0).

SPE bit (SPI Function Enable)

The SPE bit enables or disables the SPI function. The SPE bit cannot be set to 1 when the SPSR.MODF flag is 1. For details, see section 26.3.8. Error Detection.

Setting the SPE bit to 0 disables the SPI function and initializes a part of the module function. For details, see section 26.3.9. Initializing the SPI. In addition, a transmit buffer empty interrupt request is generated when the SPE bit is changed from 0 to 1 or from 1 to 0.

SPRIE bit (SPI Receive Buffer Full Interrupt Enable)

The SPRIE bit enables or disables the generation of an SPI receive buffer full interrupt request when the SPI detects a receive buffer full write after completion of a serial transfer.

26.2.2 SSLP : SPI Slave Select Polarity Register

Base address: SPI0 = 0x4007_2000

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SSL1P	SSL0P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SSL0P	SSLn0 Signal Polarity Setting 0: Set SSLn0 signal to active-low 1: Set SSLn0 signal to active-high	R/W

Bit	Symbol	Function	R/W
1	SSL1P	SSLn1 Signal Polarity Setting 0: Set SSLn1 signal to active-low 1: Set SSLn1 signal to active-high	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

26.2.3 SPPCR : SPI Pin Control Register

Base address: SPI0 = 0x4007_2000

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPLP	SPI Loopback 0: Normal mode 1: Loopback mode (receive data = inverted transmit data)	R/W
1	SPLP2	SPI Loopback 2 0: Normal mode 1: Loopback mode (receive data = transmit data)	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	MOIFV	MOSI Idle Fixed Value 0: Set level output on MOSIn pin during MOSI idling to low 1: Set level output on MOSIn pin during MOSI idling to high	R/W
5	MOIFE	MOSI Idle Value Fixing Enable 0: Set MOSI output value to equal final data from previous transfer 1: Set MOSI output value to equal value set in the MOIFV bit	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

SPLP bit (SPI Loopback)

The SPLP bit selects the mode of the SPI pins. When this bit is set to 1, the SPI shuts off the path between the MISOn pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIn pin and the shift register if the SPCR.MSTR bit is 0. The SPI then inverts the value of the input path for the shift register and connects it to the output path (loopback mode). For more information, see section 26.3.12. Loopback Mode.

SPLP2 bit (SPI Loopback 2)

The SPLP2 bit selects the mode of the SPI pins. When this bit is set to 1, the SPI shuts off the path between the MISOn pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIn pin and the shift register if the SPCR.MSTR bit is 0. The SPI then connects the value of the input path for the shift register to the output path (loopback mode) without inverting the value. For more information, see section 26.3.12. Loopback Mode.

MOIFV bit (MOSI Idle Fixed Value)

The MOIFV bit determines the MOSIn pin output value during the SSL negation period when the MOIFE bit is 1 in master mode.

MOIFE bit (MOSI Idle Value Fixing Enable)

The MOIFE bit fixes the MOSIn output value when the SPI is in master mode and in an SSL negation period.When the MOIFE bit is 0, the SPI outputs the last data from the previous serial transfer during the SSL negation period to the MOSIn pin. When the MOIFE bit is 1, the SPI outputs the fixed value set in the MOIFV bit to the MOSIn pin.

26.2.4 SPSR : SPI Status Register

Base address: SPI0 = 0x4007_2000

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SPRF	—	SPTEF	UDRF	PERF	MODF	IDLNF	OVRF
Value after reset:	0	0	1	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OVRF	Overrun Error Flag 0: No overrun error occurred 1: Overrun error occurred	R/W*1
1	IDLNF	SPI Idle Flag 0: SPI is in the idle state 1: SPI is in the transfer state	R
2	MODF	Mode Fault Error Flag 0: No mode fault or underrun error occurred 1: Mode fault error or underrun error occurred	R/W*1
3	PERF	Parity Error Flag 0: No parity error occurred 1: Parity error occurred	R/W*1
4	UDRF	Underrun Error Flag The UDRF bit is valid when MODF flag is 1. 0: Mode fault error occurred (MODF = 1) 1: Underrun error occurred (MODF = 1)	R/W*1 *2
5	SPTEF	SPI Transmit Buffer Empty Flag 0: Data is in the transmit buffer 1: No data is in the transmit buffer	R/W*3
6	—	This bit is read as 0. The write value should be 0.	R/W
7	SPRF	SPI Receive Buffer Full Flag 0: No valid data is in SPDR/SPDR_HA 1: Valid data is in SPDR/SPDR_HA	R/W*3

Note 1. Only 0 can be written to clear the flag after reading 1.
Note 2. Clear the UDRF flag at the same time as the MODF flag.
Note 3. The write value should be 1.

OVRF flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error. In master mode (SPCR.MSTR bit = 1) and when the RSPCK clock auto-stop function is enabled (SPCR1.SCKASE bit = 1), overrun errors do not occur. This flag does not set to 1. For details, see section 26.3.8.1. Overrun errors.

[Setting condition]

- When the next serial transfer ends while the SPCR.TXMD bit is 0 and the receive buffer is full.

[Clearing condition]

- When 0 is written to the OVRF flag after the OVRF flag is confirmed to be 1 by a read of SPSR.

IDLNF flag (SPI Idle Flag)

The IDLNF flag indicates the transfer status of the SPI.

[Setting conditions]

Master mode

- When none of the conditions in the master mode [Clearing condition] is met.

Slave mode

- When the SPE bit in SPCR is 1, enabling the SPI function.

[Clearing conditions]

Master mode

When condition 1 or all other conditions are satisfied.

Condition 1: The SPE bit in SPCR is 0, indicating that the SPI is initialized.

Condition 2: The transmit buffer (SPTX) is empty, indicating that data for the next transfer is not set.

Condition 3: The SPI internal sequencer is in the idle state, indicating that operation up to next-access delay is complete.

Slave mode

- When condition 1 is satisfied.

MODF flag (Mode Fault Error Flag)

The MODF flag indicates the occurrence of a mode fault error or an underrun error. The UDRF flag indicates which error occurred.

[Setting conditions]

Multi-master mode

- When the input level of the SSLni pin changes to the active level while the SPCR.MSTR bit is 1 (master mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), triggering a mode fault error.

Slave mode

- When condition 1 or 2 is satisfied.

Condition 1: The SSLni pin is negated before the RSPCK cycle required for data transfer ends while the SPCR.MSTR bit is 0 (slave mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), triggering a mode fault error.

Condition 2: The serial transfer begins with the SPCR.MSTR bit is set to 0 (slave mode), the SPCR.SPE bit is set to 1, and the transmission data not prepared, triggering an underrun error.

The active level of the SSLni signal is determined by the SSLP.SSLiP bit (SSLi signal polarity setting).

[Clearing condition]

- When SPSR is read while this flag is 1, and then 0 is written to this flag.

PERF flag (Parity Error Flag)

The PERF flag indicates the occurrence of a parity error.

[Setting condition]

- When a serial transfer ends while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1, triggering a parity error.

[Clearing condition]

- When SPSR is read while this flag is 1, and then 0 is written to this flag.

UDRF flag (Underrun Error Flag)

The UDRF flag indicates the occurrence of an underrun error.

[Setting condition]

- When the serial transfer begins with the SPCR.MSTR bit is set to 0 (slave mode), the SPCR.SPE bit is set to 1, and the transmission data not prepared, triggering an underrun error.

[Clearing condition]

- When SPSR is read while this flag is 1, and then 0 is written to this flag.

SPTEF flag (SPI Transmit Buffer Empty Flag)

The SPTEF flag indicates the status of the transmit buffer for the SPI Data Register (SPDR/SPDR_HA).

[Setting conditions]

- When condition 1. or 2. is satisfied.
  1. The SPCR.SPE bit is 0, indicating that the SPI is initialized.
  2. Transmit data is transferred from the transmit buffer to the shift register.

[Clearing condition]

- When data written to SPDR/SPDR_HA/SPDR_BY.

Data can only be written to SPDR/SPDR_HA/SPDR_BY when the SPTEF flag is 1. If data is written to the transmit buffer of SPDR/SPDR_HA when the SPTEF flag is 0, data in the transmit buffer is not updated.

SPRF flag (SPI Receive Buffer Full Flag)

The SPRF flag indicates the status of the receive buffer for the SPI Data Register (SPDR/SPDR_HA).

[Setting condition]

- Received data have been transferred to the SPDR from the shift register while the SPRF flag is 0. And satisfy the following. However, the SPRF flag does not change from 0 to 1 while the OVRF flag = 1.
  - The SPCR.TXMD bit is 0 (transmit-receive master mode, transmit-receive slave mode)

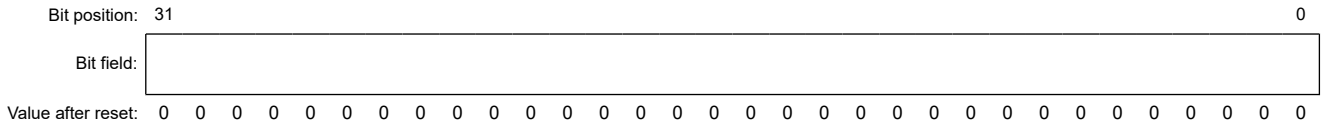
[Clearing condition]

- When received data is read from the SPDR/SPDR_HA.

26.2.5 SPDR/SPDR_HA/SPDR_BY : SPI Data Register

Base address: SPI0 = 0x4007_2000

Offset address: 0x04



Bit	Symbol	Function	R/W
31:0	n/a	SPI Data	R/W

SPDR/SPDR_HA/SPDR_BY is the interface with the buffers that hold data for transmission and reception by the SPI. When accessing this register in words (the SPDCR.SPLW bit is 1), access SPDR. When accessing it in halfwords (the SPLW bit is 0), access SPDR_HA. When accessing it in byte (the SPDCR.SPBYT bit is 1), access SPDR_BY.

The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to SPDR/SPDR_HA. Figure 26.2 shows the configuration of the SPDR/SPDR_HA register.

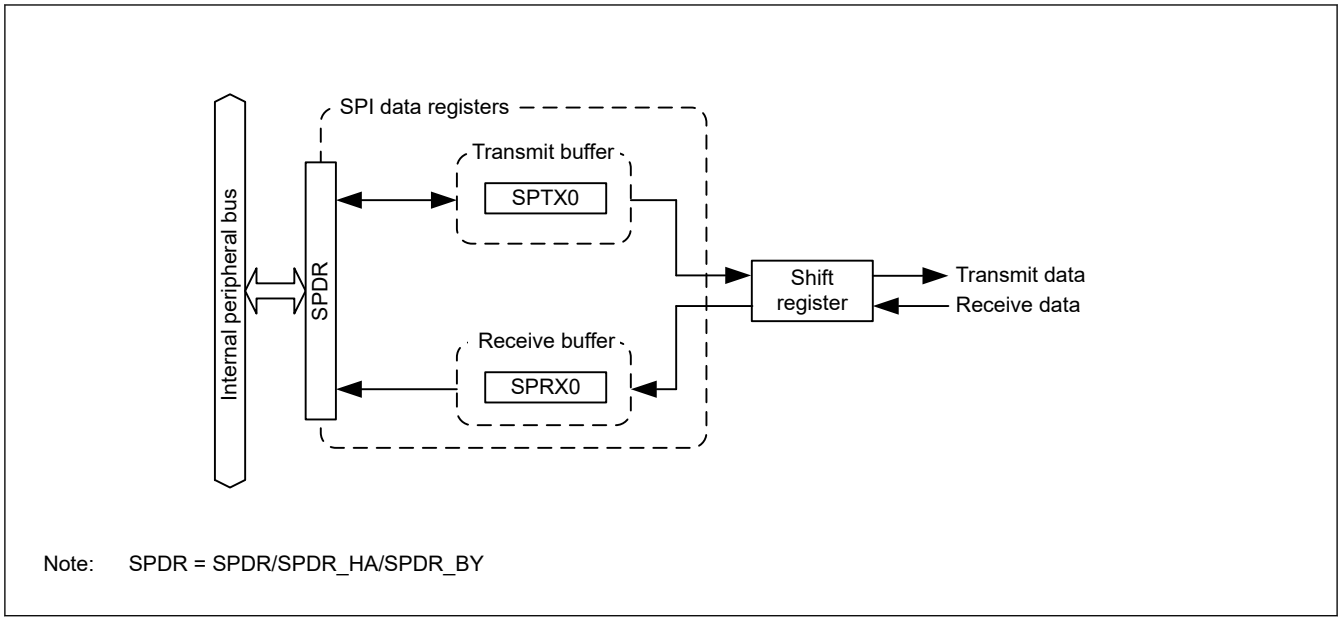


Figure 26.2 Configuration of SPDR/SPDR_HA/SPDR_BY

The transmit and receive buffers each have one stage. The two stages of the buffer are all mapped to the single address of SPDR/SPDR_HA/SPDR_BY.

Data written to SPDR/SPDR_HA/SPDR_BY is written to a transmit-buffer stage (SPTXn) (n = 0), and then transmitted from the buffer. The receive buffer holds received data on completion of reception. The receive buffer is not updated if an overrun is generated.

Additionally, if the data length is not 32 bits, bits not referred to in SPTXn (n = 0) are stored in the associated bits in SPRXn (n = 0). For example, if the data length is 9 bits, the received data is stored in the SPRXn[8:0] bits, and the SPTXn[31:9] bits are stored in the SPRXn[31:9] bits.

(1) Bus interface

SPDR/SPDR_HA/SPDR_BY is an interface with 32-bit wide transmit and receive buffers, each of which has one stage stages, for a total of 8 bytes. The 8 bytes are mapped to the 4-byte address space for SPDR/SPDR_HA/SPDR_BY. Additionally, the unit of access for SPDR/SPDR_HA/SPDR_BY is selected by the SPI Word Access/Halfword Access Specification bit in the SPI Data Control Register (SPDCR.SPLW). SPDR can also be accessed with the access size specified by the SPI Byte Access bit in the SPI Data Control Register (SPDCR.SPBYT).

Flush the transmission data at the LSB end of the register, and store the received data at the LSB end.

The following sections describe the operations involved in writing to and reading from SPDR/SPDR_HA/SPDR_BY.

Writing

Data written to SPDR/SPDR_HA/SPDR_BY is written to a transmit buffer (SPTXn). This is not affected by the value of the SPDCR.SPRDTD bit, unlike when reading from SPDR/SPDR_HA/SPDR_BY.

Figure 26.3 shows the configuration of the bus interface with the transmit buffer when writing to SPDR/SPDR_HA/SPDR_BY.

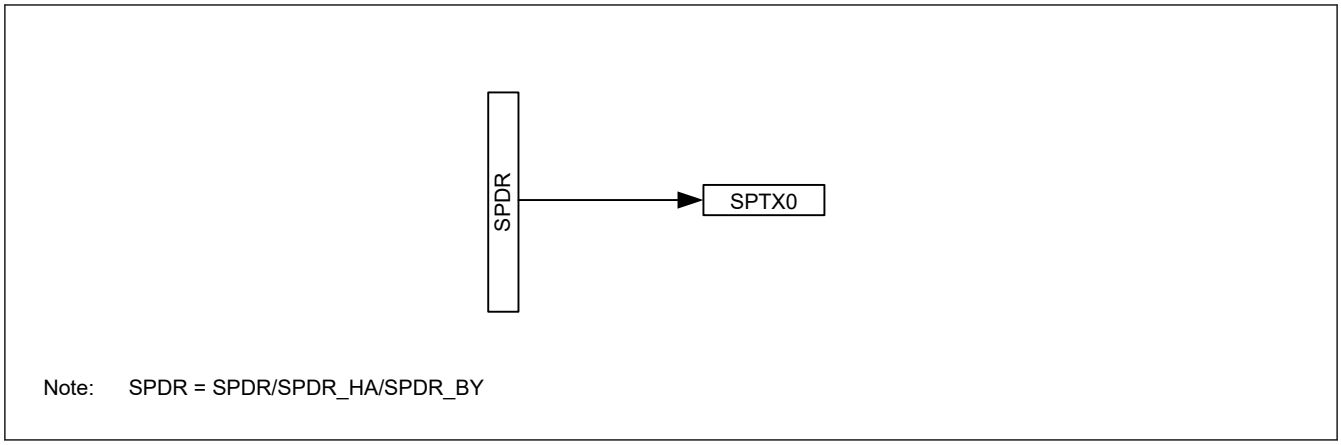


Figure 26.3 Configuration of SPDR/SPDR_HA/SPDR_BY for write access

Even when the specified number of frames is written to the transmit buffer (SPTXn), the value of the buffer is not updated after completion of the writing and before generation of the next transmit buffer empty interrupt (when SPTEF is 0).

Reading

SPDR/SPDR_HA/SPDR_BY can be accessed to read the value of a receive buffer (SPRXn) or a transmit buffer (SPTXn). The setting in the SPI Receive/Transmit Data Select bit in the SPI Data Control Register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer.

Figure 26.4 shows the configuration of the bus interface with the receive and transmit buffers for reading from SPDR/SPDR_HA/SPDR_BY.

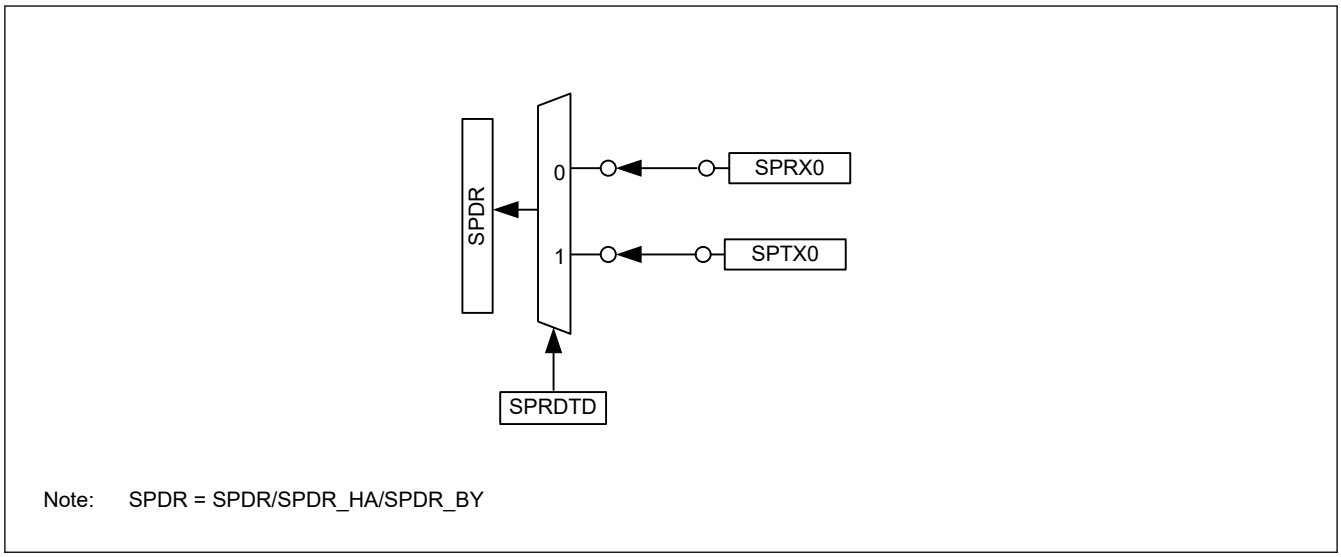


Figure 26.4 Configuration of SPDR/SPDR_HA/SPDR_BY for read access

After a transmit buffer empty interrupt is generated, reading from the transmit buffer returns all 0s after the completion of writing the data, until the next buffer empty interrupt is generated (when SPTEF is 0).

26.2.6 SPBR : SPI Bit Rate Register

Base address: SPI0 = 0x4007_2000

Offset address: 0x0A

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

Bit	Symbol	Function	R/W
7:0	n/a	Bit rate	R/W

SPBR sets the bit rate in master mode.

When the SPI is in slave mode, the bit rate depends on the bit rate of the input clock, regardless of the settings in SPBR and the SPCMD0.BRDV[1:0] bits (bit rate division setting). Use bit rates that satisfy the electrical characteristics of the device.

The bit rate is determined by combinations of the SPBR and SPCMD0.BRDV[1:0] settings in the SPI Command Register. The equation for calculating the bit rate is given as follows:

$$\text{Bit rate} = \frac{f(\text{PCLK})}{2 \times (n+1) \times 2^{N}}$$

(PCLK = PCLKB)

In the equation, n denotes an SPBR setting (0, 1, 2, …, 255), and N denotes a BRDV[1:0] setting (0, 1, 2, 3).

Table 26.3 lists examples of the relationship between the SPBR settings, the BRDV[1:0] settings, and bit rates.

Table 26.3 Relationship between SPBR settings, BRDV[1:0] settings, and bit rates

SPBR (n)	BRDV[1:0] bits (N)	Division ratio	Bit rate
			PCLKB = 32 MHz
0	0	2	16.0 Mbps
1	0	4	8.00 Mbps
2	0	6	5.33 Mbps
3	0	8	4.00 Mbps
4	0	10	3.20 Mbps
5	0	12	2.67 Mbps
5	1	24	1.33 Mbps
5	2	48	667 kbps
5	3	96	333 kbps
255	3	4096	7.81 kbps

26.2.7 SPDCR : SPI Data Control Register

Base address: SPI0 = 0x4007_2000

Offset address: 0x0B

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

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	SPRDTD	SPI Receive/Transmit Data Select 0: Read SPDR/SPDR_HA values from receive buffer 1: Read SPDR/SPDR_HA values from transmit buffer, but only if the transmit buffer is empty	R/W
5	SPLW	SPI Word Access/Halfword Access Specification 0: Set SPDR_HA to valid for halfword access 1: Set SPDR to valid for word access	R/W
6	SPBYT	SPI Byte Access Specification 0: SPDR/SPDR_HA is accessed in halfword or word (SPLW is valid) 1: SPDR_BY is accessed in byte (SPLW is invalid)	R/W

Bit	Symbol	Function	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

The SPI data control register (SPDCR) is used to read the SPDR register, and to set the access width for the SPDR register to word access, halfword access, or byte access.

SPRDTD bit (SPI Receive/Transmit Data Select)

The SPRDTD bit selects whether the SPDR/SPDR_HA reads values from the receive buffer or from the transmit buffer. If reading is from the transmit buffer, the last value written to SPDR/SPDR_HA register is read. Read the transmit buffer after an SPI transmit buffer empty interrupt is generated.

For details, see section 26.2.5. SPDR/SPDR_HA/SPDR_BY : SPI Data Register.

SPLW bit (SPI Word Access/Halfword Access Specification)

The SPLW bit specifies the access width for SPDR. Access to SPDR_HA in halfwords is valid when the SPLW bit is 0 and access to SPDR in words is valid when the SPLW bit is 1. Also, when this bit is 0, set the SPI data length setting bits, SPCMD0.SPB[3:0], from 8 to 16 bits. Do not perform any operations when a data length of 20, 24, or 32 bits is specified.

SPBYT bit (SPI Byte Access Specification)

The SPBYT bit is used to set the data width of access to the SPI Data Register (SPDR). When SPBYT = 0, use word or half word access to SPDR/SPDR_HA. When SPBYT = 1 (in that case, SPLW is invalid), use byte access to SPDR_BY.

When SPBYT = 1, set the SPI data length bits (SPB[3:0]) in the SPI Command Register 0 (SPCMD0) to 8 bits. If SPB[3:0] are set to 9 to 16, 20, 24, or 32 bit, subsequent operation is not guaranteed.

26.2.8 SPCKD : SPI Clock Delay Register

Base address: SPI0 = 0x4007_2000

Offset address: 0x0C

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

Bit	Symbol	Function	R/W
2:0	SCKDL[2:0]	RSPCK Delay Setting 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

SPCKD specifies the RSPCK delay, the period from the beginning of SSLni signal assertion to RSPCK oscillation, when the SPCMD0.SCKDEN bit is 1.

SCKDL[2:0] bits (RSPCK Delay Setting)

The SCKDL[2:0] bits specify an RSPCK delay value when the SPCMD0.SCKDEN bit is 1. When using the SPI in slave mode, set the SCKDL[2:0] bits to 000b.

26.2.9 SSLND : SPI Slave Select Negation Delay Register

Base address: SPI0 = 0x4007_2000

Offset address: 0x0D

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

Bit	Symbol	Function	R/W
2:0	SLNDL[2:0]	SSL Negation Delay Setting 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

SSLND specifies the SSL negation delay, the period from the transmission of a final RSPCK edge to the negation of the SSLni signal during a serial transfer by the SPI in master mode. If the contents of SSLND are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, do not perform subsequent operations.

SLNDL[2:0] bits (SSL Negation Delay Setting)

The SLNDL[2:0] bits specify an SSL negation delay value when the SLNDEN bit in SPCMDn is 1 and the SPI is in master mode. When using the SPI in slave mode, set the SLNDL[2:0] bits to 000b.

26.2.10 SPND : SPI Next-Access Delay Register

Base address: SPI0 = 0x4007_2000

Offset address: 0x0E

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

Bit	Symbol	Function	R/W
2:0	SPNDL[2:0]	SPI Next-Access Delay Setting 0 0 0: 1 RSPCK + 2 PCLKB 0 0 1: 2 RSPCK + 2 PCLKB 0 1 0: 3 RSPCK + 2 PCLKB 0 1 1: 4 RSPCK + 2 PCLKB 1 0 0: 5 RSPCK + 2 PCLKB 1 0 1: 6 RSPCK + 2 PCLKB 1 1 0: 7 RSPCK + 2 PCLKB 1 1 1: 8 RSPCK + 2 PCLKB	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

SPND specifies the next-access delay, the non-active period of the SSLni signal after termination of a serial transfer, when the SPCMD0.SPNDEN bit is 1.

SPNDL[2:0] bits (SPI Next-Access Delay Setting)

The SPNDL[2:0] bits specify a next-access delay when the SPCMD0.SPNDEN bit is 1. When using the SPI in slave mode, set the SPNDL[2:0] bits to 000b.

26.2.11 SPCR2 : SPI Control Register 2

Base address: SPI0 = 0x4007_2000

Offset address: 0x0F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SCKASE	PTE	SPIIE	SPOE	SPPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPPE	Parity Enable 0: Do not add parity bit to transmit data and do not check parity bit of receive data 1: When SPCR.TXMD = 0: Add parity bit to transmit data and check parity bit of receive data When SPCR.TXMD = 1: Add parity bit to transmit data but do not check parity bit of receive data	R/W
1	SPOE	Parity Mode 0: Select even parity for transmission and reception 1: Select odd parity for transmission and reception	R/W
2	SPIIE	SPI Idle Interrupt Enable 0: Disable idle interrupt requests 1: Enable idle interrupt requests	R/W
3	PTE	Parity Self-Testing 0: Disable self-diagnosis function of the parity circuit 1: Enable self-diagnosis function of the parity circuit	R/W
4	SCKASE	RSPCK Auto-Stop Function Enable 0: Disable RSPCK auto-stop function 1: Enable RSPCK auto-stop function	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

SPPE bit (Parity Enable)

The SPPE bit enables or disables the parity function.

When the SPCR.TXMD bit is 0 and this bit is 1, the parity bit is added to transmit data and parity checking is performed for receive data.

When the SPCR.TXMD bit is 1 and this bit is 1, the parity bit is added to transmit data but parity checking is not performed for receive data.

SPOE bit (Parity Mode)

The SPOE bit specifies odd or even parity.

When even parity is set, parity bit addition is performed so that the total number of bits whose value is 1 in the transmit or receive character plus the parity bit is even. Similarly, when odd parity is set, parity bit addition is performed so that the total number of bits whose value is 1 in the transmit or receive character plus the parity bit is odd.

The SPOE bit is only valid when the SPPE bit is 1.

SPIIE bit (SPI Idle Interrupt Enable)

The SPIIE bit enables or disables the generation of SPI idle interrupt requests when an idle state is detected in the SPI and the SPSR.IDLNF flag clears is set to 0.

PTE bit (Parity Self-Testing)

The PTE bit enables self-diagnosis of the parity circuit to check whether the parity function is operating correctly.

SCKASE bit (RSPCK Auto-Stop Function Enable)

The SCKASE bit enables or disables the RSPCK auto-stop function. When this function is enabled, the RSPCK clock is stopped before an overrun error occurs, when data is received in master mode. For details, see section 26.3.8.1. Overrun errors.

26.2.12 SPCMD0 : SPI Command Register 0

Base address: SPI0 = 0x4007_2000

Offset address: 0x10

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]				—	SSLA[2:0]			BRDV[1:0]		CPOL	CPHA
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1

Bit	Symbol	Function	R/W
0	CPHA	RSPCK Phase Setting 0: Select data sampling on leading edge, data change on trailing edge 1: Select data change on leading edge, data sampling on trailing edge	R/W
1	CPOL	RSPCK Polarity Setting 0: Set RSPCK low during idle 1: Set RSPCK high during idle	R/W
3:2	BRDV[1:0]	Bit Rate Division Setting 0 0: Base bit rate 0 1: Base bit rate divided by 2 1 0: Base bit rate divided by 4 1 1: Base bit rate divided by 8	R/W
6:4	SSLA[2:0]	SSL Signal Assertion Setting 0 0 0: SSL0 0 0 1: SSL1 Others: Setting prohibited	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
11:8	SPB[3:0]	SPI Data Length Setting 0x0: 20 bits 0x1: 24 bits 0x2: 32 bits 0x3: 32 bits 0x8: 9 bits 0x9: 10 bits 0xA: 11 bits 0xB: 12 bits 0xC: 13 bits 0xD: 14 bits 0xE: 15 bits 0xF: 16 bits Others: 8 bits	R/W
12	LSBF	SPI LSB First 0: MSB-first 1: LSB-first	R/W
13	SPNDEN	SPI Next-Access Delay Enable 0: Select next-access delay of 1 RSPCK + 2 PCLKB 1: Select next-access delay equal to the setting in the SPI Next-Access Delay Register (SPND)	R/W
14	SLNDEN	SSL Negation Delay Setting Enable 0: Select SSL negation delay of 1 RSPCK 1: Select SSL negation delay equal to the setting in the SPI Slave Select Negation Delay Register (SSLND)	R/W

Bit	Symbol	Function	R/W
15	SCKDEN	RSPCK Delay Setting Enable 0: Select RSPCK delay of 1 RSPCK 1: Select RSPCK delay equal to the setting in the SPI Clock Delay Register (SPCKD)	R/W

The SPCMD0 register sets the transfer format for the SPI in master mode.

Set this register while the transmit buffer is empty (SPSR.SPTEF is 1 and data for the next transfer is not set), and before the setting of data to be transmitted when this register is referenced.

If the contents of SPCMD0 are changed while the SPCR.SPE bit is 1, do not perform subsequent operations.

CPHA bit (RSPCK Phase Setting)

The CPHA bit selects the RSPCK phase of the SPI in master or slave mode. Data communications between SPI modules require the same RSPCK phase setting between the modules.

CPOL bit (RSPCK Polarity Setting)

The CPOL bit selects the RSPCK polarity of the SPI in master or slave mode. Data communications between SPI modules require the same RSPCK polarity setting between the modules.

BRDV[1:0] bits (Bit Rate Division Setting)

The BRDV[1:0] bits determine the bit rate in combination with the settings in the SPBR register. (see section 26.2.6. SPBR : SPI Bit Rate Register). The SPBR settings determine the base bit rate. The BRDV[1:0] setting selects the bit rate obtained by dividing the base bit rate by 1, 2, 4, or 8. Different BRDV[1:0] bit settings can be specified in the SPCMD0 register. This enables execution of serial transfers at different bit rates for each command.

SSLA[2:0] bits (SSL Signal Assertion Setting)

The SSLA[2:0] bits control the SSLni signal assertion when the SPI performs serial transfers in master mode. When an SSLni signal is asserted, its polarity is determined by the value set in the associated SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as the SSLn0 pin acts as input).

When using the SPI in slave mode, set the SSLA[2:0] bits to 000b.

SPB[3:0] bits (SPI Data Length Setting)

The SPB[3:0] bits specify the transfer data length for the SPI in master or slave mode.

LSBF bit (SPI LSB First)

The LSBF bit specifies the data format of the SPI in master or slave mode to MSB-first or LSB-first.

SPNDEN bit (SPI Next-Access Delay Enable)

The SPNDEN bit specifies the next-access delay, the period from the time the SPI in master mode terminates a serial transfer and sets the SSLni signal inactive until the SPI enables the SSLni signal assertion for the next access. If the SPNDEN bit is 0, the SPI sets the next-access delay to 1 RSPCK + 2 PCLKB. If the SPNDEN bit is 1, the SPI inserts a next-access delay according to the SPND setting.

When using the SPI in slave mode, set the SPNDEN bit to 0.

SLNDEN bit (SSL Negation Delay Setting Enable)

The SLNDEN bit specifies the SSL negation delay, the period from the time the SPI in master mode stops RSPCK oscillation until the SPI sets the SSLni signal to inactive. If the SLNDEN bit is 0, the SPI sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, the SPI negates the SSL signal at the SSL negation delay according to the SSLND setting.

When using the SPI in slave mode, set the SLNDEN bit to 0.

SCKDEN bit (RSPCK Delay Setting Enable)

The SCKDEN bit specifies the SPI clock delay, the period from the point when the SPI in master mode asserts the SSLni signal until the RSPCK starts oscillation. If the SCKDEN bit is 0, the SPI sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, the SPI starts the oscillation of RSPCK at an RSPCK delay according to the SPCKD setting.

When using the SPI in slave mode, set the SCKDEN bit to 0.

26.3 Operation

In this section, the serial transfer period refers to the period from the beginning of driving valid data to the fetching of the final valid data.

26.3.1 Overview of SPI Operation

The SPI is capable of synchronous serial transfers in the following modes:

- Slave mode (SPI operation)
- Single master mode (SPI operation)
- Multi-master mode (SPI operation)
- Slave mode (clock synchronous operation)
- Master mode (clock synchronous operation)

The SPI mode can be selected by using the MSTR, MODFEN, and SPMS bits in SPCR. Table 26.4 lists the relationship between SPI modes and SPCR settings, and a description of each mode.

Table 26.4 Relationship between SPCR settings and SPI modes (1 of 2)

Mode	Slave (SPI operation)	Single-master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
RSPCKn pins	Input	Output	Output/Hi-Z	Input	Output
MOSIn pin	Input	Output	Output/Hi-Z	Input	Output
MISOn pin	Output/Hi-Z	Input	Input	Output	Input
SSLn0 pin	Input	Output	Input	Hi-Z*1	Hi-Z*1
SSLn1 pin	Hi-Z*1	Output	Output/Hi-Z	Hi-Z*1	Hi-Z*1
SSL polarity change function	Supported	Supported	Supported	—	—
Max transfer rate	PCLKB/6	PCLKB/2	PCLKB/2	PCLKB/6	PCLKB/2
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	Two				
Clock phase	Two	Two	Two	One (CPHA = 1)	Two
First transfer bit	MSB/LSB				
Transfer data length	8 to 16, 20, 24, 32 bits				
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer trigger	SSL input active or RSPCK oscillation	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)	RSPCK oscillation	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)
Transmit buffer empty detection	Supported				

Table 26.4 Relationship between SPCR settings and SPI modes (2 of 2)

Mode	Slave (SPI operation)	Single-master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
Receive buffer full detection	Supported*2				
Overrun error detection	Supported*2	Supported*2*4	Supported*2*4	Supported*2	Supported*2
Parity error detection	Supported*3*2				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported
Underrun error detection	Supported	Not supported	Not supported	Supported	Not supported

Note 1. This function is not supported in this mode.
Note 2. When the SPCR.TXMD bit is 1, detection of receiver buffer full, overrun error, and parity error are not performed.
Note 3. When the SPCR2.SPPE bit is 0, parity error detection is not performed.
Note 4. When the SPCR2.SCKASE bit is 1, overrun error detection does not proceed.

26.3.2 Controlling the SPI Pins

Based on the settings of the MSTR, MODFEN, and SPMS bits in SPCR and the PmnPFS.NCODR bit for I/O Ports, the SPI can switch pin states. Table 26.5 lists the relationship between pin states and bit settings. Setting the PmnPFS.NCODR bit for an I/O port to 0 selects the CMOS output. Setting it to 1 selects the open-drain output. The I/O port settings must follow this relationship.

Table 26.5 Relationship between pin states and bit settings (1 of 2)

Mode	Pin	Pin state*2	
		PmnPFS.NCODR bit for I/O ports = 0	PmnPFS.NCODR bit for I/O ports = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCKn	CMOS output	Open-drain output
	SSLn0, SSLn1	CMOS output	Open-drain output
	MOSIn	CMOS output	Open-drain output
	MISOn	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCKn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSLn0	Input	Input
	SSLn1*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSIn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISOn	Input	Input
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCKn	Input	Input
	SSLn0	Input	Input
	SSLn1*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
	MISOn*4	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master mode (clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCKn	CMOS output	Open-drain output
	SSLn0, SSLn1*5	Hi-Z*1	Hi-Z*1
	MOSIn	CMOS output	Open-drain output
	MISOn	Input	Input
Slave mode (clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCKn	Input	Input

Table 26.5 Relationship between pin states and bit settings (2 of 2)

Mode	Pin	Pin state*2	
		PmnPFS.NCODR bit for I/O ports = 0	PmnPFS.NCODR bit for I/O ports = 1
	SSLn0, SSLn1*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
	MISOn	CMOS output	Open-drain output

Note 1. This function is not supported in this mode.
Note 2. SPI settings are not reflected in multiplexed pins for which the SPI function is not selected.
Note 3. When SSLn0 is at the active level, the pin state is Hi-Z. Whether or not the input signal is at the active level determines the setting of the SSLP.SSL0P bit.
Note 4. When SSLn0 is at the non-active level or the SPCR.SPE bit is 0, the pin state is Hi-Z. Whether or not the input signal is at the active level determines the setting of the SSLP.SSL0P bit.
Note 5. These pins are available for use as I/O port pins.

The SPI in single-master mode (SPI operation) or multi-master mode (SPI operation) determines the MOSI signal values during the SSL negation period based on the MOIFE and MOIFV bit settings in SPPCR, as listed in Table 26.6.

Table 26.6 MOSI signal value determination during SSL negation

MOIFE bit	MOIFV bit	MOSIn signal value during SSL negation
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

26.3.3 SPI System Configuration Examples

26.3.3.1 Single-master/single-slave with the MCU as a master

Figure 26.5 shows a single-master/single-slave SPI system configuration example where the MCU is used as a master. In the single-master/single-slave configuration, the SSLni outputs of the MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in the selected state.*1

Note 1. In the transfer format configured when the SPCMD0.CPHA bit is 0, the SSL signal for some slave devices cannot be fixed to an active level. In this case, always connect the SSLni output of the MCU to the SSL input of the slave device.

The MCU (master) drives the RSPCKn and MOSIn signals. The SPI slave drives the MISO signal.

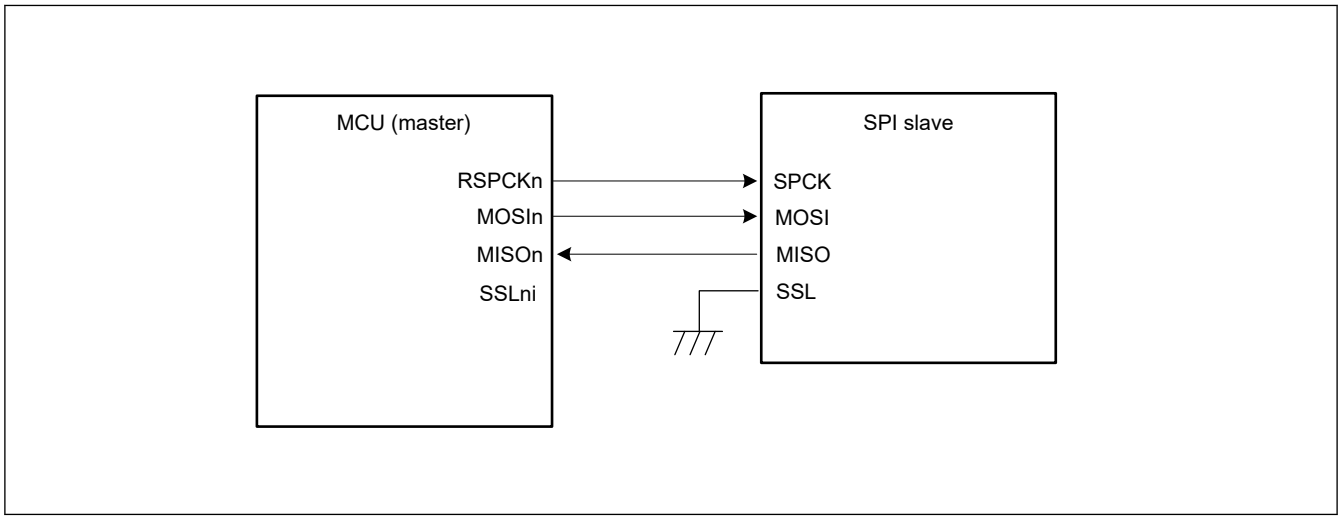


Figure 26.5 Single-master/single-slave configuration example with the MCU as a master

26.3.3.2 Single-master/single-slave with the MCU as a slave

Figure 26.6 shows a single-master/single-slave SPI system configuration example where the MCU is used as a slave. When the MCU operates as a slave, the SSLn0 pin is used as SSL input. The SPI master drives the RSPCK and MOSI signals. The MCU (slave) drives the MISOn signal.*1

Note 1. When SSLn0 is at a non-active level, the pin state is Hi-Z.

In the single-slave configuration when the SPCMD0.CPHA bit is set to 1, the SSLn0 input of the MCU (slave) is fixed to the low level and the MCU (slave) is maintained in the selected state. This enables serial transfer execution (Figure 26.7).

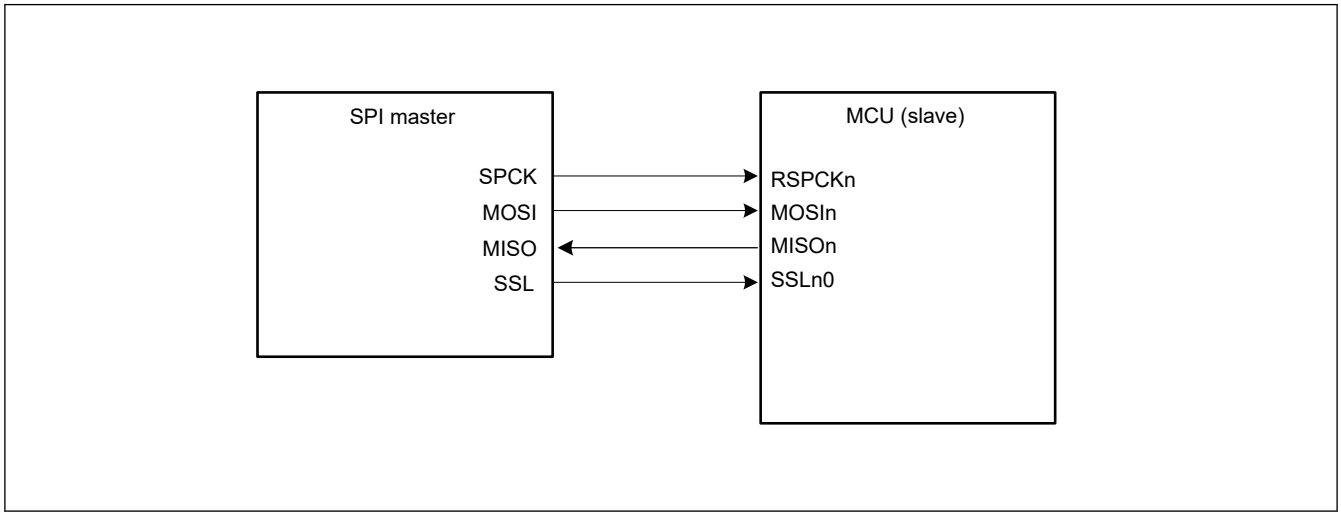


Figure 26.6 Single-master/single-slave configuration example with the MCU as a slave and CPHA = 0

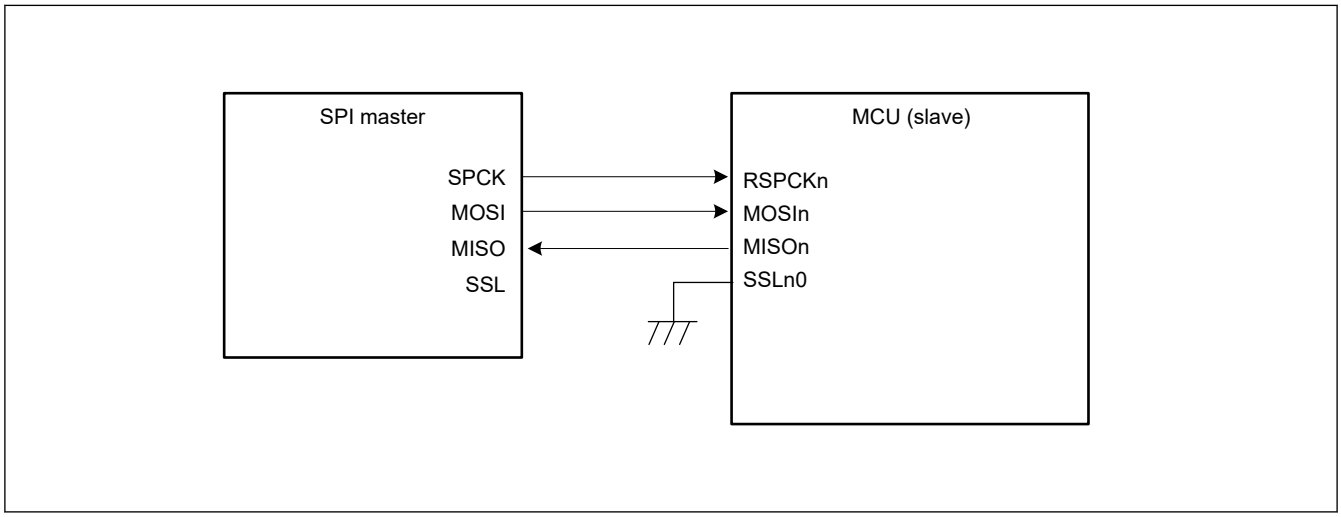


Figure 26.7 Single-master/single-slave configuration example with the MCU as a slave and CPHA = 1

26.3.3.3 Single-master/multi-slave with the MCU as a master

Figure 26.8 shows a single-master/multi-slave SPI system configuration example where the MCU is used as a master. In this example, the SPI system includes the MCU (master) and two slaves (SPI slave 0 and SPI slave 1).

The RSPCKn and MOSIn outputs of the MCU (master) are connected to the RSPCK and MOSI inputs of SPI slaves 0 and 1. The MISO outputs of SPI slaves 0 and 1 are all connected to the MISOn input of the MCU (master). The SSLn0 and SSLn1 outputs of the MCU (master) are connected to the SSL inputs of SPI slaves 0 and 1, respectively.

The MCU (master) drives the RSPCKn, MOSIn, and SSLn0 and SSLn1 signals. Of the SPI slaves 0 and 1, the slave that receives low-level input into the SSL input drives the MISO signal.

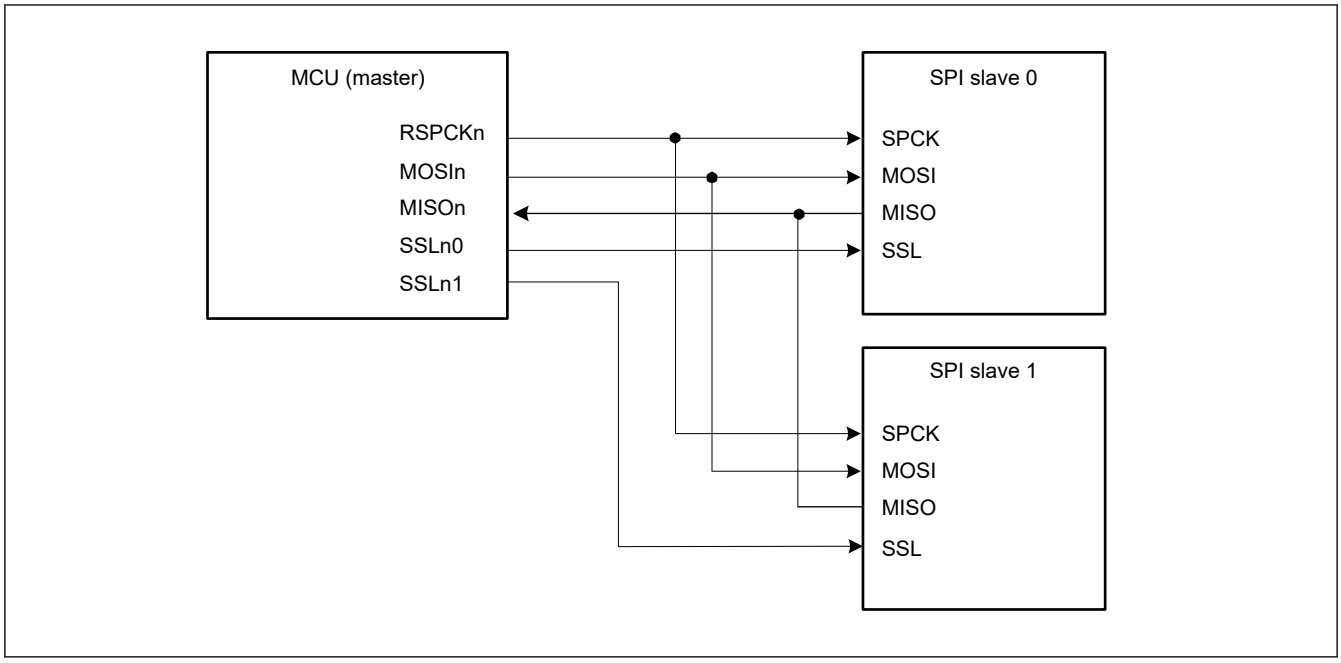


Figure 26.8 Single-master/multi-slave configuration example with the MCU as a master

26.3.3.4 Single-master/multi-slave with the MCU as a slave

Figure 26.9 shows a single-master/multi-slave SPI system configuration example where the MCU is used as a slave. In this example, the SPI system includes an SPI master and two MCUs (slaves X and Y).

The SPCK and MOSI outputs of the SPI master are connected to the RSPCKn and MOSIn inputs of the MCUs (slaves X and Y). The MISOn outputs of the MCUs (slaves X and Y) are all connected to the MISO input of the SPI master. The SSLX and SSLY outputs of the SPI master are connected to the SSLn0 inputs of the MCUs (slaves X and Y, respectively).

The SPI master drives the SPCK, MOSI, SSLX, and SSLY signals. Of the MCUs (slaves X and Y), the slave that receives low-level input into the SSLn0 input drives the MISOn signal.

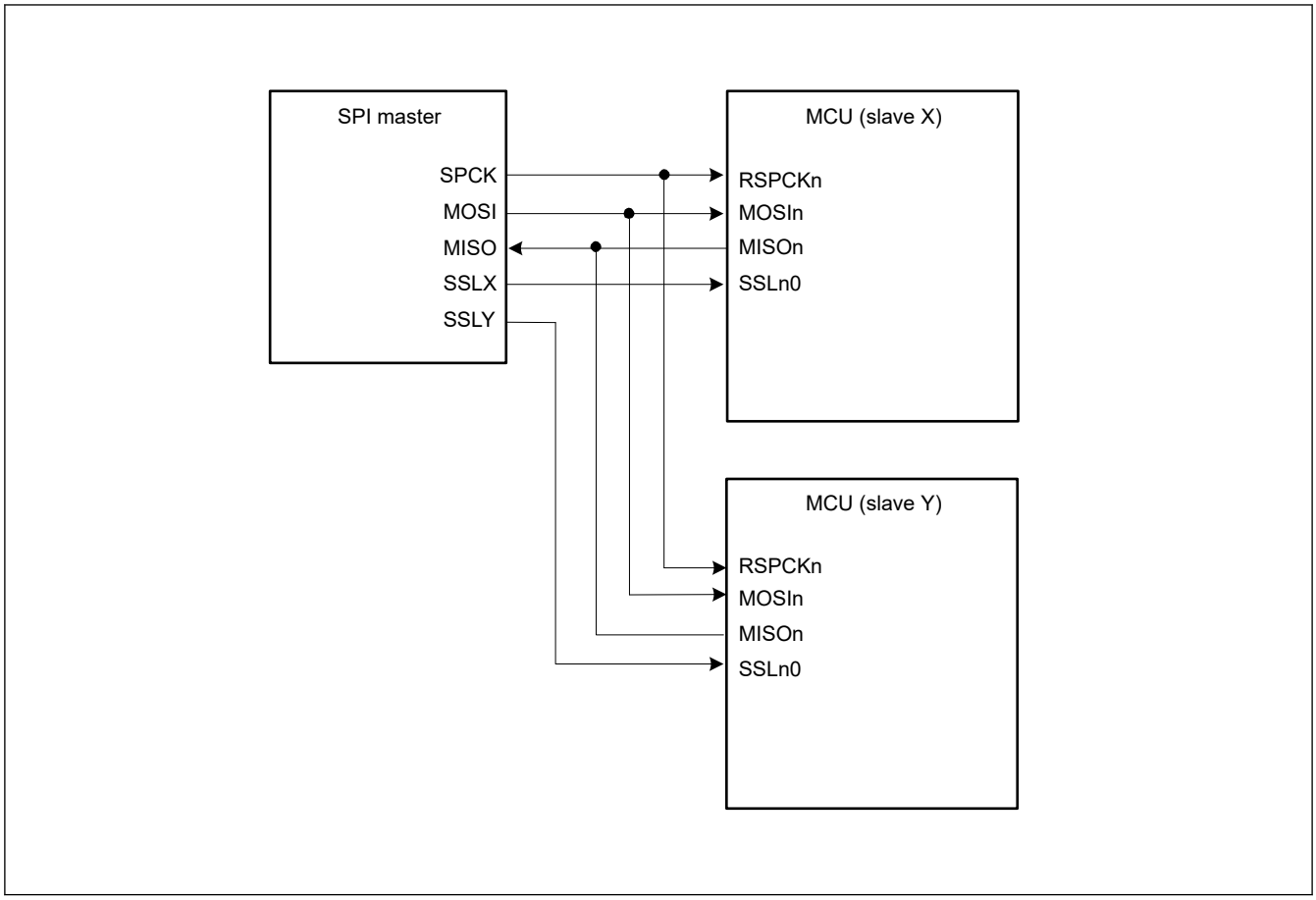


Figure 26.9 Single-master/multi-slave configuration example with the MCU as a slave

26.3.3.5 Master and slave in clock synchronous mode with the MCU configured as a master

Figure 26.10 shows a master and slave in clock synchronous mode configuration example where the MCU is used as a master. In this configuration, SSLni of the MCU (master) are not used.

The MCU (master) drives the RSPCKn and MOSIn signals. The SPI slave drives the MISO signal.

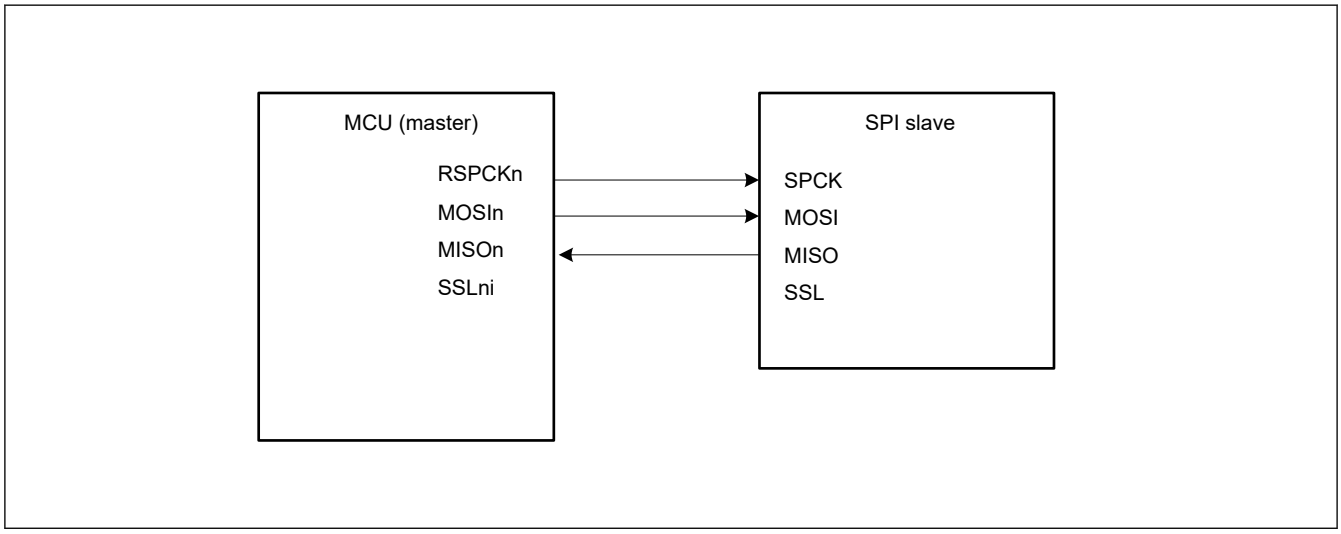


Figure 26.10 Clock synchronous master/slave configuration example with the MCU as a master

26.3.3.6 Master and slave in clock synchronous mode with the MCU as a slave

Figure 26.11 shows a master and slave in clock synchronous mode configuration example where the MCU is used as a slave. When the MCU operates as a slave (clock synchronous operation), the MCU (slave) drives the MISOn signal and the SPI master drives the SPCK and MOSI signals. In addition, SSLn0 and SSLn1 of the MCU (slave) are not used.

The MCU (slave) can only execute serial transfers in the single-slave configuration when the SPCMD0.CPHA bit is set to 1.

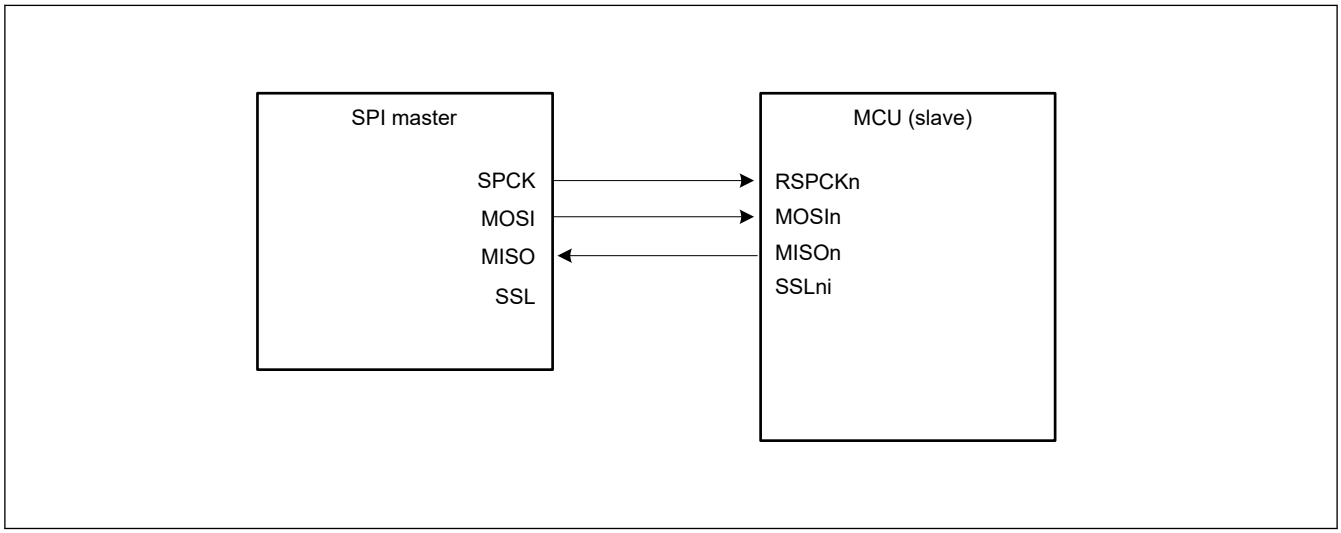


Figure 26.11 Clock synchronous master/slave configuration example with the MCU as a slave and CPHA = 1

26.3.4 Data Formats

The data format of the SPI depends on the settings in SPI Command Register 0 (SPCMD0) and the parity enable bit in SPI Control Register 2 (SPCR2.SPPE). Regardless of whether the MSB or LSB is first, the SPI treats the range from the LSB bit in the SPI Data Register (SPDR/SPDR_HA) to the bit associated with the selected data length, as transfer data.

This section shows the format of one frame of data before or after transfer.

Data format with parity disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the SPI data length setting in SPI Command Register 0 (SPCMD0.SPB[3:0]).

Data format with parity enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the SPI data length setting in SPI Command Register 0 (SPCMD0.SPB[3:0]). In this case, however, the last bit is a parity bit.

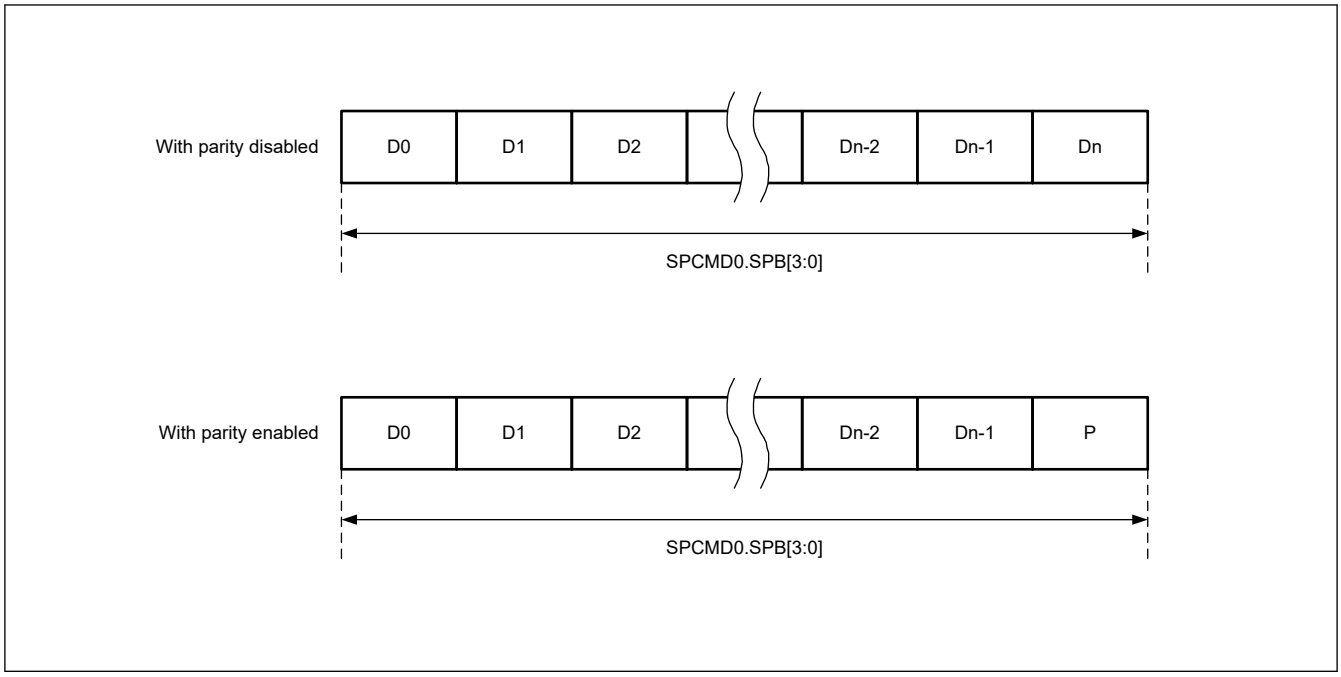


Figure 26.12 Data format with parity disabled and enabled

26.3.4.1 Operation when parity is disabled (SPCR2.SPPE = 0)

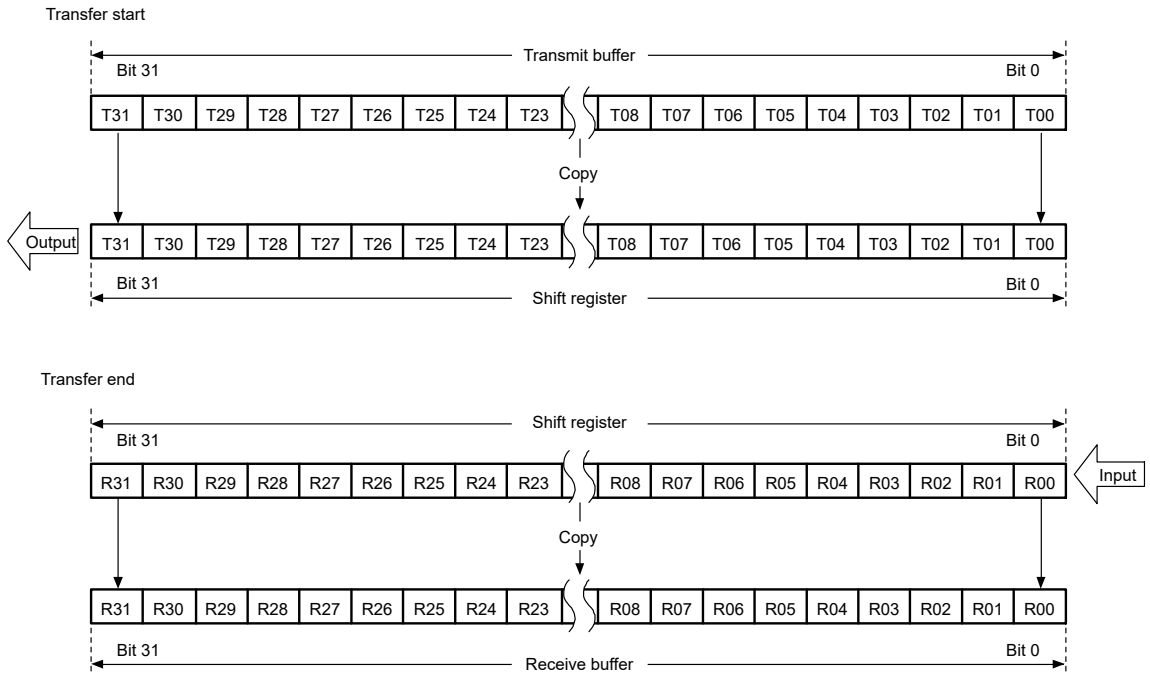
When parity is disabled, data for transmission is copied to the shift register with no pre-processing. This section describes the connection between the SPI Data Register (SPDR/SPDR_HA) and the shift register in terms of the combination of MSB- or LSB-first order and data length.

(1) MSB-first transfer with 32-bit data

Figure 26.13 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, a SPI data length of 32 bits, and MSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission is shifted out from the shift register from T31 to T30, and continuing to T00.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R31 to R00 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

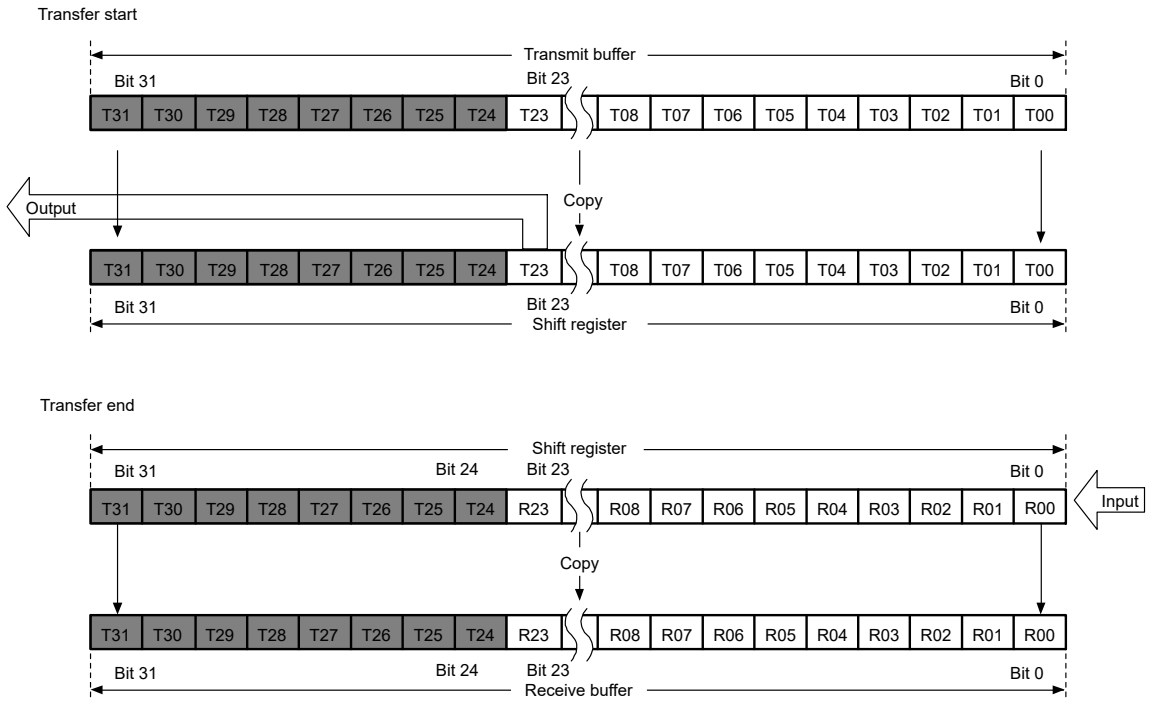
Figure 26.13 MSB-first transfer with 32-bit data and parity disabled

(2) MSB-first transfer with 24-bit data

Figure 26.14 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 24 bits for an example that is not 32 bits, and MSB-first selected.

In transmission, the lower 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission is shifted out from the shift register from T23 to T22, and continuing to T00.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R23 to R00 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. The upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 during transmission leads to 0 being inserted in the upper 8 bits of the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

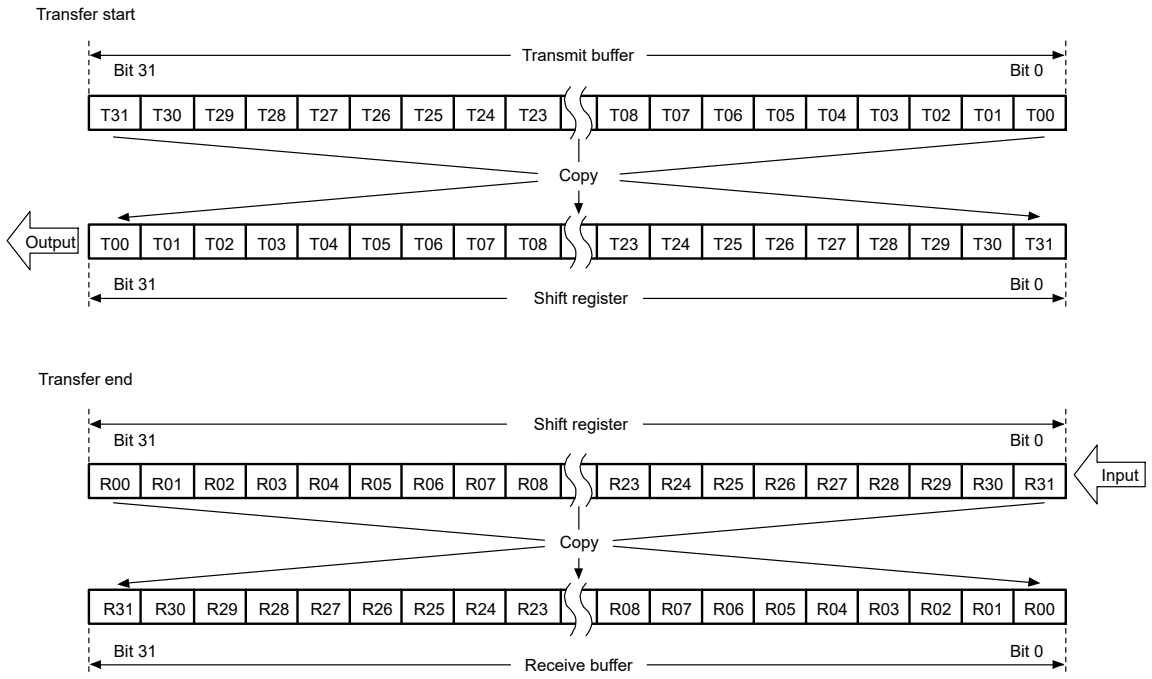
Figure 26.14 MSB-first transfer with 24-bit data and parity disabled

(3) LSB-first transfer with 32-bit data

Figure 26.15 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 32 bits, and LSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit-by-bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission is shifted out from the shift register in order from T00 to T01, and continuing to T31.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R00 to R31 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 26.15 LSB-first transfer with 32-bit data and parity disabled

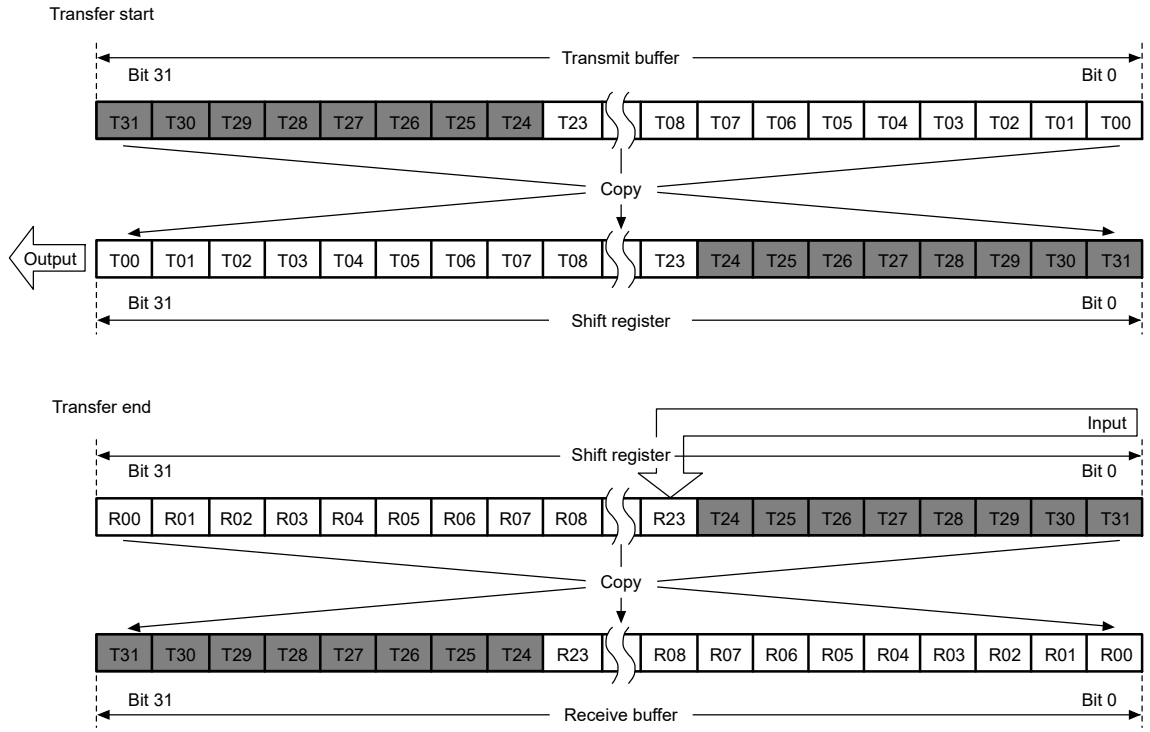
(4) LSB-first transfer with 24-bit data

Figure 26.16 shows the operation of the SPI Data Register (SPDR) and the shift register in transfers with parity disabled, an SPI data length of 24 bits for an example that is not 32, and LSB-first selected.

In transmission, the lower 24 bits (T23 to T00) from the current stage of the transmit buffer are reordered bit-by-bit to obtain the order T00 to T23 for copying to the shift register. Data for transmission is shifted out from the shift register from T00 to T01, and continuing to T23.

In reception, received data is shifted in bit-by-bit through bit[8] of the shift register. When the R00 to R23 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.

The upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to T31 to T24 during transmission leads to 0 being inserted in the upper 8 bits of the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 26.16 LSB-first transfer with 24-bit data and parity disabled

26.3.4.2 Operation when parity is enabled (SPCR2.SPPE = 1)

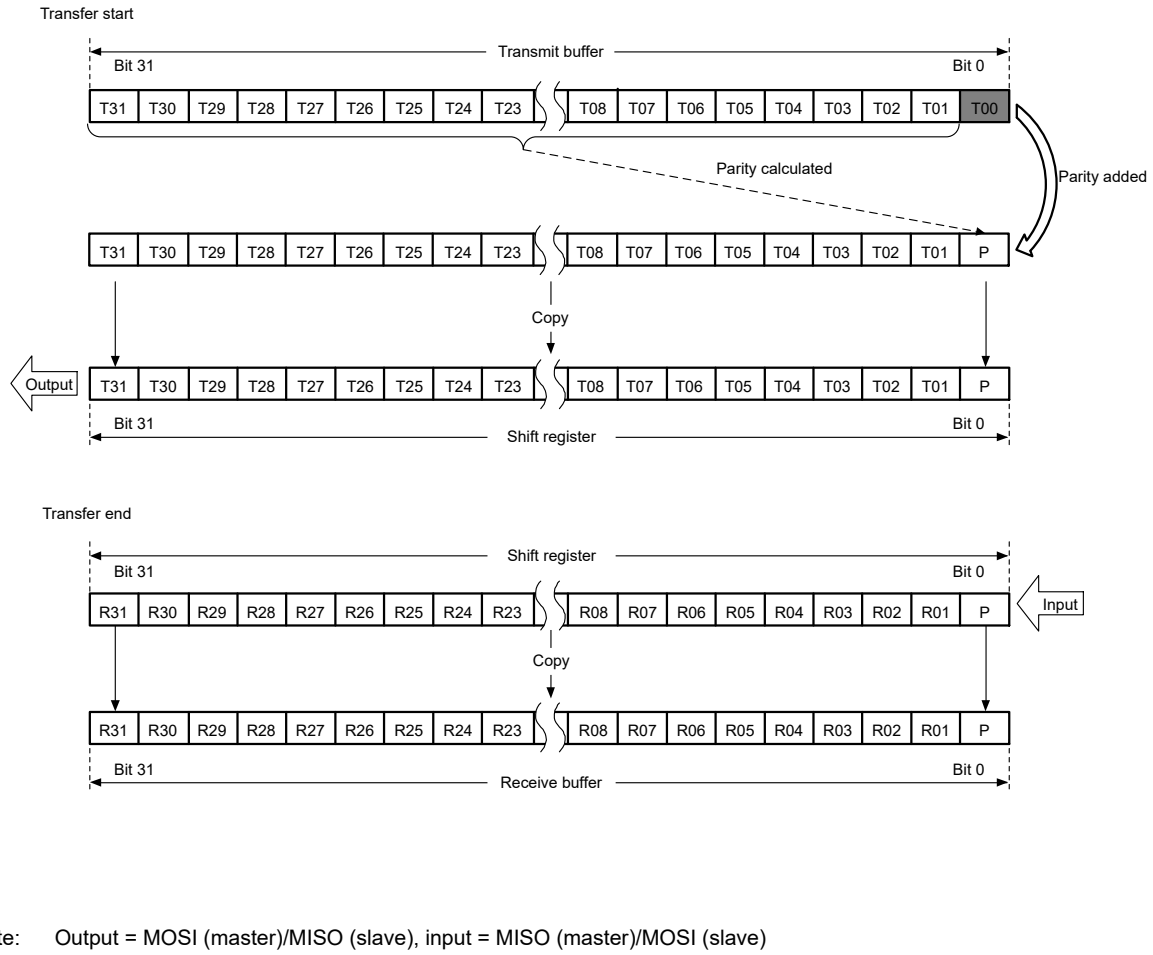
When parity is enabled, the lowest-order bit of the data for transmission becomes a parity bit. Hardware calculates the value of the parity bit.

(1) MSB-first transfer with 32-bit data

Figure 26.17 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, an SPI data length of 32 bits, and MSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole value is copied to the shift register. Data is transmitted in the order T31, T30, …, T01, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R31 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R31 to P is checked for parity.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

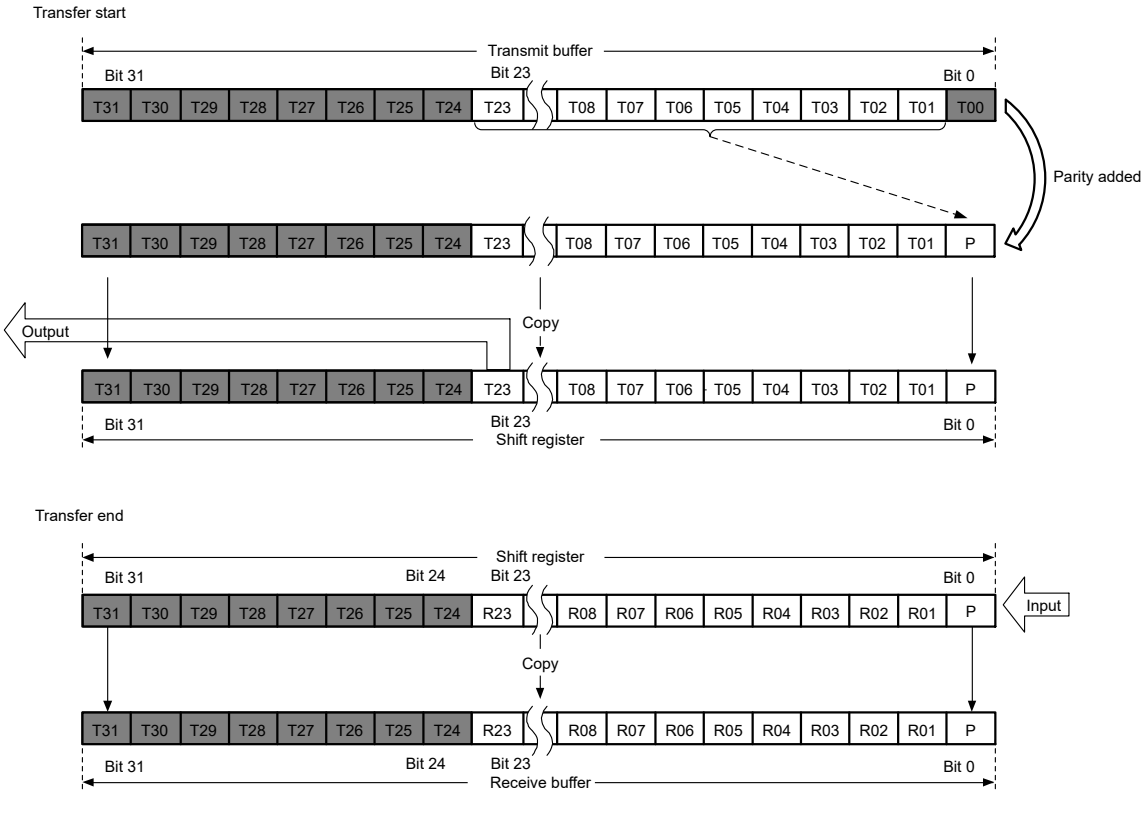
Figure 26.17 MSB-first transfer with 32-bit data and parity enabled

(2) MSB-first transfer with 24-bit data

Figure 26.18 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, a SPI data length of 24 bits, and MSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole value is copied to the shift register. Data is transmitted in the order T23, T22, …, T01, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R23 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R23 to P is checked for parity. The upper 8 bits of the transmit buffer is stored in the upper 8 bits of the receive buffer. Writing 0 to T31 to T24 during transmission leads to 0 being inserted in the upper 8 bits of the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

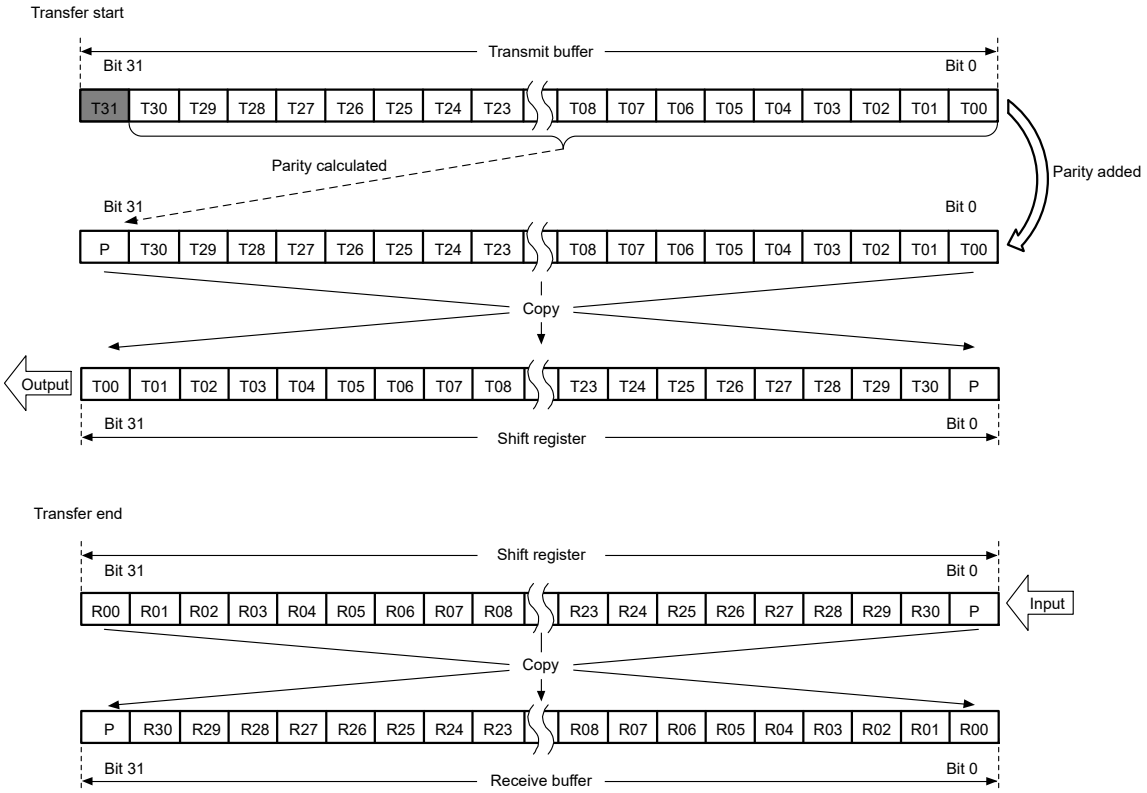
Figure 26.18 MSB-first transfer with 24-bit data and parity enabled

(3) LSB-first transfer with 32-bit data

Figure 26.19 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, an SPI data length of 32 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole value is copied to the shift register. Data is transmitted in the order T00, T01, …, T30, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R00 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P is checked for parity.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

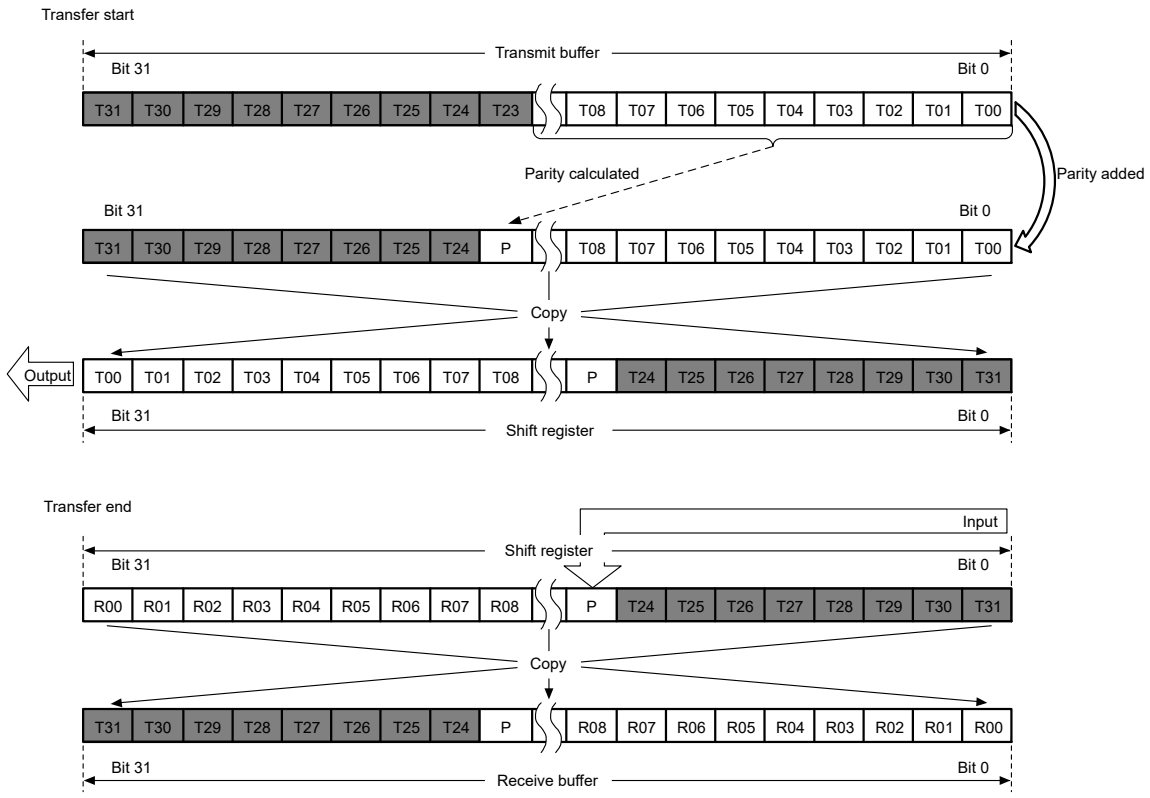
Figure 26.19 LSB-first transfer with 32-bit data and parity enabled

(4) LSB-first transfer with 24-bit data

Figure 26.20 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, a SPI data length of 24 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole value is copied to the shift register. Data is transmitted in the order T00, T01, …, T22, and P.

In reception, received data is shifted in bit-by-bit through bit[8] of the shift register. When the R00 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P is checked for parity. The upper 8 bits of the transmit buffer is stored in the upper 8 bits of the receive buffer. Writing 0 to T31 to T24 during transmission leads to 0 being inserted in the upper 8 bits of the receive buffer.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 26.20 LSB-first transfer with 24-bit data and parity enabled

26.3.5 Transfer Formats

26.3.5.1 When CPHA = 0

Figure 26.21 shows an example transfer format for the serial transfer of 8-bit data when the SPCMD0.CPHA bit is 0. Do not perform clock synchronous operation (SPCR.SPMS = 1) when the SPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 26.21, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMD0.CPOL bit is 0, and RSPCKn (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the SPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the SPI settings. For details, see section 26.3.2. Controlling the SPI Pins.

When the SPCMD0.CPHA bit is 0, the driving of valid data to the MOSIn and MISOn signals begins at an SSLni signal assertion. The first RSPCKn signal change that occurs after the SSLni signal assertion becomes the first transfer data fetch. After this, data is sampled every 1 RSPCKn cycle. The change timing for MOSIn and MISOn signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCK signal operation timing as it only affects the signal polarity.

t1 denotes the RSPCK delay, the period from an SSLni signal assertion to RSPCKn oscillation. t2 denotes the SSL negation delay, the period from the termination of RSPCKn oscillation to an SSLni signal negation. t3 denotes the next-access delay, the period in which SSLni signal assertion is suppressed for the next transfer after the end of serial transfer. t1, t2, and t3 are controlled by a master device running on the SPI system. For a description of t1, t2, and t3 when the SPI is in master mode, see section 26.3.10.1. Master mode operation.

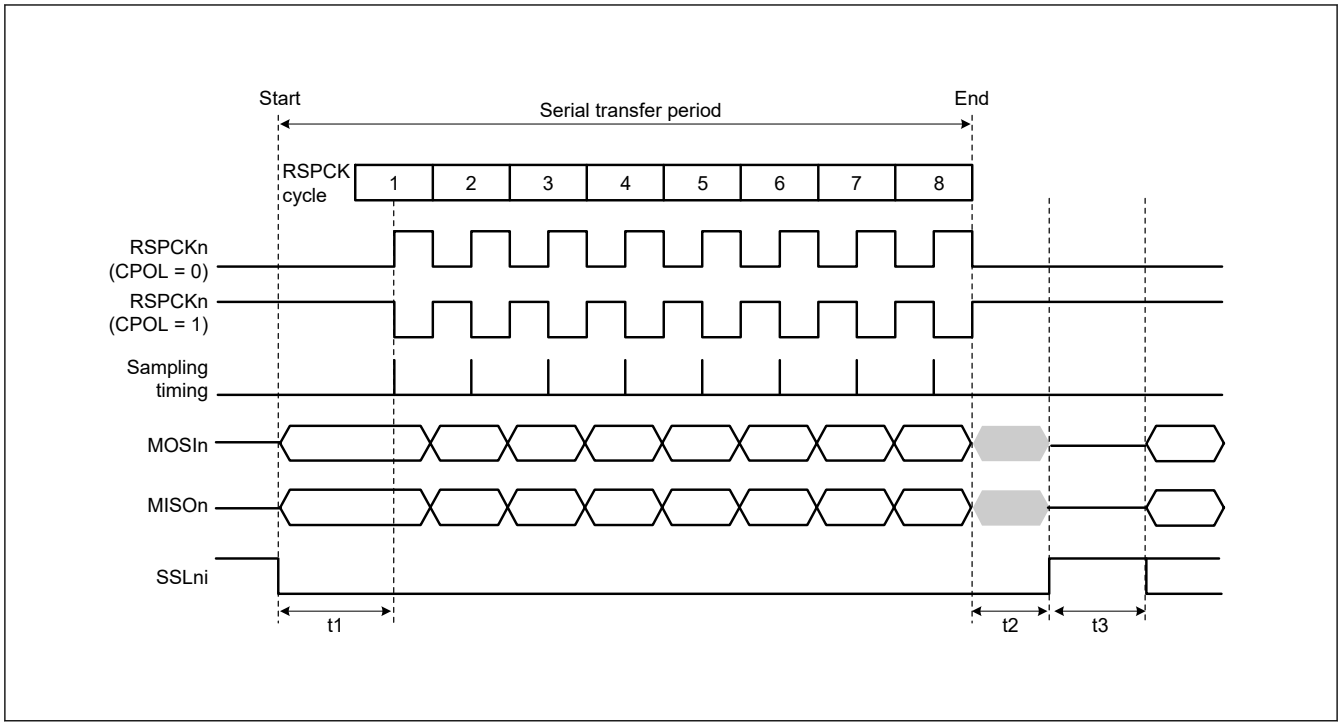


Figure 26.21 SPI transfer format when CPHA = 0

26.3.5.2 When CPHA = 1

Figure 26.22 shows an example transfer format for the serial transfer of 8-bit data when the SPCMD0.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLni signals are not used, and only the three signals RSPCKn, MOSIn, and MISOn handle communications. In Figure 26.22, RSPCK (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMD0.CPOL bit is 0 and RSPCK (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the SPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the SPI mode (master or slave mode). For details, see section 26.3.2. Controlling the SPI Pins.

When the SPCMD0.CPHA bit is 1, the driving of invalid data to the MISOn signal begins at an SSLni signal assertion. The output of valid data to the MOSIn and MISOn signals begins at the first RSPCKn signal change that occurs after the SSLni signal assertion. After this, data is updated every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMD0.CPOL bit setting does not affect the RSPCKn signal operation timing. It only affects the signal polarity.

t1, t2, and t3 are the same as those when CPHA = 0. For a description of t1, t2, and t3 when the SPI of the MCU is in master mode, see section 26.3.10.1. Master mode operation.

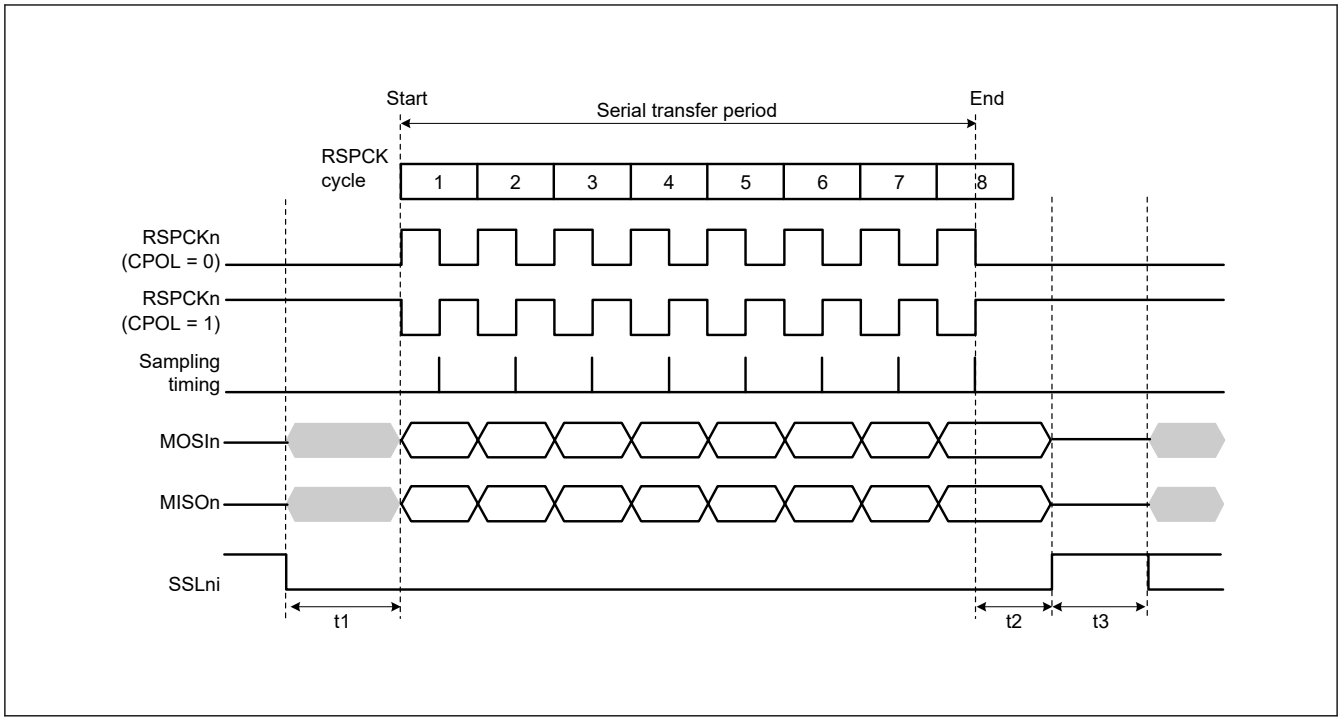


Figure 26.22 SPI transfer format when CPHA = 1

26.3.6 Data Transfer Modes

Full-duplex synchronous serial communications or transmit operations can only be selected in the Communications Operating Mode Select bit (SPCR.TXMD). The register accesses shown in Figure 26.23 and Figure 26.24 indicate the condition of access to the SPDR/SPDR_HA register, where W denotes a write cycle.

26.3.6.1 Full-duplex synchronous serial communications (SPCR.TXMD = 0)

Figure 26.23 shows an example of operation when the Communications Operating Mode Select bit (SPCR.TXMD) is set to 0. In this example, the SPI performs an 8-bit serial transfer when the SPCMD0.CPHA bit is 1, and the SPCMD0.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

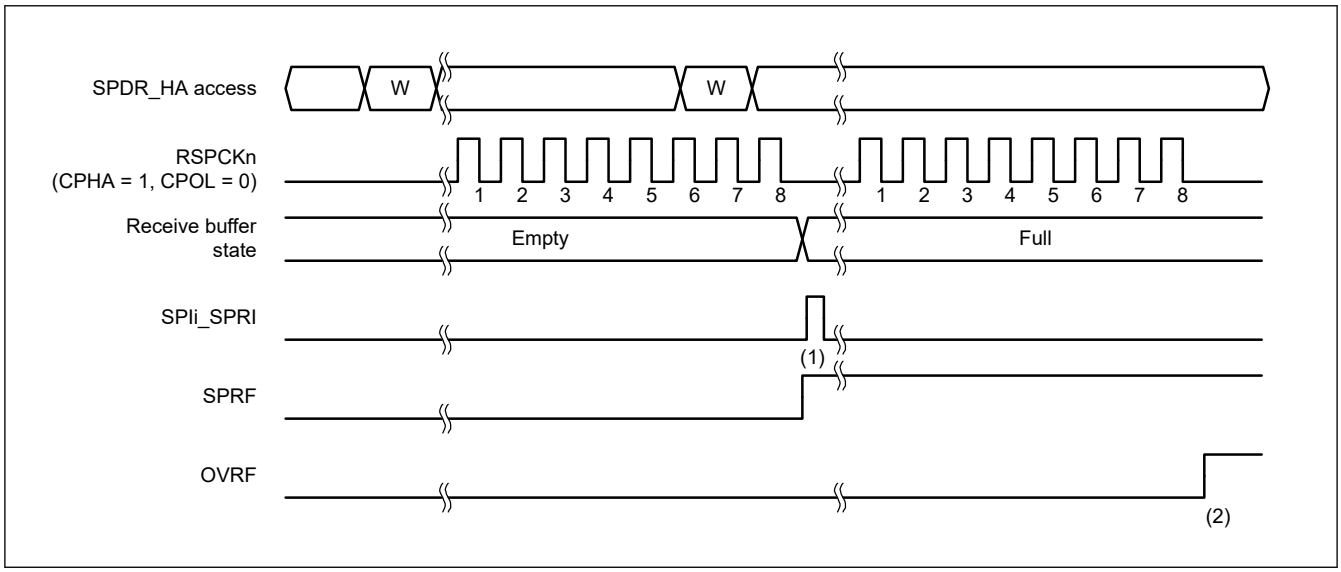


Figure 26.23 Operation example when SPCR.TXMD = 0

The operation of the flags at timings (1) and (2) in Figure 26.23 is as follows:

1. When a serial transfer ends with the receive buffer of SPDR_HA empty, the SPI generates a receive buffer full interrupt request (SPIi_SPRI), the SPI sets the SPSR.SPRF flag to 1, and copies the received data in the shift register to the receive buffer.
2. When a serial transfer ends with the receive buffer of SPDR_HA holding data that was received in the previous serial transfer, the SPI sets the SPSR.OVRF flag to 1, and discards the received data in the shift register. For details about the operation of the SPSR.OVRF flag, see section 26.3.8.1. Overrun errors.

26.3.6.2 Transmit-Only Serial Communications (SPCR.TXMD = 1)

Figure 26.24 shows an example of operation when the Communications Operating Mode Select bit (SPCR.TXMD) is set to 1. In this example, the SPI performs an 8-bit serial transfer when the SPCMD0.CPHA bit is 1, and the SPCMD0.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

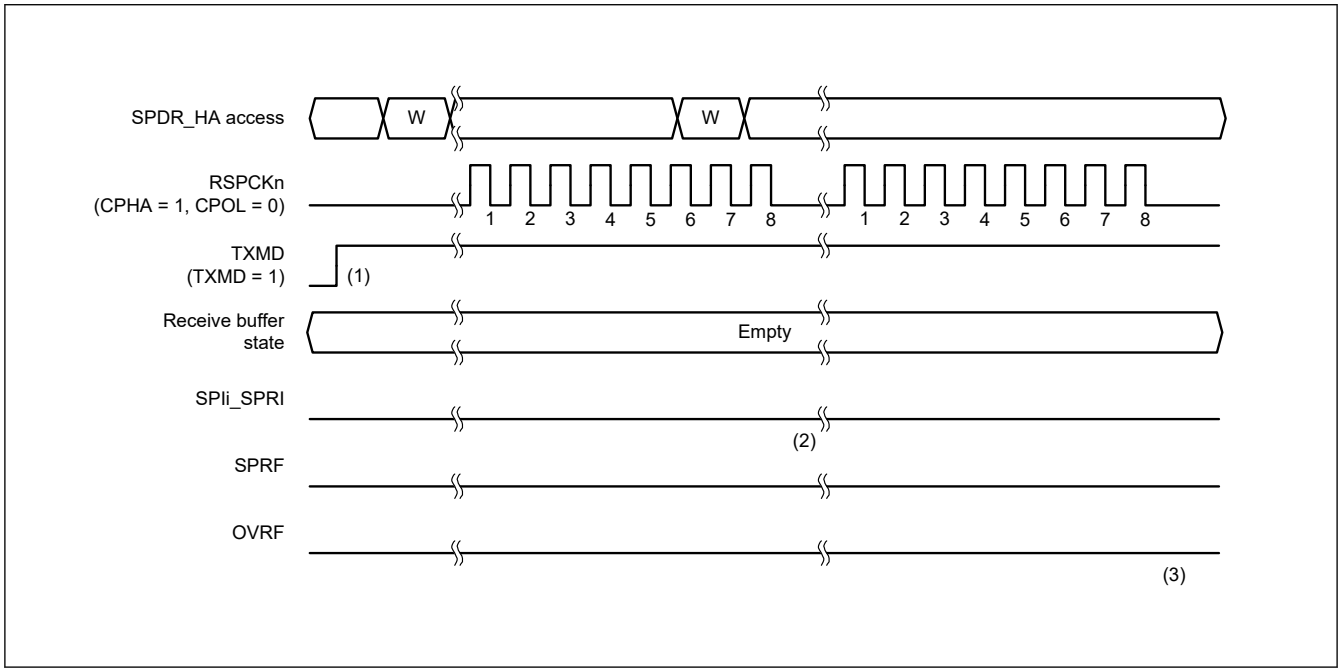


Figure 26.24 Operation example when SPCR.TXMD = 1

The operation of the flags at timings (1) to (3) in Figure 26.24 is as follows:

1. Make sure there is no data left in the receive buffer (the SPSR.SPRF flag is 0) and the SPSR.OVRF flag is 0 before entering transmit-only mode (SPCR.TXMD = 1).
2. When a serial transfer ends with the receive buffer of SPDR_HA empty, if the transmit-only mode is selected (SPCR.TXMD = 1), the SPSR.SPRF flag retains the value of 0, and the SPI does not copy the data in the shift register to the receive buffer.
3. Because the receive buffer of SPDR_HA does not hold data that was received in the previous serial transfer, even when a serial transfer ends, the SPSR.OVRF flag retains the value of 0, and the data in the shift register is not copied to the receive buffer.

In transmit-only mode (SPCR.TXMD = 1), the SPI transmits data but does not receive data. Therefore, the SPSR.SPRF and SPSR.OVRF flags remain 0 at timings (1) to (3).

26.3.7 Transmit Buffer Empty and Receive Buffer Full Interrupts

Figure 26.25 and Figure 26.26 show examples of operation of the transmit buffer empty interrupt (SPIi_SPTI) and the receive buffer full interrupt (SPIi_SPRI). The SPDR_HA register accesses shown in these figures indicate the conditions of access to the register, where W denotes a write cycle and R a read cycle. In Figure 26.25, the SPI performs an 8-bit serial transfer when SPCR.TXMD bit is 0, the SPCMD0.CPHA bit is 0, and the SPCMD0.CPOL bit is 0. In Figure 26.26, the SPI performs an 8-bit serial transfer when SPCR.TXMD bit is 0, the SPCMD0.CPHA bit is 1, and the SPCMD0.CPOL bit is 0.

The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

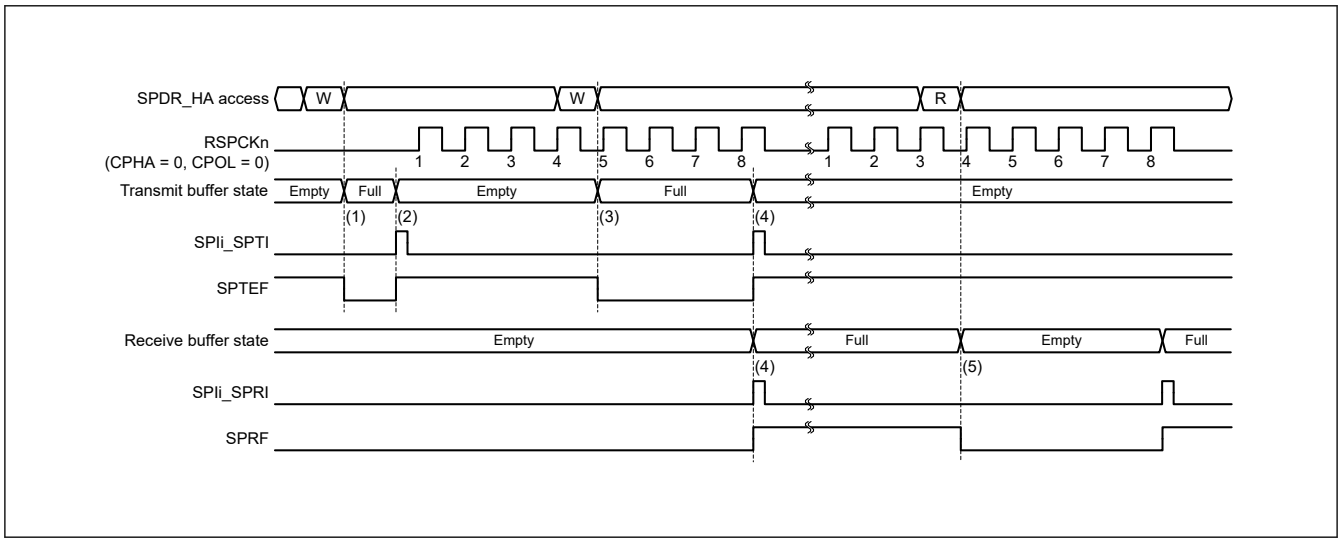


Figure 26.25 Operation example of the SPIi_SPTI and SPIi_SPRI interrupts when CPHA = 0 and CPOL = 0 in master mode

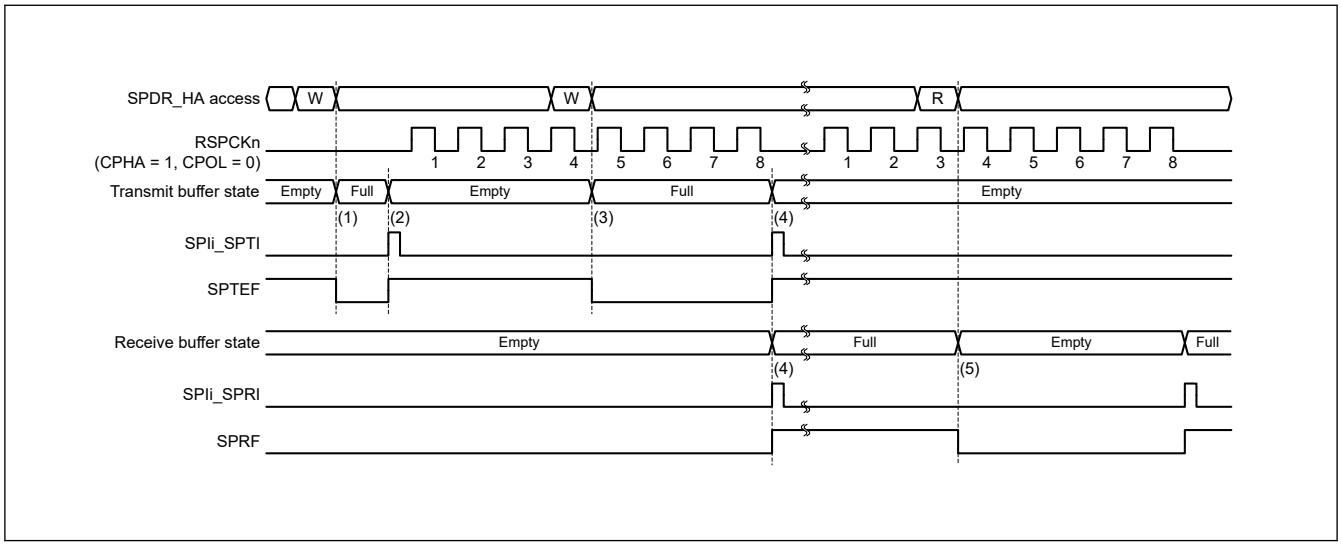


Figure 26.26 Operation example of the SPIi_SPTI and SPIi_SPRI interrupts when CPHA = 1 and CPOL = 0 in master mode

The operation of the SPI at timings (1) to (5) in Figure 26.25 and Figure 26.26 is as follows:

1. When transmit data is written to SPDR_HA with the transmit buffer of SPDR_HA is empty and data for the next transfer is not set, the SPI writes data to the transmit buffer and clears the SPSR.SPTEF flag to 0.
2. If the shift register is empty, the SPI copies the data in the transmit buffer to the shift register, generates a transmit buffer empty interrupt request (SPIi_SPTI), and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the SPI mode. For details, see section 26.3. Operation, and section 26.3.11. Clock Synchronous Operation.
3. When transmit data is written to SPDR_HA either by the transmit buffer empty interrupt routine, or by the processing of the transmit buffer empty using the SPTEF flag, the SPI writes data to the transmit buffer and clears the SPTEF flag to 0. Because the data being transferred serially is stored in the shift register, the SPI does not copy the data in the transmit buffer to the shift register.
4. When the serial transfer ends with the receive buffer of SPDR_HA empty, the SPI copies the receive data in the shift register to the receive buffer, generates a receive buffer full interrupt request (SPIi_SPRI), and sets the SPRF flag to 1. Because the shift register becomes empty on completion of the serial transfer, if the transmit buffer is full before the serial transfer ended, the SPI sets the SPTEF flag to 1 and copies data in the transmit buffer to the shift register. Even

when received data is not copied from the shift register to the receive buffer in an overrun error status, on completion of the serial transfer, the SPI determines that the shift register is empty, so data transfer from the transmit buffer to the shift register is enabled.

5. When SPDR_HA is read either by the receive buffer full interrupt routine or processing of the receive buffer full interrupt using the SPRF flag, the receive data can be read.

If SPDR_HA is written to when the transmit buffer holds data that is not yet transmitted (the SPTEF flag is 0), the SPI does not update data in the transmit buffer. When writing to SPDR_HA, always use either a transmit buffer empty interrupt request or processing of the transmit buffer empty interrupt using the SPTEF flag. To use a transmit buffer empty interrupt, set the SPTIE bit in SPCR to 1. If the SPI function is disabled (the SPCR.SPE bit is 0), set the SPTIE bit to 0.

When serial transfer ends and the receive buffer is full (the SPRF flag is 1), the SPI does not copy data from the shift register to the receive buffer, and it detects an overrun error (see section 26.3.8. Error Detection). To prevent a receive data overrun error, read the received data using a receive buffer full interrupt request before the next serial transfer ends. To use an SPI receive buffer full interrupt, set the SPCR.SPRIE bit to 1.

Transmission and reception interrupts or the associated IELSRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmit and receive buffers.

Similarly, the SPTEF and SPRF flags can be used to confirm the states of the transmit and receive buffers. See section 12, Interrupt Controller Unit (ICU) for the interrupt vector numbers.

26.3.8 Error Detection

In normal SPI serial transfers, data written to the transmit buffer of SPDR/SPDR_HA is transmitted, and received data can be read from the receive buffer of SPDR/SPDR_HA. If access is made to SPDR/SPDR_HA, an abnormal transfer might occur, depending on the status of the transmit or receive buffer or the status of the SPI at the beginning or end of serial transfer.

If an abnormal transfer occurs, the SPI detects the event as an underrun error, overrun error, parity error, or mode fault error. Table 26.7 lists the relationship between non-normal transfer operations and the SPI error detection function.

Table 26.7 Relationship between non-normal transfer operations and SPI error detection (1 of 2)

Operation	Occurrence condition	SPI operation	Error detection
1	SPDR/SPDR_HA is written when the transmit buffer is full.	• The contents of the transmit buffer are kept • Write data is missing	None
2	SPDR/SPDR_HA is read when the receive buffer is empty.	The contents of the receive buffer and previously received data are output.	None
3	Serial transfer is started in slave mode when the SPI is not able to transmit data.	• Serial transfer is suspended • Transmit or receive data is missing • Driving of the MISOn output signal is stopped • SPI function is disabled	Underrun error
4	Serial transfer terminates when the receive buffer is full.	• Keeps the contents of the receive buffer • Missing receive data	Overrun error
5	An incorrect parity bit is received during full-duplex synchronous serial communication with the parity function enabled in following mode: • Transmit-receive master mode • Transmit-receive slave mode	The parity error flag is asserted	Parity error
6	The SSLn0 input signal is asserted when the serial transfer is idle in multi-master mode.	• Driving of the RSPCKn, MOSIn, SSLn1 output signals is stopped • SPI function is disabled	Mode fault error
7	The SSLn0 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended • Transmit or receive data is missing • Driving of the RSPCKn, MOSIn, SSLn1 output signals is stopped • SPI function is disabled	Mode fault error

Table 26.7 Relationship between non-normal transfer operations and SPI error detection (2 of 2)

Operation	Occurrence condition	SPI operation	Error detection
8	The SSLn0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended • Transmit or receive data is missing • Driving of the MISOn output signal is stopped • SPI function is disabled	Mode fault error

In operation 1 described in Table 26.7, the SPI does not detect an error. To prevent data omission during writes to SPDR/SPDR_HA, the writes to SPDR/SPDR_HA must be executed using a transmit buffer empty interrupt request (when the SPSR.SPTEF flag is 1).

Similarly, the SPI does not detect an error in operation 2. To prevent extraneous data from being read, SPDR/SPDR_HA read must be executed with an SPI receive buffer full interrupt request (when the SPSR.SPRF flag is 1).

For information on the other errors, see the following sections:

- Underrun error, indicated in operation 3, see section 26.3.8.4. Underrun errors
- Overrun error, indicated in operation 4, see section 26.3.8.1. Overrun errors
- Parity error, indicated in operation 5, see section 26.3.8.2. Parity errors
- Mode fault error, indicated in operations 6 to 8, see section 26.3.8.3. Mode fault errors
- For the transmit and receive interrupts, see section 26.3.7. Transmit Buffer Empty and Receive Buffer Full Interrupts.

26.3.8.1 Overrun errors

If a serial transfer ends when the receive buffer of SPDR/SPDR_HA is full, the SPI detects an overrun error and sets the SPSR.OVRF flag to 1. When the OVRF flag is 1, the SPI does not copy data from the shift register to the receive buffer, so the data prior to the error occurrence is retained in the receive buffer. To set the OVRF flag to 0, write 0 to the OVRF flag after the CPU reads SPSR with the OVRF flag set to 1.

Figure 26.27 shows an example of operation of the OVRF and SPRF flags. The SPSR and SPDR_HA accesses shown in Figure 26.27 indicate the condition of accesses to the SPSR and SPDR_HA register, where W denotes a write cycle, and R a read cycle. In this example, the SPI performs an 8-bit serial transfer when SPCMD0.CPHA bit is 1 and the SPCMD0.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

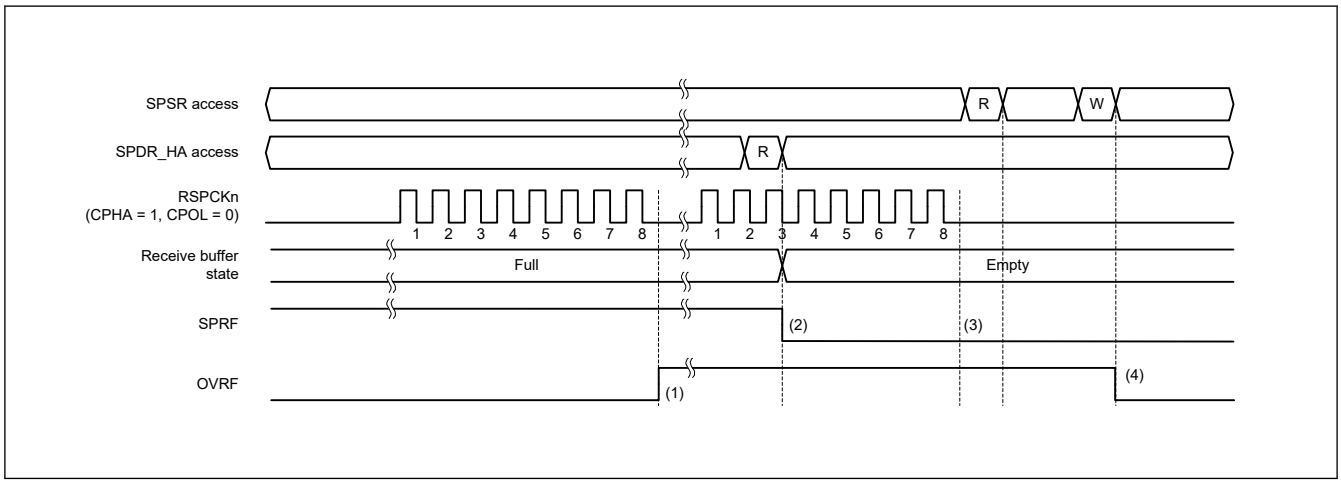


Figure 26.27 Operation example of the OVRF and SPRF flags

The operation of the flags at timings (1) to (4) in Figure 26.27 is as follows:

1. If a serial transfer terminates with the SPRF flag set to 1 (receive buffer full), the SPI detects an overrun error, and sets the OVRF flag to 1. The SPI does not copy the data in the shift register to the receive buffer. Even when the SPPE bit is 1, parity errors are not detected.
2. When SPDR/SPDR_HA is read, the SPI outputs the data in the receive buffer. The SPRF flag is then set to 0. The receive buffer becoming empty does not set the OVRF flag to 0.

3. If the serial transfer ends with the OVRF flag set to 1 (overrun error occurred), the SPI does not copy data in the shift register to the receive buffer (the SPRF flag does not set to 1). A receive buffer full interrupt is not generated. Even when the SPPE bit is 1, parity errors are not detected. In an overrun error state when the SPI does not copy the received data from the shift register to the receive buffer, on termination of the serial transfer, the SPI determines that the shift register is empty. This enables data transfer from the transmit buffer to the shift register.
4. If 0 is written to the OVRF flag after SPSR is read when the OVRF flag is 1, the OVRF flag clears is set to 0.

The occurrence of an overrun can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. When executing a serial transfer, you must ensure that overrun errors are detected early, for example by reading SPSR immediately after SPDR/SPDR_HA/SPDR_BY is read.

If an overrun error occurs and the OVRF flag sets to 1, normal reception operations cannot be performed until the OVRF flag is set to 0.

When the RSPCK auto-stop function is enabled (SPCR2.SCKASE = 1) in master mode, an overrun error does not occur. Figure 26.28 and Figure 26.29 show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.

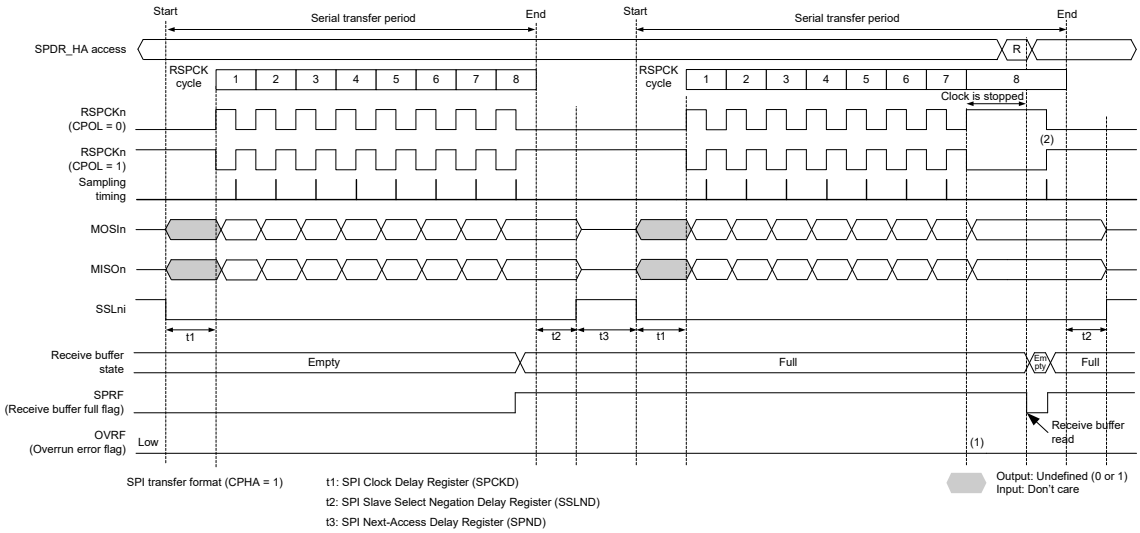


Figure 26.28 Clock stop waveform when serial transfer continues while receive buffer is full in master mode (CPHA = 1)

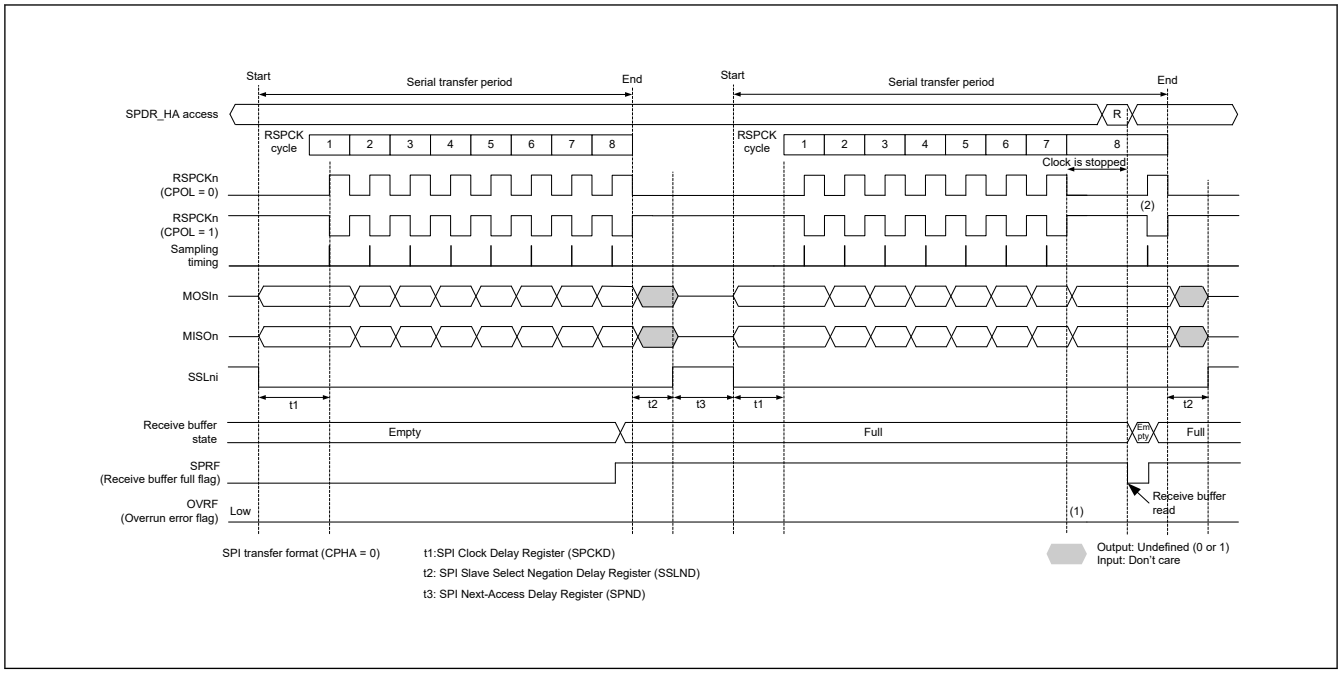


Figure 26.29 Clock stop waveform when serial transfer continues while receive buffer is full in master mode (CPHA = 0)

The operation of the flags at timings (1) and (2) in Figure 26.28 and Figure 26.29 is as follows:

1. When the receive buffer is full, an overrun error does not occur because the RSPCK clock is stopped.
2. If SPDR/SPDR_HA is read while the clock is stopped, data in the receive buffer can be read. The RSPCK clock restarts after reading the receive buffer (after the SPSR.SPRF flag is set to 0).

26.3.8.2 Parity errors

When full-duplex synchronous serial communications is performed with the SPCR.TXMD bit set to 0 and the SPCR2.SPPE bit set to 1, when serial transfer ends, the SPI checks whether there are parity errors. On detecting a parity error in the received data, the SPI sets the SPSR.PERF flag to 1. Because the SPI does not copy data in the shift register to the receive buffer when the SPSR.OVRF flag is set to 1, parity error detection is not performed for the received data. To set the PERF flag to 0, write 0 to the PERF flag after the SPSR register is read with the PERF flag set to 1.

Figure 26.30 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 26.30 indicates the condition of access to the register, where W denotes a write cycle, and R a read cycle. In this example, full-duplex serial communication is performed while the SPCR2.SPPE bit is 1. The SPI performs an 8-bit serial transfer when SPCMD0.CPHA bit is 1 and the SPCMD0.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.

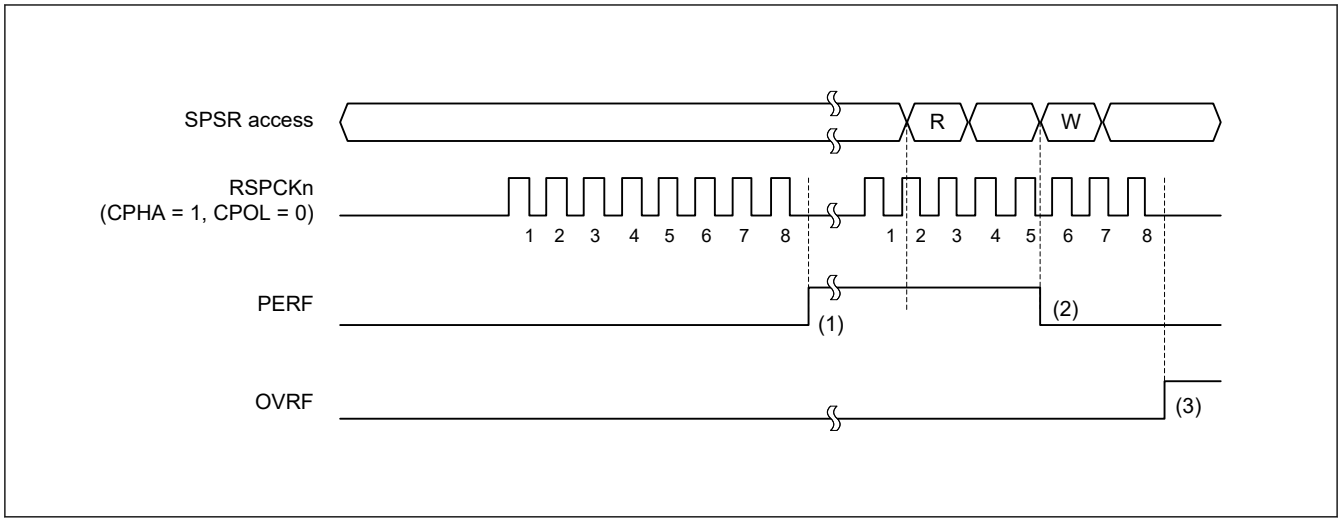


Figure 26.30 Operation example of the OVRF and PERF flags

The operation of the flags at timings (1) to (3) in Figure 26.30 is as follows:

1. If a serial transfer terminates with the SPI not detecting an overrun error, the SPI copies the data in the shift register to the receive buffer. The SPI checks the received data at this time and sets the PERF flag to 1 if a parity error is detected.
2. If 0 is written to the PERF flag after the SPSR register is read when the PERF flag is 1, the PERF flag is set to 0.
3. When the SPI detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The SPI does not perform parity error detection at this time.

Parity errors can be checked for by either reading the SPSR register or using an SPI error interrupt and reading the SPSR register. When executing a serial transfer, such checks are required to ensure early detection of parity errors.

26.3.8.3 Mode fault errors

The SPI operates in multi-master mode when the SPCR.MSTR bit is 1, the SPCR.SPMS bit is 0, and the SPCR.MODFEN bit is 1. If the active level is input for the SSLn0 input signal of the SPI in multi-master mode, the SPI detects a mode fault error regardless of the status of the serial transfer, and sets the SPSR.MODF flag to 1. The active level of the SSLn0 signal is determined by the SSLP.SSL0P bit.

When the MSTR bit is 0, the SPI operates in slave mode. The SPI detects a mode fault error if the MODFEN bit of the SPI in slave mode is 1, and the SPMS bit is 0, and if the SSLn0 input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

On detecting a mode fault error, the SPI stops the driving of the output signals and clears the SPCR.SPE bit to 0 (see section 26.3.9. Initializing the SPI). For multi-master configuration, detection of a mode fault error is used to stop the driving of output signals and the SPI function, which allows the master to be released.

The occurrence of a mode fault error can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. Detecting mode-fault errors without using the SPI error interrupt requires polling of SPSR.

When the MODF flag is 1, writing 1 to the SPE bit is ignored by the SPI. To enable the SPI function after the detection of a mode fault error, the MODF flag must be set to 0.

26.3.8.4 Underrun errors

While the SPI is operating in slave mode (SPCR.MSTR bit = 0), if serial transfer is started before transmit data output is ready with the SPCR.SPE bit set to 1 (SPI function enabled), the SPI detects an underrun error and sets the SPSR.MODF and SPSR.UDRF flags to 1.

On detecting an underrun error, the SPI stops the driving of output signals and clears the SPCR.SPE bit to 0 (see section 26.3.9. Initializing the SPI).

The occurrence of underrun errors can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. Detecting underrun errors without using the SPI error interrupt requires polling of SPSR.

When the MODF flag is 1, writing 1 to the SPE bit is ignored by the SPI. To enable the SPI function after the detection of an underrun error, the MODF flag must be set to 0.

26.3.9 Initializing the SPI

If 0 is written to the SPCR.SPE bit or if the SPI sets the SPE bit to 0 because it detected a mode fault error or an underrun error, the SPI disables the SPI function and initializes some of the module functions. When a system reset is generated, the SPI initializes all of the module functions. This section describes initialization by clearing of the SPCR.SPE bit and by a system reset.

26.3.9.1 Initialization by clearing of the SPCR.SPE bit

When the SPCR.SPE bit is set to 0, the SPI initializes by:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state of the SPI
- Initializing the transmit buffer of the SPI (the SPSR.SPTEF flag sets to 1)

Initialization by clearing of the SPE bit does not initialize the control bits of the SPI. For this reason, the SPI can be started in the same transfer mode in use prior to initialization when the SPE bit is set to 1 again.

The SPSR.SPRF, SPSR.OVRF, SPSR.MODF, SPSR.PERF, and SPSR.UDRF flags are not initialized. Therefore, even after the SPI is initialized, data from the receive buffer can be read to check the error status during an SPI transfer.

The transmit buffer is initialized to an empty state (the SPSR.SPTEF flag sets to 1). Therefore, if the SPCR.SPTIE bit is set to 1 after SPI initialization, a transmit buffer empty interrupt is generated. To disable any transmit buffer empty interrupts when the SPI is initialized, write 0 to the SPTIE bit simultaneously while writing 0 to the SPE bit.

26.3.9.2 Initialization by system reset

A system reset completely initializes the SPI by initializing all SPI control bits, status bits, and data registers, in addition to meeting the requirements described in section 26.3.9.1. Initialization by clearing of the SPCR.SPE bit.

26.3.10 SPI Operation

26.3.10.1 Master mode operation

The only difference between single- and multi-master mode operation is the use of mode fault error detection (see section 26.3.8. Error Detection). In single-master mode, the SPI does not detect mode fault errors whereas in multi-master mode, it does. This section explains operations that are common to both modes.

(1) Starting a serial transfer

The SPI updates the data in the transmit buffer (SPTX) when data is written to the SPI Data Register (SPDR/SPDR_HA) with the SPI transmit buffer empty, data for the next transfer is not set, and the SPSR.SPTEF flag is 0. When the shift register is empty, the SPI copies data from the transmit buffer to the shift register and starts serial transfer. On copying transmit data to the shift register, the SPI changes the status of the shift register to full. On termination of the serial transfer, it changes the status of the shift register to empty. The status of the shift register cannot be referenced.

The polarity of the SSLni output pins depends on the SSLP register settings. For details on the SPI transfer format, see section 26.3.5. Transfer Formats.

(2) Terminating a serial transfer

Regardless of the SPCMD0.CPHA bit setting, the SPI terminates the serial transfer after transmitting an RSPCKn edge associated with the final sampling timing. If free space is available in the receive buffer (SPRX) (the SPSR.SPRF flag is 0), on termination of the serial transfer, the SPI copies data from the shift register to the receive buffer of the SPDR/SPDR_HA register.

The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMD0.SPB[3:0] bit settings. The polarity of the SSLni output pin depends on the SSLP register settings. For details on the SPI transfer format, see section 26.3.5. Transfer Formats.

(3) RSPCK delay (t1)

The RSPCK delay value of the SPI in master mode depends on the SPCMD0.SCKDEN bit setting and the SPCKD.SCKDL[2:0] bits setting. The SPI determines an RSPCK delay using the SPCMD0.SCKDEN bit and SPCKD.SCKDL[2:0] bits, as listed in Table 26.8. For a definition of RSPCK delay, see section 26.3.5. Transfer Formats.

Table 26.8 Relationship between the SPCMDm.SCKDEN bit, SPCKD.SCKDL[2:0] bits, and RSPCK delay

SPCMD0.SCKDEN bit	SPCKD.SCKDL[2:0] bits	RSPCK delay
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(4) SSL negation delay (t2)

The SSL negation delay value of the SPI in master mode depends on the SPCMD0.SLNDEN bit setting and the SSLND.SLNDL[2:0] bits setting. The SPI determines an SSL negation delay using the SPCMD0.SLNDEN bit and SSLND.SLNDL[2:0] bits, as listed in Table 26.9. For a definition of SSL negation delay, see section 26.3.5. Transfer Formats.

Table 26.9 Relationship between the SPCMDm.SLNDEN bit, SSLND.SLNDL[2:0] bits, and SSL negation delay

SPCMD0.SLNDEN bit	SSLND.SLNDL[2:0] bits	SSL negation delay
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(5) Next-access delay (t3)

The next-access delay value of the SPI in master mode depends on the SPCMD0.SPNDEN bit setting and the SPND.SPNDL[2:0] bits setting. The SPI determines a next-access delay during serial transfer using the SPCMD0.SPNDEN bit and SPND.SPNDL[2:0] bits, as listed in Table 26.10. For a definition of next-access delay, see section 26.3.5. Transfer Formats.

Table 26.10 Relationship between the SPCMDm.SPNDEN bit, SPND.SPNDL[2:0] bits, and next-access delay (1 of 2)

SPCMD0.SPNDEN bit	SPND.SPNDL[2:0] bits	Next-access delay
0	000b to 111b	1 RSPCK + 2 PCLKB

Table 26.10 Relationship between the SPCMDm.SPNDEN bit, SPND.SPNDL[2:0] bits, and next-access delay (2 of 2)

SPCMD0.SPNDEN bit	SPND.SPNDL[2:0] bits	Next-access delay
1	000b	1 RSPCK + 2 PCLKB
	001b	2 RSPCK + 2 PCLKB
	010b	3 RSPCK + 2 PCLKB
	011b	4 RSPCK + 2 PCLKB
	100b	5 RSPCK + 2 PCLKB
	101b	6 RSPCK + 2 PCLKB
	110b	7 RSPCK + 2 PCLKB
	111b	8 RSPCK + 2 PCLKB

(6) Initialization flow

Figure 26.31 shows an example of SPI initialization flow when the SPI is in master mode. For information on how to set up the Interrupt Controller Unit (ICU), and I/O ports, see the descriptions given in the individual blocks.

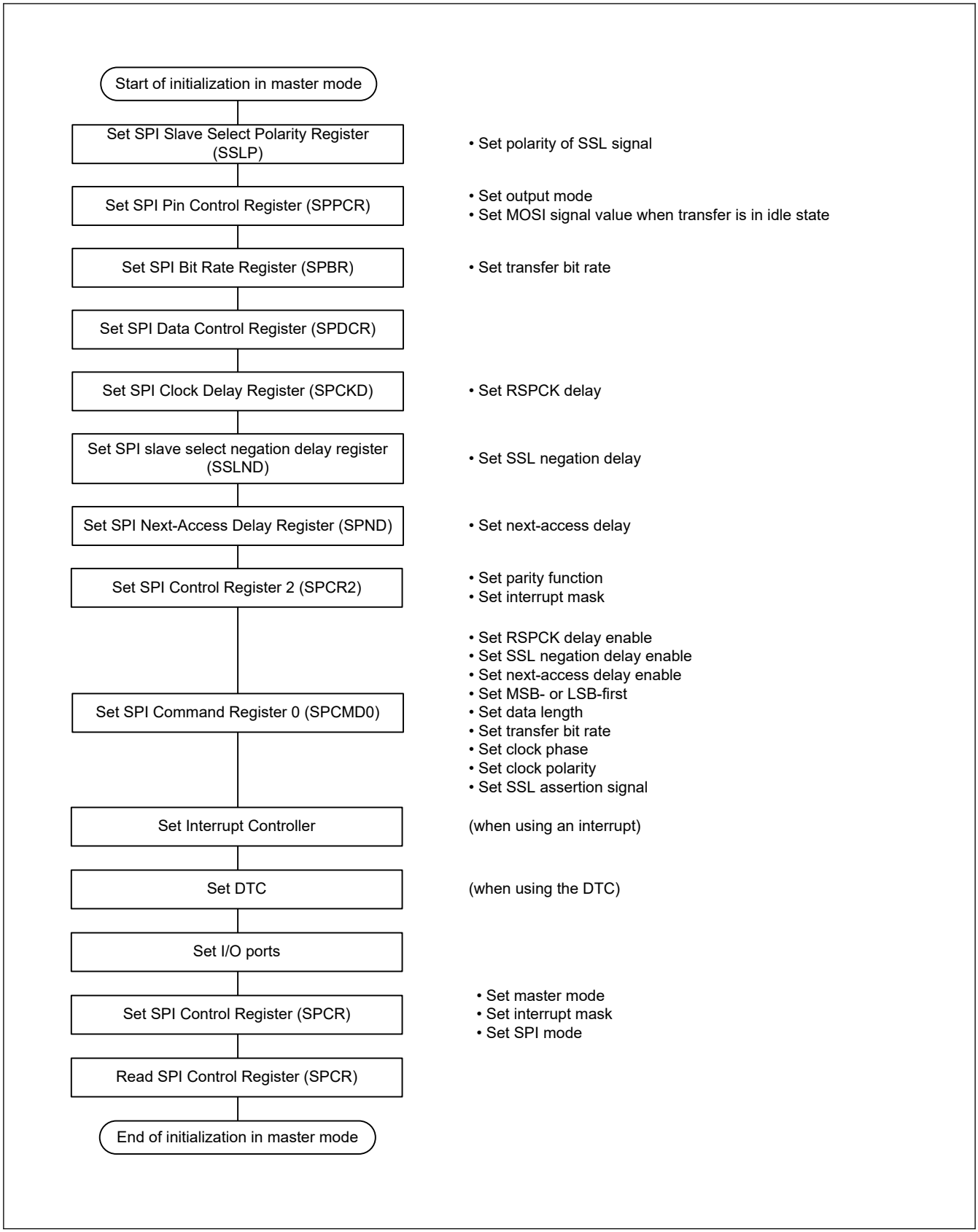


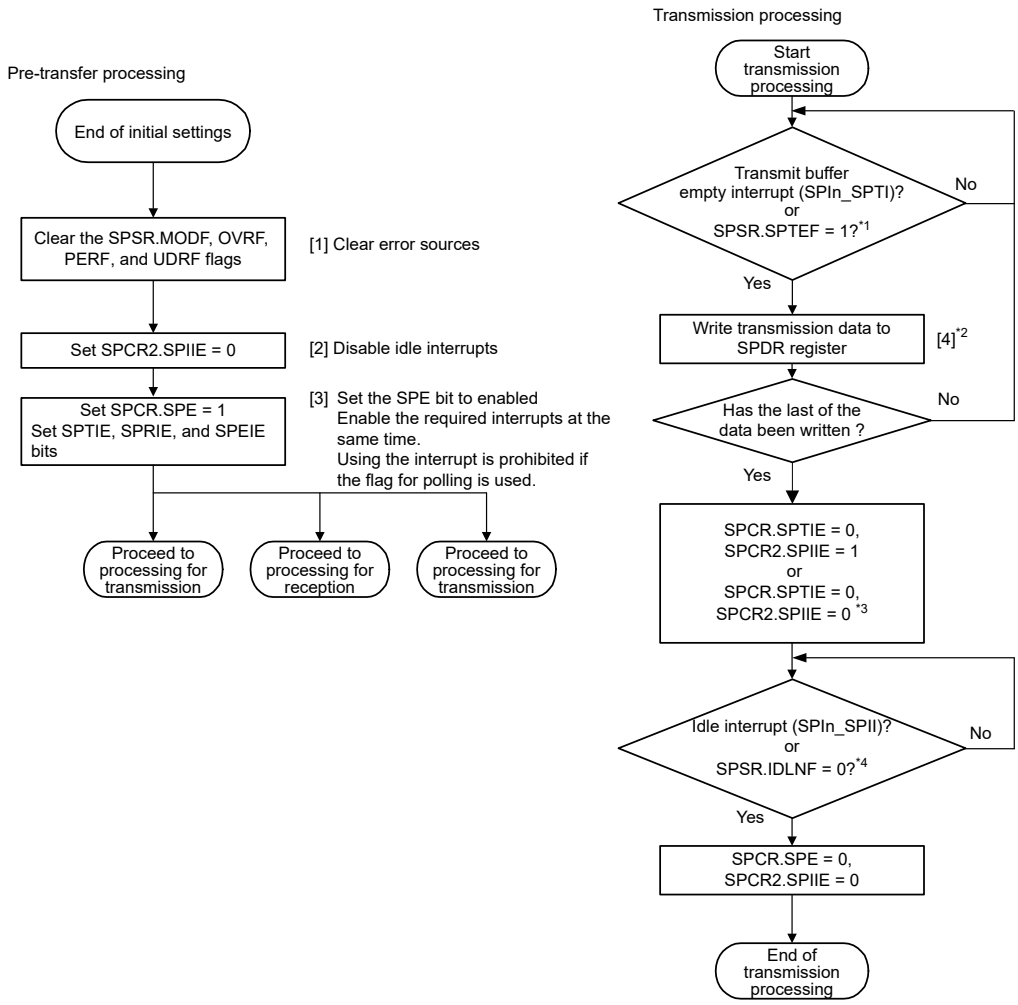
Figure 26.31 Example of initialization flow in master mode for SPI operation

(7) Software processing flow

Figure 26.32 to Figure 26.34 show examples of the software processing flow.

Transmit processing flow

When transmitting data, with the SPIi_SPII interrupt enabled, the CPU is notified of the completion of data transmission after the last data write for transmission.



Note 1. Before writing data for transmission to SPDR/SPDR_HA/SPDR_BY, check that the transmit buffer is empty by reading the SPSR.SPTEF flag, if the flag for polling is used.
Note 2. Setting SPCR2.SPIIE is prohibited if the flag for polling is used.
Note 3. Wait more than 1 PCLKB, after writing data for transmission to SPDR register, and before starting to poll SPSR.IDLNF flag, if the flag for polling is used.
Note 4. SPDR = SPDR/SPDR_HA/SPDR_BY

Figure 26.32 Transmission flow in master mode

Receive processing flow

The SPI cannot handle receive-only operation. Even when there is no data to transmit, it is necessary to transmit dummy data.

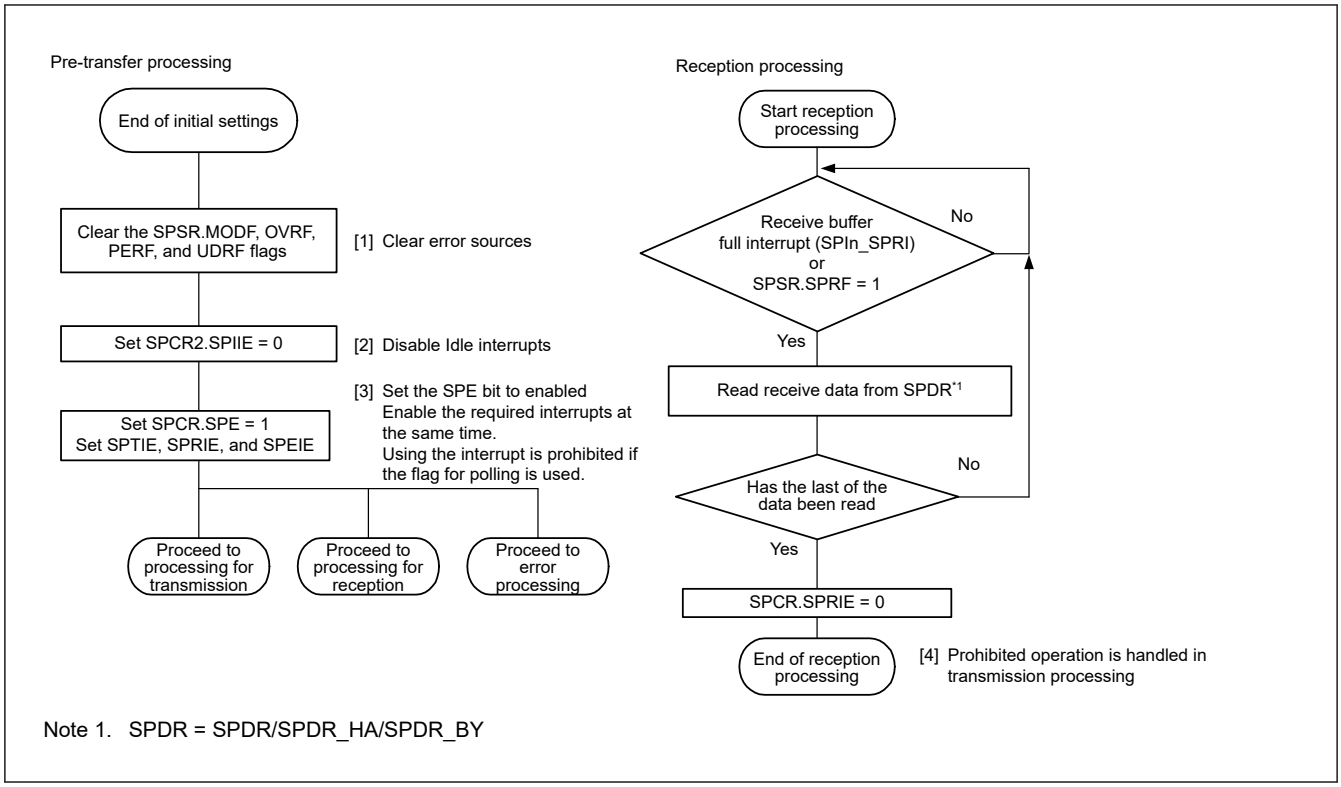


Figure 26.33 Reception flow in master mode

Error processing flow

The SPI detects the following errors:

- Mode fault error
- Underrun error
- Overrun error
- Parity error

When a mode fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. Then Renesas recommends clearing the SPCR.SPE bit to stop operations for errors other than mode fault errors.

When an error is detected using an interrupt, clear the ICU.IELSRn.IR flag in the error processing routine. If this is not done, the ICU.IELSRn.IR flag might continue to indicate the SPIi_SPTI or SPIi_SPRI interrupt request. If the SPIi_SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

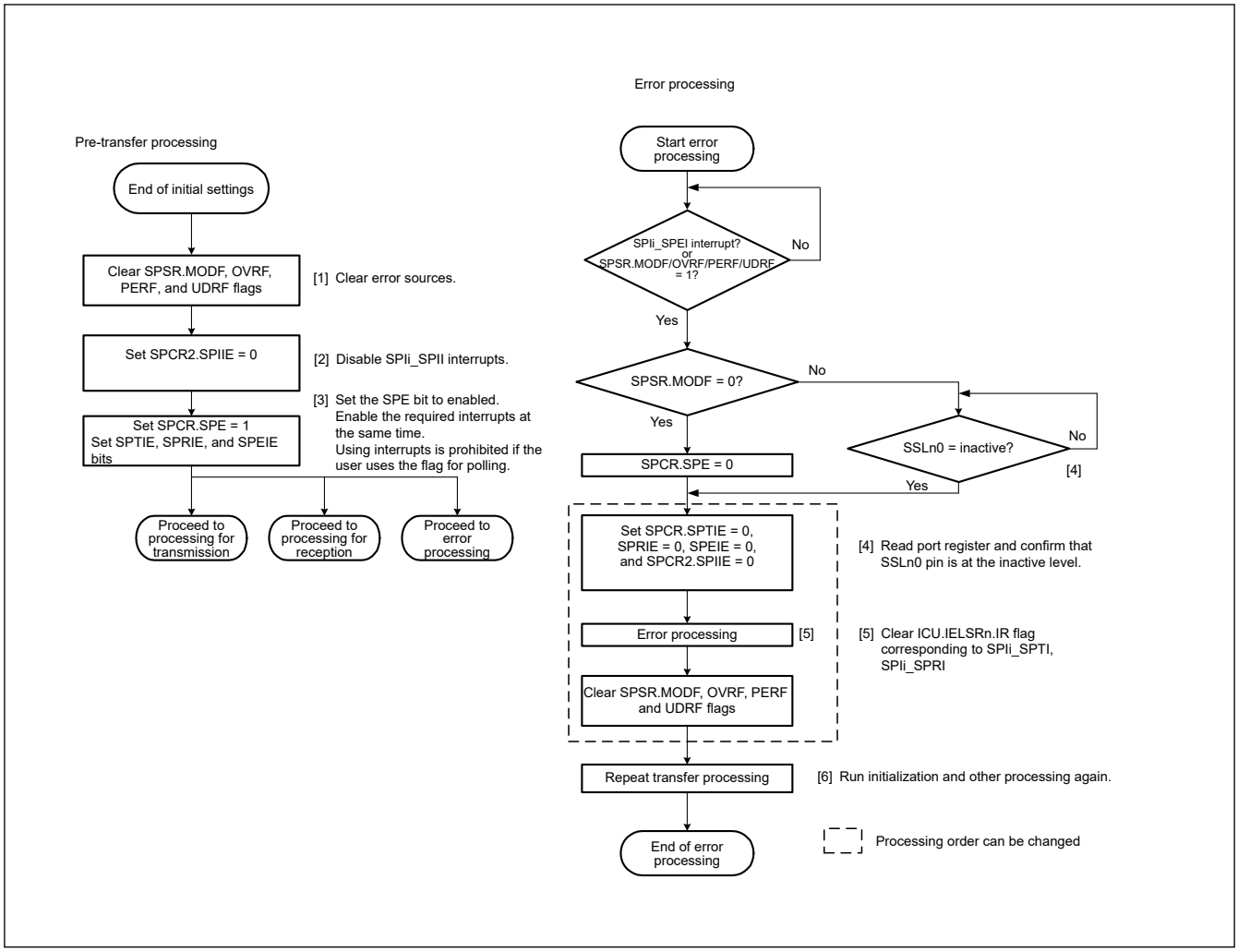


Figure 26.34 Error processing flow in master mode

26.3.10.2 Slave mode operation

(1) Starting a serial transfer

When the SPCMD0.CPHA bit is 0, if the SPI detects an SSLn0 input signal assertion, it must drive valid data to the MISOn output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLn0 input signal triggers the start of a serial transfer.

When the CPHA bit is 1, if the SPI detects the first RSPCKn edge in an SSLn0 signal asserted condition, it must drive valid data to the MISOn output signal. For this reason, when the CPHA bit is 1, the first RSPCKn edge in an SSLn0 signal asserted condition triggers the start of a serial transfer.

Regardless of the CPHA bit setting, the SPI drives the MISOn output signal on SSLn0 signal assertion. The data that is output by the SPI is either valid or invalid, depending on the CPHA bit setting.

For details on the SPI transfer format, see section 26.3.5. Transfer Formats. The polarity of the SSLn0 input signal depends on the SSLP.SSL0P setting.

(2) Terminating a serial transfer

Regardless of the SPCMD0.CPHA bit setting, the SPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPSR.SPRF flag is 0), on termination of serial transfer, the SPI copies received data from the shift register to the receive buffer of the SPDR/SPDR_HA register. On termination of a serial transfer, the SPI changes the status of the shift register to empty, regardless of the receive buffer state. A mode fault error occurs if the SPI detects an SSLn0 input signal negation from the beginning of serial transfer to the end of serial transfer (see section 26.3.8. Error Detection).

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the SPI data length is determined by the SPCMD0.SPB[3:0] bits setting. The polarity of the SSLn0 input signal is determined by the SSLP.SSL0P bit setting. For details on the SPI transfer format, see section 26.3.5. Transfer Formats.

(3) Notes on single-slave operations

If the SPCMD0.CPHA bit is 0, the SPI starts serial transfers when it detects the assertion edge for an SSLn0 input signal. In the configuration shown in Figure 26.7, if the SPI is used in single-slave mode, the SSLn0 signal is fixed at an active state. Therefore, when the CPHA bit is set to 0, the SPI cannot correctly start a serial transfer. For the SPI to correctly execute transmit and receive operations in slave mode when the SSLn0 input signal is fixed at an active state, the CPHA bit must be set to 1. Do not fix the SSLn0 input signal if there is a requirement for setting the CPHA bit to 0.

(4) Burst transfer

If the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSLn0 input signal. When the CPHA bit is 1, the serial transfer period is the period from the first RSPCKn edge to the sampling timing for the reception of the final bit in an SSLn0 signal active state. Even when the SSLn0 input signal remains at the active level, the SPI can accommodate burst transfers, because it can detect the start of an access.

When the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

(5) Initialization flow

Figure 26.35 shows an example of initialization flow for SPI operation when the SPI is in slave mode. For a description of how to set up the ICU, DTC, and I/O ports, see the descriptions given in the individual blocks.

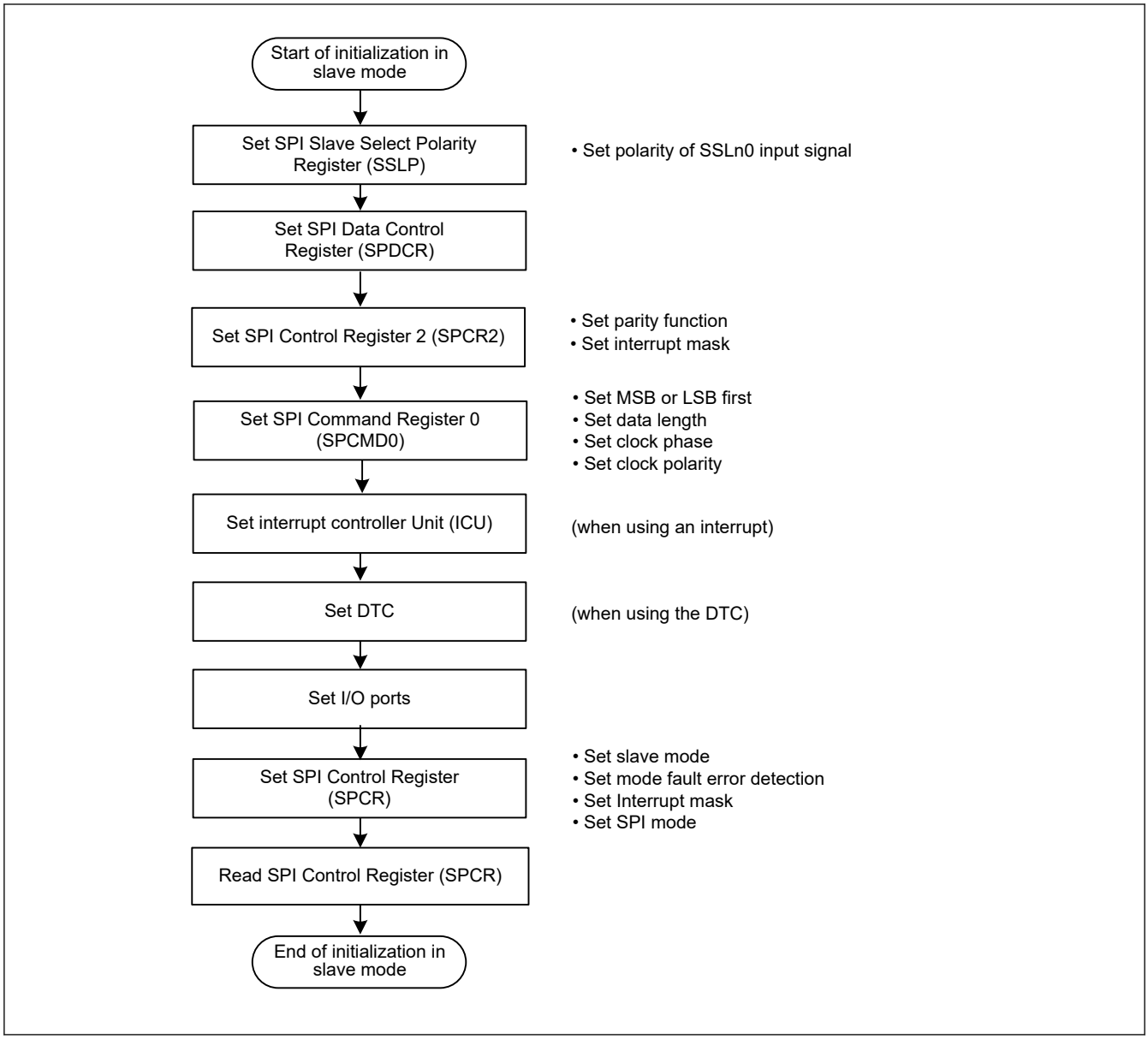
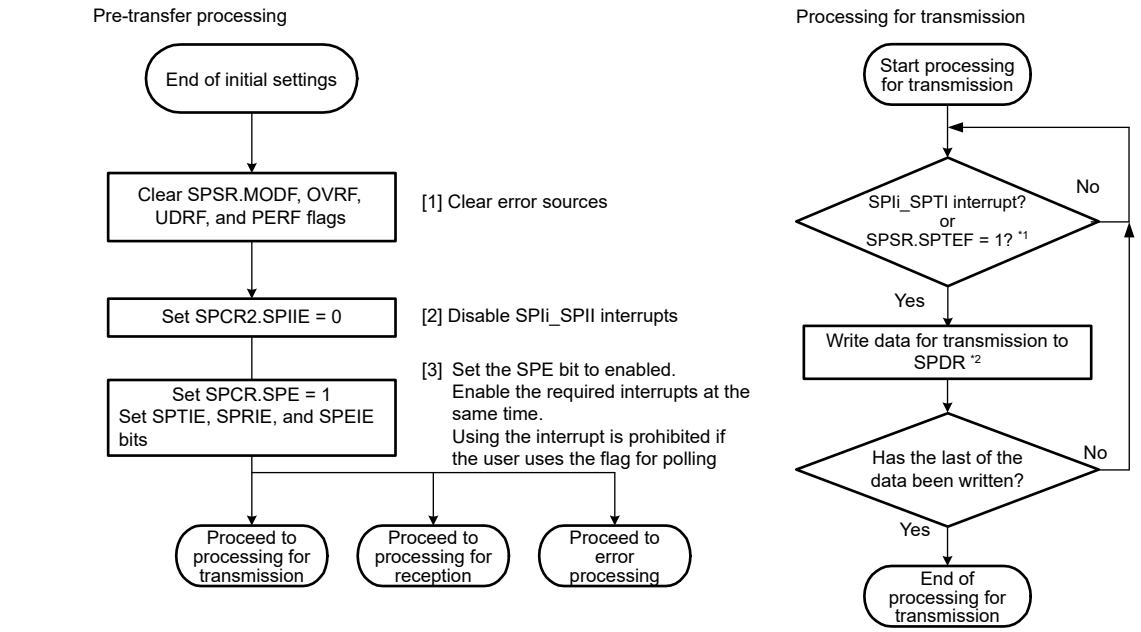


Figure 26.35 Example initialization flow in slave mode for SPI operation

(6) Software processing flow

Transmit processing flow

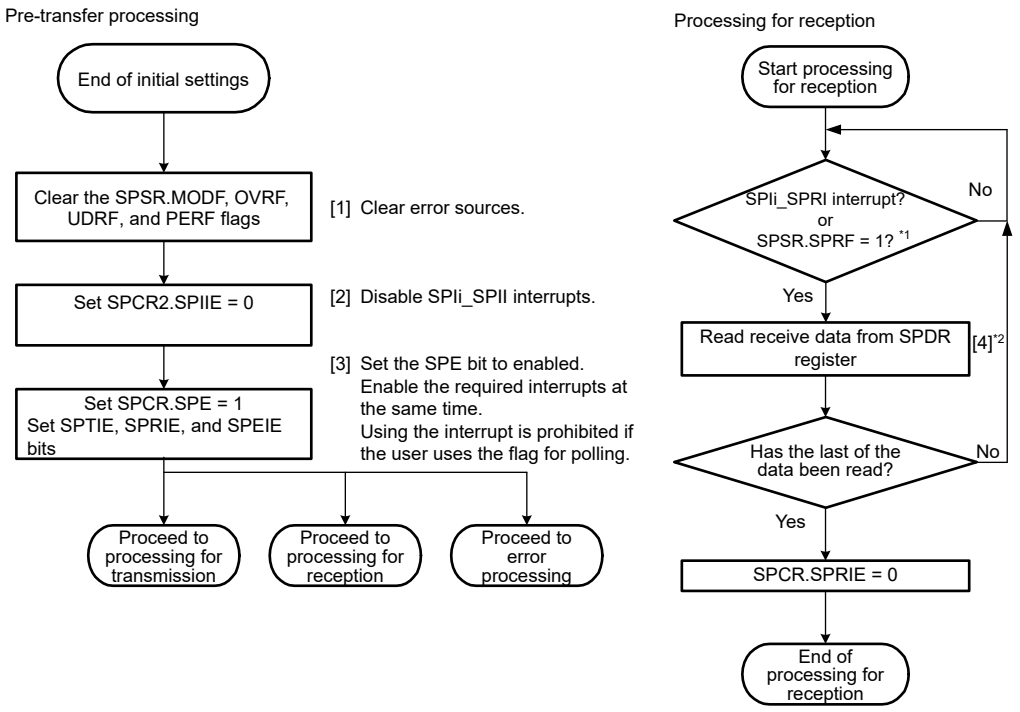


Note 1. Before writing data for transmission to SPDR register, check the transmit buffer empty interrupt by reading SPSR.SPTEF flag, if the flag for polling is used.
Note 2. SPDR = SPDR/SPDR_HA/SPDR_BY

Figure 26.36 Transmission flow in slave mode

Receive processing flow

The SPI does not handle receive-only operation, so processing for transmission is required.



Note 1. Before writing data for transmission to the SPDR register, check that the transmit buffer is empty by reading the SPSR.SPTEF flag, if the flag for polling is used.
Note 2. SPDR = SPDR/SPDR_HA/SPDR_BY

Figure 26.37 Reception flow in slave mode

Error processing flow

In slave mode operation, even when a mode fault error is generated, the SPSR.MODF flag can be cleared regardless of the state of the SSLn0 pin.

When an error is detected by using an interrupt, clear the ICU.IELSRn.IR flag in the error processing routine. If this is not done, the ICU.IELSRn.IR flag might continue to indicate the SPIi_SPTI or SPIi_SPRI interrupt request. If the SPIi_SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

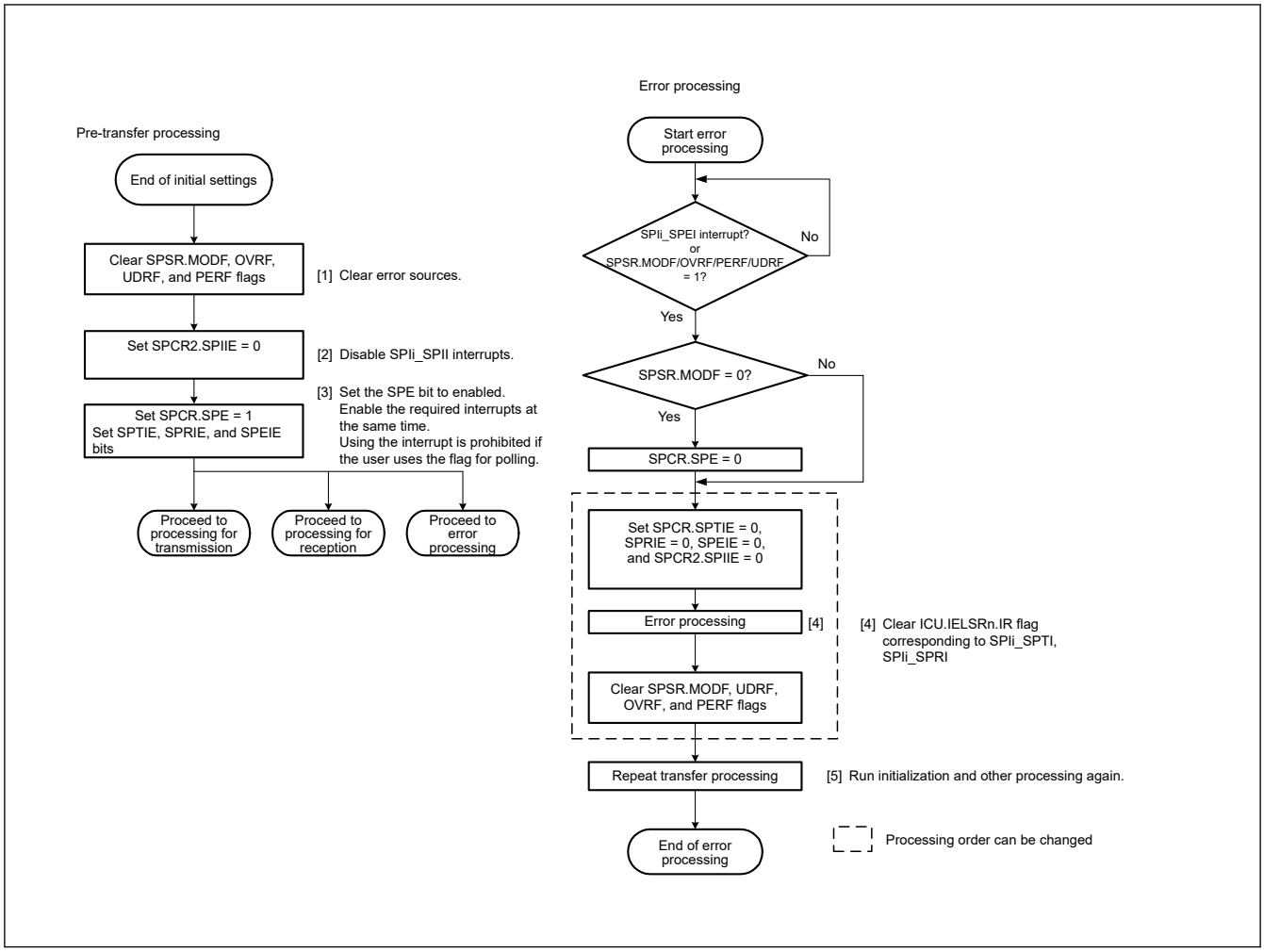


Figure 26.38 Error processing flow for slave mode

26.3.11 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the SPI. In clock synchronous operation, the SSLni pin is not used, and the RSPCKn, MOSIn, and MISOn pins handle communications. All SSLni pins are available as I/O port pins.

Although clock synchronous operation does not require the use of the SSLni pin, operation of the module is the same as in SPI operation. In both master mode and slave mode operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected, because the SSLni pin is not used.

Additionally, do not perform operation if clock synchronous operation is enabled when the SPCMD0.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

26.3.11.1 Master mode operation

(1) Starting serial transfer

The SPI updates the data in the transmit buffer (SPTX) of SPDR/SPDR_HA when data is written to the SPDR/SPDR_HA register with the transmit buffer empty, the data for the next transfer not set and the SPSR.SPTEF flag is 1. When the shift register is empty the SPI copies data from the transmit buffer to the shift register and starts serial transmission. On copying transmit data to the shift register, the SPI changes the status of the shift register to full, and on termination of serial transfer, it changes the status of the shift register to empty. The status of the shift register cannot be referenced.

Transfer in clock synchronous operation is conducted without the SSLn0 output signal. For details on the SPI transfer format, see section 26.3.5. Transfer Formats.

(2) Terminating serial transfer

The SPI terminates the serial transfer after transmitting an RSPCKn edge corresponding to the sampling timing. If free space is available in the receive buffer (the SPSR.SPRF flag is 0), on termination of serial transfer, the SPI copies data from the shift register to the receive buffer of the SPI Data Register (SPDR/SPDR_HA).

The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMD0.SPB[3:0] bits setting. Transfer in clock synchronous operation is conducted without the SSLn0 output signal. For details on the SPI transfer format, see section 26.3.5. Transfer Formats.

(3) Initialization flow

Figure 26.39 shows an example of initialization flow for clock synchronous operation when the SPI is used in master mode. For information on how to set up the ICU, DTC, and I/O ports, see the descriptions given in the individual blocks.

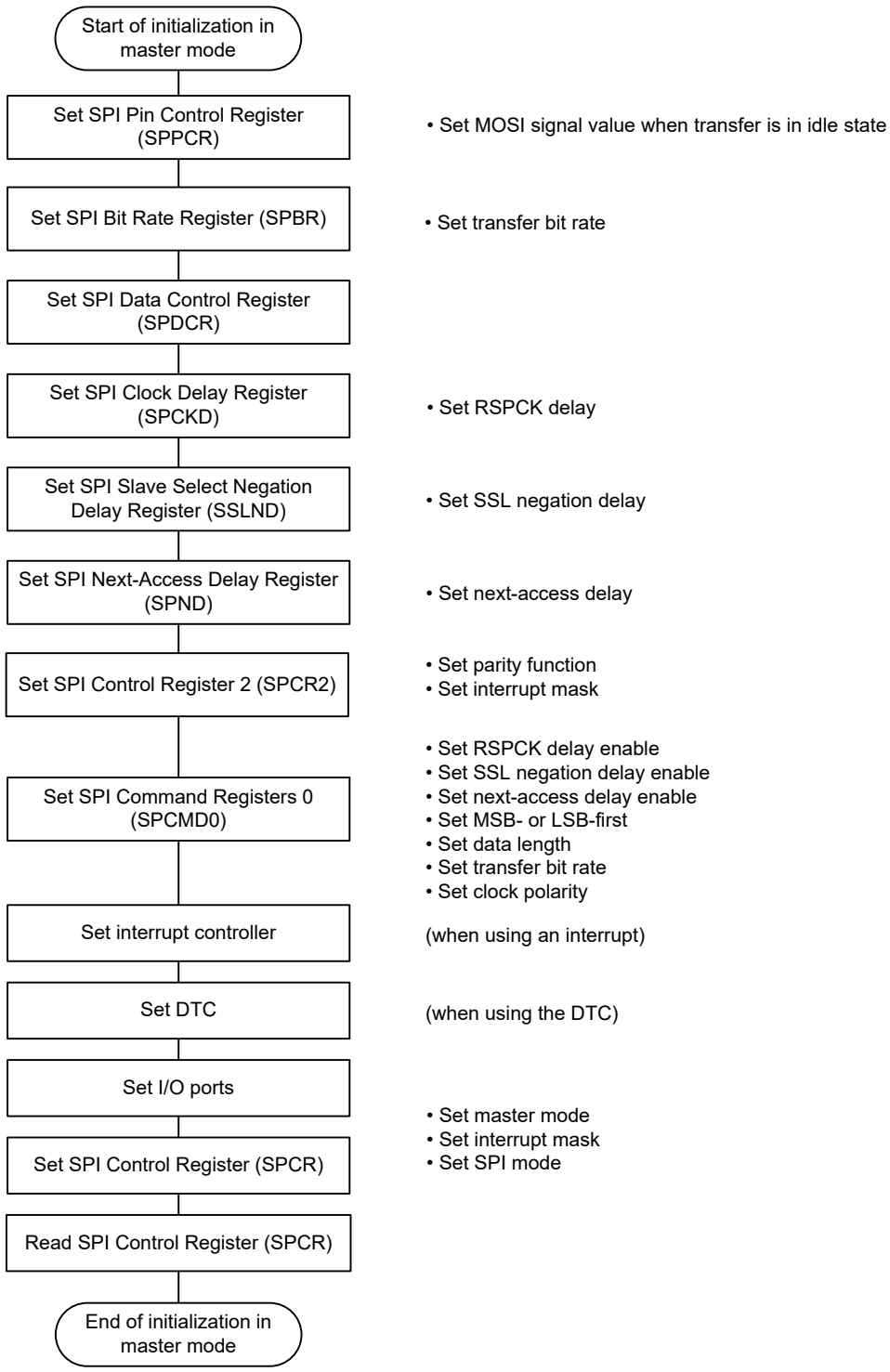


Figure 26.39 Example of initialization flow in master mode for clock synchronous operation

(4) Software processing flow

Software processing during clock synchronous master operation is the same as that for SPI master operation. For details, see (9) Software processing flow in section 26.3.10.1. Master mode operation. Mode fault errors do not occur in clock synchronous operation.

26.3.11.2 Slave mode operation

(1) Starting serial transfer

When the SPCR.SPMS bit is 1, the first RSPCKn edge triggers the start of a serial transfer in the SPI, and the SPI drives the MISOn output signal. The SSL0 input signal is not used in clock synchronous operation. For details on the SPI transfer format, see section 26.3.5. Transfer Formats.

(2) Terminating serial transfer

The SPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPSR.SPRF flag is 0), on termination of serial transfer, the SPI copies received data from the shift register to the receive buffer of the SPDR/SPDR_HA register. On termination of a serial transfer, the SPI changes the status of the shift register to empty regardless of the receive buffer.

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the SPI data length depends on the SPCMD0.SPB[3:0] bits setting. For details on the SPI transfer format, see section 26.3.5. Transfer Formats.

(3) Initialization flow

Figure 26.40 shows an example of initialization flow for clock synchronous operation when the SPI is used in slave mode. For a description of how to set up the ICU, DTC, and I/O ports, see the descriptions given in the individual blocks.

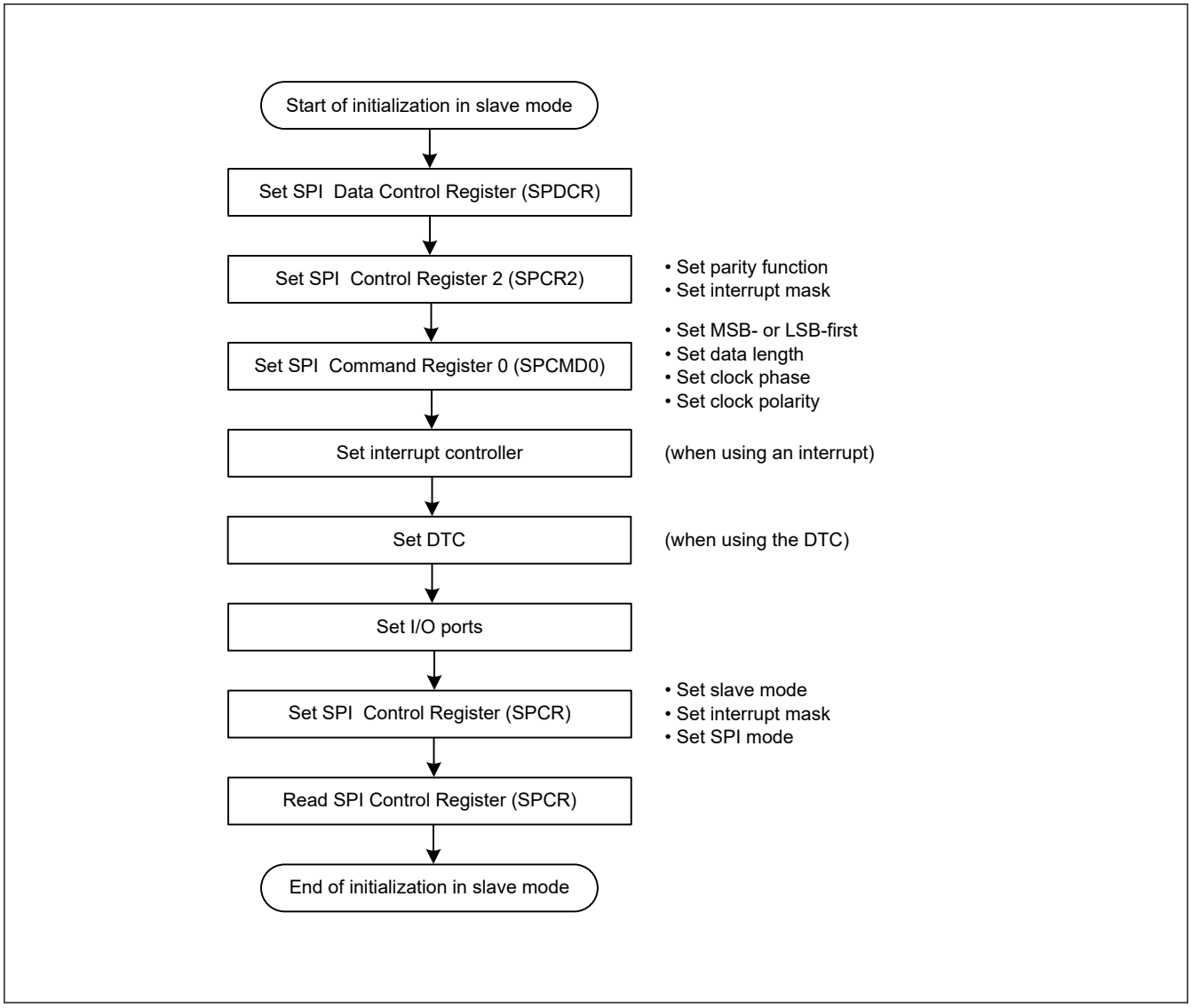


Figure 26.40 Example of initialization flow in slave mode for clock synchronous operation

(4) Software processing flow

Software processing during clock synchronous slave operation is the same as that for SPI slave operation. For details, see (6) Software processing flow. Mode fault errors do not occur in clock synchronous mode.

26.3.12 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the SPI shuts off the path between the MISOn pin and the shift register if the SPCR.MSTR bit is 1, or between the MOSIn pin and the shift register if the SPCR.MSTR bit is 0, and connects the input and output paths of the shift register, establishing a loopback mode. The SPI does not shut off the path between the MOSIn pin and the shift register if the SPCR.MSTR bit is 1, or between the MISOn pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the SPI or the reversed transmit data becomes the received data for the SPI.

Table 26.11 lists the relationship between the SPLP2 and SPLP bits and the received data. Figure 26.41 shows the configuration of the shift register I/O paths when the SPI in master mode is set to loopback mode (SPPCR.SPLP2 = 0, SPPCR.SPLP = 1).

Table 26.11 SPLP2 and SPLP bit settings and received data

SPPCR.SPLP2 bit	SPPCR.SPLP bit	Received data
0	0	Input data from the MOSIn pin or MISOn pin
0	1	Inverted transmit data
1	0	Transmit data
1	1	Transmit data

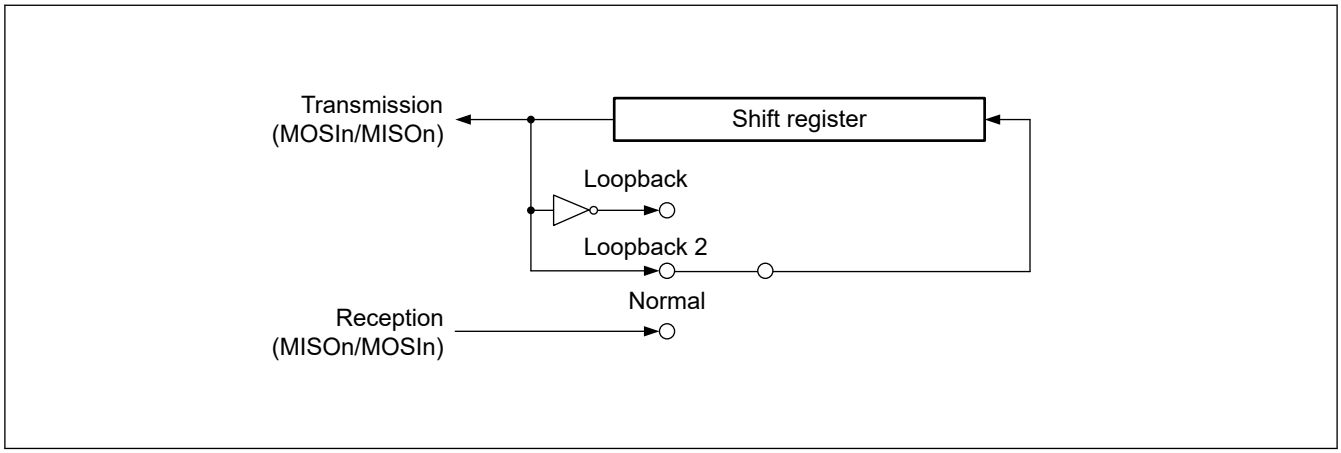


Figure 26.41 Configuration of shift register I/O paths in loopback mode for master mode

26.3.13 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. To detect defects in the parity bit adding unit and error detecting unit, the parity circuit performs self-diagnosis as shown in Figure 26.42.

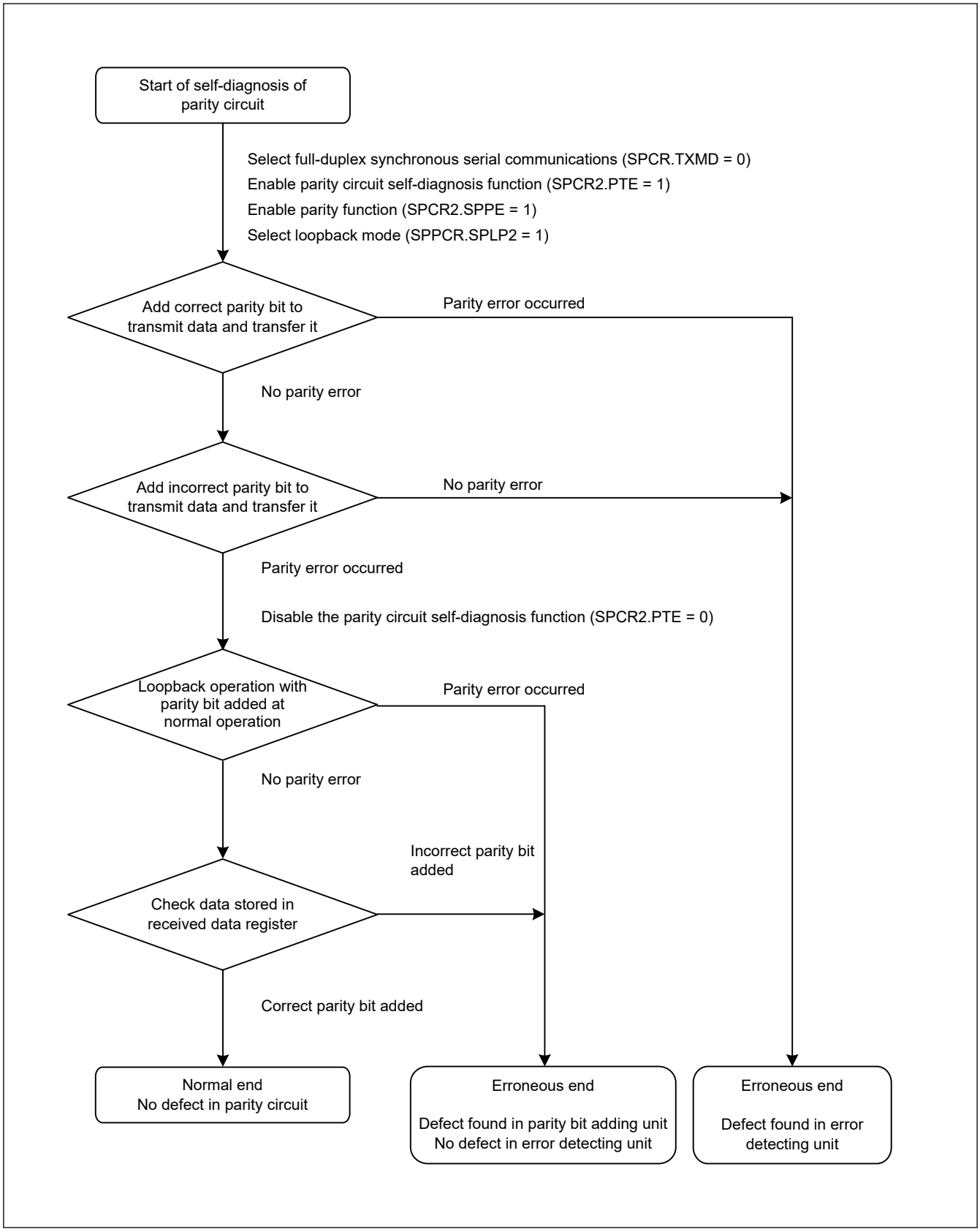


Figure 26.42 Self-diagnosis flow for parity circuit

26.3.14 Interrupt Sources

The SPI has the following interrupt sources:

- Receive buffer full
- Transmit buffer empty
- SPI error (mode-fault, underrun, overrun, or parity error)
- SPI idle
- Transmission-complete

The DTC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Because the vector address for the SPIi_SPEI (SPI error interrupt) is allocated to interrupt requests on mode-fault, underrun, overrun, and parity errors, the actual interrupt source must be determined from the flags. Interrupt sources for the SPI are listed in Table 26.12. An interrupt is generated on satisfaction of one of the interrupt conditions in Table 26.12. Clear the receive buffer full and transmit buffer empty sources through a data transfer.

When using the DTC to perform data transmission and reception, you must first set up the DTC to be in a transfer-enabled status before setting the SPI. For information on setting up the DTC, see section 15, Data Transfer Controller (DTC).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt occur while the ICU.IELSRn.IR flag is 1, the interrupt is not output as a request for the ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request is output when the ICU.IELSRn.IR flag becomes 0. A retained interrupt request is automatically discarded when it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be set to 0.

Table 26.12 SPI interrupt sources

Interrupt source	Symbol	Interrupt condition	DTC activation
Receive buffer full	SPIi_SPRI	The receive buffer becomes full (SPSR.SPRF flag is 1) while the SPCR.SPRIE bit is 1	Possible
Transmit buffer empty	SPIi_SPTI	The transmit buffer becomes empty (SPSR.SPTEF flag is 1) while the SPCR.SPTIE bit is 1	Possible
SPI error (mode-fault, underrun, overrun, or parity error)	SPIi_SPEI	The SPSR.MODF, OVRF, UDRF or PERF flag sets to 1 while the SPCR.SPEIE bit is 1	Impossible
SPI idle	SPIi_SPII	The SPSR.IDLNF flag sets to 0 while the SPCR2.SPIIE bit is 1	Impossible
Transmission-complete	SPIi_SPTEND	• Master mode: an interrupt is generated when the IDLNF flag (SPI idle flag) changes from 1 to 0 • Slave mode: an interrupt occurs on conditions shown in Table 26.14	Impossible

26.4 Event Link Controller Event Output

The Event Link Controller (ELC) can produce the following event output signals:

- Receive buffer full event output
- Transmit buffer empty event output
- Mode-fault, underrun, overrun, or parity error event output
- SPI idle event output
- Transmission-completed event output

The event link output signal is output regardless of the interrupt enable bit setting.

26.4.1 Receive Buffer Full Event Output

This event signal is output when received data is transferred from the shift register to the SPDR/SPDR_HA on completion of serial transfer.

26.4.2 Transmit Buffer Empty Event Output

This event signal is output when data for transmission is transferred from the transmit buffer to the shift register and when the value of the SPE bit changes from 0 to 1.

26.4.3 Mode-Fault, Underrun, Overrun, or Parity Error Event Output

This event signal is output when mode-fault, underrun, overrun, or parity error is detected. See section 26.5.4. Constraints on Mode-Fault, Underrun, Overrun, or Parity Error Event Output if using this event signal.

(1) Mode-fault

Table 26.13 lists the conditions for occurrence of a mode-fault event.

Table 26.13 Conditions for mode-fault occurrence

SPI mode	SPCR.MODFEN bit	SSLn0 pin	Remarks
SPI operation (SPMS = 0) Slave (SPCR.MSTR = 0)	1	Not active	Event is output only when the SSLn0 pin is deactivated during transmission

(2) Underrun

This event signal is output in response to an underrun when a serial transfer starts while the transmission data is not ready, and the value of the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1. Under these conditions, the MODF and UDRF flags are set to 1.

(3) Overrun

This event signal is output in response to an overrun when a serial transfer completes while the receive buffer contains unread data and the value of the SPCR.TXMD bit is 0. Under these conditions, the OVRF flag is set to 1.

(4) Parity error

This event signal is output in response to a parity error detected on completion of a serial transfer while the value of the TXMD bit in SPCR is 0 and the value of the SPPE bit in SPCR2 is 1.

26.4.4 SPI Idle Event Output

(1) In master mode

In master mode, an event is output when the condition for setting the IDLNF flag (SPI idle flag) to 0 is satisfied.

(2) In slave mode

In slave mode, an event is output when the SPCR.SPE bit is set to 0 (SPI is initialized).

26.4.5 Transmission-Completed Event Output

During both SPI and clock synchronous operations in master mode, an event is output when the IDLNF flag (SPI idle flag) changes from 1 to 0. Table 26.14 lists the conditions for occurrence of a transmission-completed event in slave mode.

Table 26.14 Conditions for generation of transmission-complete event in slave mode

Conditions	Transmit buffer state	Shift register state	Other
SPI operation (SPMS = 0)	Empty	Empty	Negation of SSLn0 input
Clock synchronous operation (SPMS = 1)	Empty	Empty	Edge detection of the last RSPCKn

Whether the operation is in master mode or slave mode, an event is not output if 0 is written to the SPCR.SPE bit in transmission or the SPCR.SPE bit is cleared by the mode-fault error or the underrun error.

26.5 Usage Notes

26.5.1 Settings for the Module-Stop State

The Module Stop Control Register B (MSTPCRB) can enable or disable the SPI operation. The SPI is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details on the Module Stop Control Register B, see section 10, Low Power Modes.

26.5.2 Constraint on Low-Power Functions

When using the module-stop function and entering a low-power mode other than Sleep mode, set the SPCR.SPE bit to 0 before completing communication.

26.5.3 Constraints on Starting Transfer

If the ICU.IELSRn.IR flag is 1 when transfer starts, the interrupt request is internally retained, which can lead to unanticipated behavior of the ICU.IELSRn.IR flag.

To prevent this, use the following procedure to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1):

1. Confirm that transfer stopped (the SPCR.SPE bit is 0).
2. Set the associated interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) to 0.
3. Read the associated interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) and confirm that its value is 0.
4. Set the ICU.IELSRn.IR flag to 0.

26.5.4 Constraints on Mode-Fault, Underrun, Overrun, or Parity Error Event Output

Using the mode-fault, underrun, overrun, or parity error event is prohibited if the SPI is in multi-master mode (when the SPCR.SPMS bit is 0, the SPCR.MSTR bit is 1, and the SPCR.MODFEN bit is 1).

26.5.5 Constraints on the SPSR.SPRF and SPSR.SPTEF Flags

If the polling flags, SPRF and SPTEF, are used, using the interrupts is prohibited, and you must set the SPCR.SPRIE and SPCR.SPTIE bits to 0. Either the interrupts or the flags can be used, but not both.

27. Cyclic Redundancy Check (CRC)

27.1 Overview

The Cyclic Redundancy Check (CRC) generates CRC codes to detect errors in the data. The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication. Additionally, various CRC-generation polynomials are available. The snoop function allows to monitor the access to specific addresses. This function is useful in applications that require CRC code to be generated automatically in certain events, such as monitoring writes to the serial transmit buffer and reads from the serial receive buffer.

Table 27.1 lists the CRC calculator specifications and Figure 27.1 shows a block diagram.

Table 27.1 CRC calculator specifications

Item	Description	
Data size	8-bit	32-bit
Data for CRC calculation*1	CRC code generated for data in 8n-bit units (where n is a natural number)	CRC code generated for data in 32n-bit units (where n is a natural number)
CRC processor unit	Operation executed on 8 bits in parallel	Operation executed on 32 bits in parallel
CRC generating polynomial	One of three generating polynomials that is selectable: [8-bit CRC] • $X^8 + X^2 + X + 1$ (CRC-8) [16-bit CRC] • $X^{16} + X^{15} + X^2 + 1$ (CRC-16) • $X^{16} + X^{12} + X^5 + 1$ (CRC-CCITT).	One of two generating polynomials that is selectable: [32-bit CRC] • $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ (CRC-32) • $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$ (CRC-32C).
CRC calculation switching	The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication	
Module-stop function	Module-stop state can be set to reduce power consumption	
CRC snoop	Monitor reads from and writes to a certain register address	—

Note 1. This function cannot divide data used in CRC calculations. Write data in 8-bit or 32-bit units.

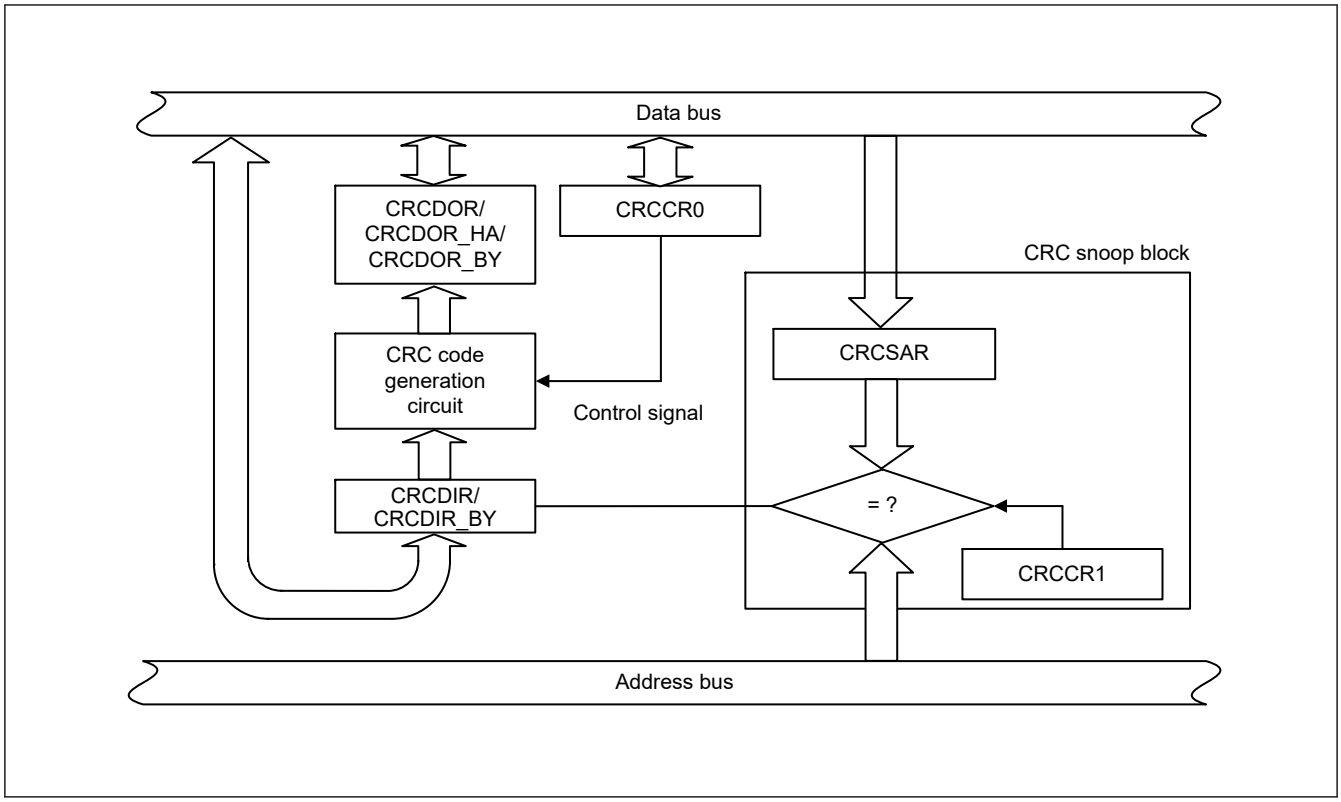


Figure 27.1 CRC calculator block diagram

27.2 Register Descriptions

27.2.1 CRCCR0 : CRC Control Register 0

Base address: CRC = 0x4007_4000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DORCLR	LMS	—	—	—	GPS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	GPS[2:0]	CRC Generating Polynomial Switching 0 0 1: 8-bit CRC-8 ($X^8 + X^2 + X + 1$) 0 1 0: 16-bit CRC-16 ($X^{16} + X^{15} + X^2 + 1$) 0 1 1: 16-bit CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) 1 0 0: 32-bit CRC-32 ($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$) 1 0 1: 32-bit CRC-32C ($X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$) Others: No calculation is executed	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	LMS	CRC Calculation Switching 0: Generate CRC code for LSB-first communication 1: Generate CRC code for MSB-first communication	R/W
7	DORCLR	CRCDOR/CRCDOR_HA/CRCDOR_BY Register Clear 0: No effect 1: Clear the CRCDOR/CRCDOR_HA/CRCDOR_BY register	W

GPS[2:0] bits (CRC Generating Polynomial Switching)

The GPS[2:0] bits select the CRC generating polynomial.

LMS bit (CRC Calculation Switching)

The LMS bit selects the bit order of generated CRC code. Transmit the lower byte of the CRC code first for LSB-first communication and the upper byte first for MSB-first communication. For details on transmitting and receiving CRC code, see section 27.3. Operation.

DORCLR bit (CRCDOR/CRCDOR_HA/CRCDOR_BY Register Clear)

Write 1 to the DORCLR bit to set the CRCDOR/CRCDOR_HA/CRCDOR_BY register to 0x00000000. This bit is read as 0. Only 1 can be written to it.

27.2.2 CRCCR1 : CRC Control Register 1

Base address: CRC = 0x4007_4000

Offset address: 0x01

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

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	CRCSWR	Snoop-On-Write/Read Switch 0: Snoop-on-read 1: Snoop-on-write	R/W
7	CRCSEN	Snoop Enable 0: Disabled 1: Enabled	R/W

CRCSWR bit (Snoop-On-Write/Read Switch)

The CRCSWR bit selects the direction of access in the CRC snoop function.

When this bit is set to 0 (initial value), the CRC snoop operation to read a specific register is enabled. Similarly, when this bit is set to 1, the CRC snoop operation to write a specific register is enabled.

CRCSEN bit (Snoop Enable)

When the CRCSEN bit is set to 1, the CRC snoop operation is enabled. When this bit is set to 0, the CRC snoop operation is disabled.

27.2.3 CRCDIR/CRCDIR_BY : CRC Data Input Register

Base address: CRC = 0x4007_4000

Offset address: 0x04

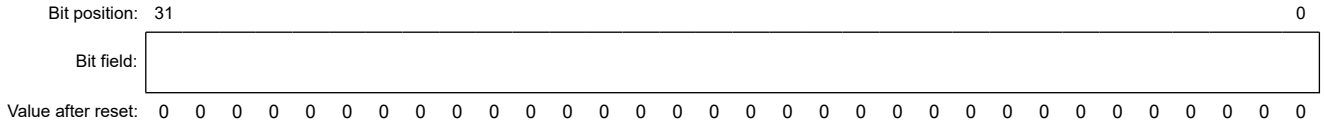
Bit position:	31	0
Bit field:		
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	n/a	CRC input data The CRCDIR register is a 32-bit read/write register to write data for CRC-32 or CRC-32C calculation. The CRCDIR_BY (CRCDIR[31:24], address: 0x4007_4004) is an 8-bit read/write register to write data for CRC-8, CRC-16, or CRC-CCITT calculation.	R/W

27.2.4 CRCDOR/CRCDOR_HA/CRCDOR_BY : CRC Data Output Register

Base address: CRC = 0x4007_4000

Offset address: 0x08

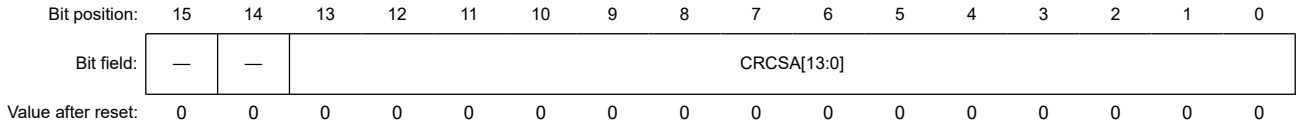


Bit	Symbol	Function	R/W
31:0	n/a	CRC output data The CRCDOR register is a 32-bit read/write register for CRC-32 or CRC-32C calculation. The CRCDOR_HA (CRCDOR[31:16], address: 0x4007_4008) register is a 16-bit read/write register for CRC-16 or CRC-CCITT calculation. The CRCDOR_BY (CRCDOR[31:24], address: 0x4007_4008) register is an 8-bit read/write register for CRC-8 calculation. Because its initial value is 0x00000000, rewrite the CRCDOR/CRCDOR_HA/CRCDOR_BY register to perform the calculations using a value other than the initial value. Data written to the CRCDIR/CRCDIR_BY register is CRC calculated and the result is stored in the CRCDOR/CRCDOR_HA/CRCDOR_BY register. If the CRC code is calculated following the transferred data and the result is 0x00000000, there is no CRC error.	R/W

27.2.5 CRCSAR : Snoop Address Register

Base address: CRC = 0x4007_4000

Offset address: 0x0C



Bit	Symbol	Function	R/W
13:0	CRCSA[13:0]	Register Snoop Address These bits store the TDR or RDR address in the SCI module to snoop	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

CRCSA[13:0] bits (Register Snoop Address)

The CRCSA[13:0] bits specify the lower address 14 bits of the register monitored by the CRC snoop operation.

Only the following addresses can be used for the CRCSA[13:0] bits:

- 0x4007_0003: SCI0.TDR, 0x4007_0005:SCI0.RDR
- 0x4007_0023: SCI1.TDR, 0x4007_0025:SCI1.RDR
- 0x4007_0043: SCI2.TDR, 0x4007_0045: SCI2.RDR
- 0x4007_0123: SCI9.TDR, 0x4007_0125:SCI9.RDR
- 0x4007_000F: SCI0.FTDRL, 0x4007_0011: SCI0.FRDRL.

27.3 Operation

27.3.1 Basic Operation

The CRC calculator generates CRC codes for use in LSB-first or MSB-first transfer.

The following examples show CRC code generation for input data (0xF0) using the 16-bit CRC-CCITT generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC Data Output Register (CRCDOR_HA) is cleared before CRC calculation.

When an 8-bit CRC (with the polynomial $X^8 + X^2 + X + 1$) is in use, the valid bits of the CRC code are obtained in CRCDOR_BY. When a 32-bit CRC is in use, the valid bits of the CRC code are obtained in CRCDOR.

Figure 27.2 and Figure 27.3 show the LSB-first and MSB-first data transmission examples respectively. Figure 27.4 and Figure 27.5 show the LSB-first and MSB-first data reception examples.

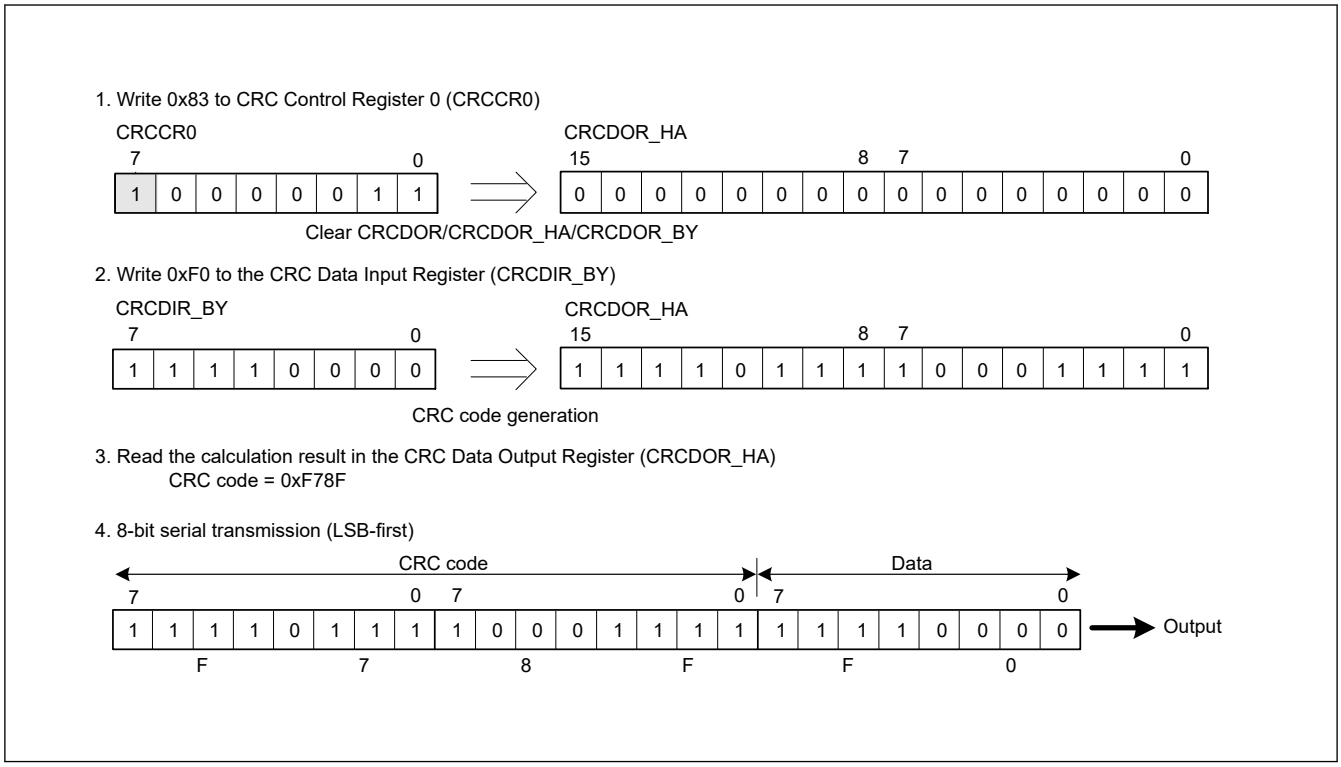


Figure 27.2 LSB-first data transmission

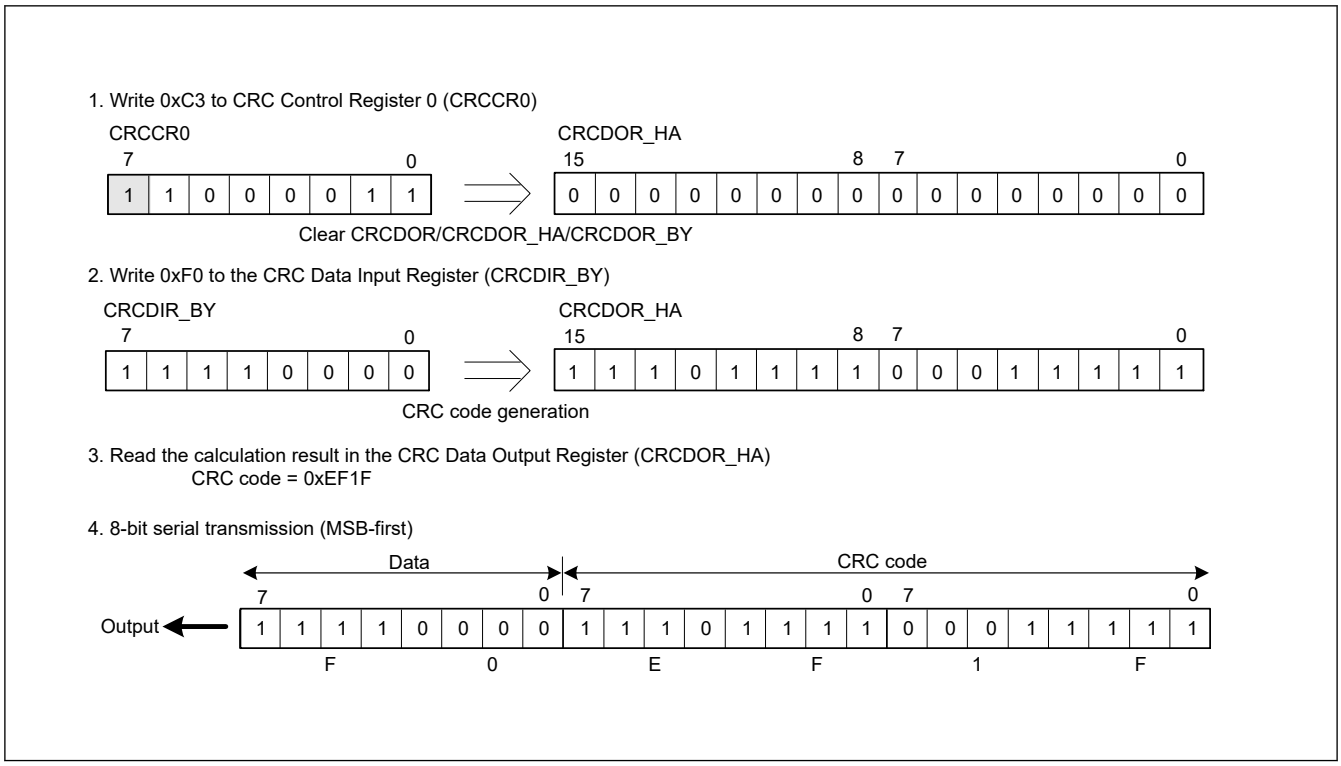


Figure 27.3 MSB-first data transmission

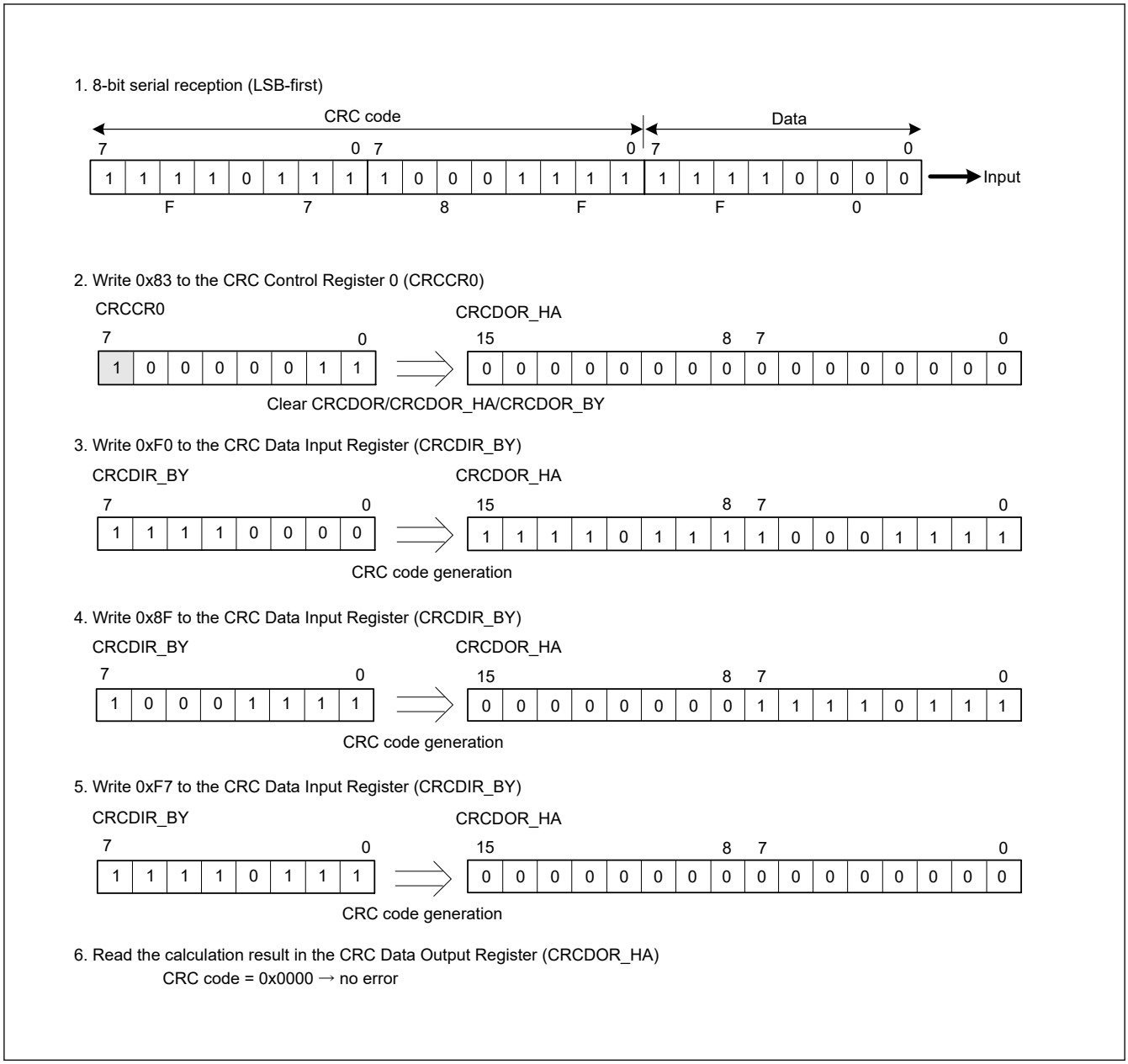


Figure 27.4 LSB-first data reception

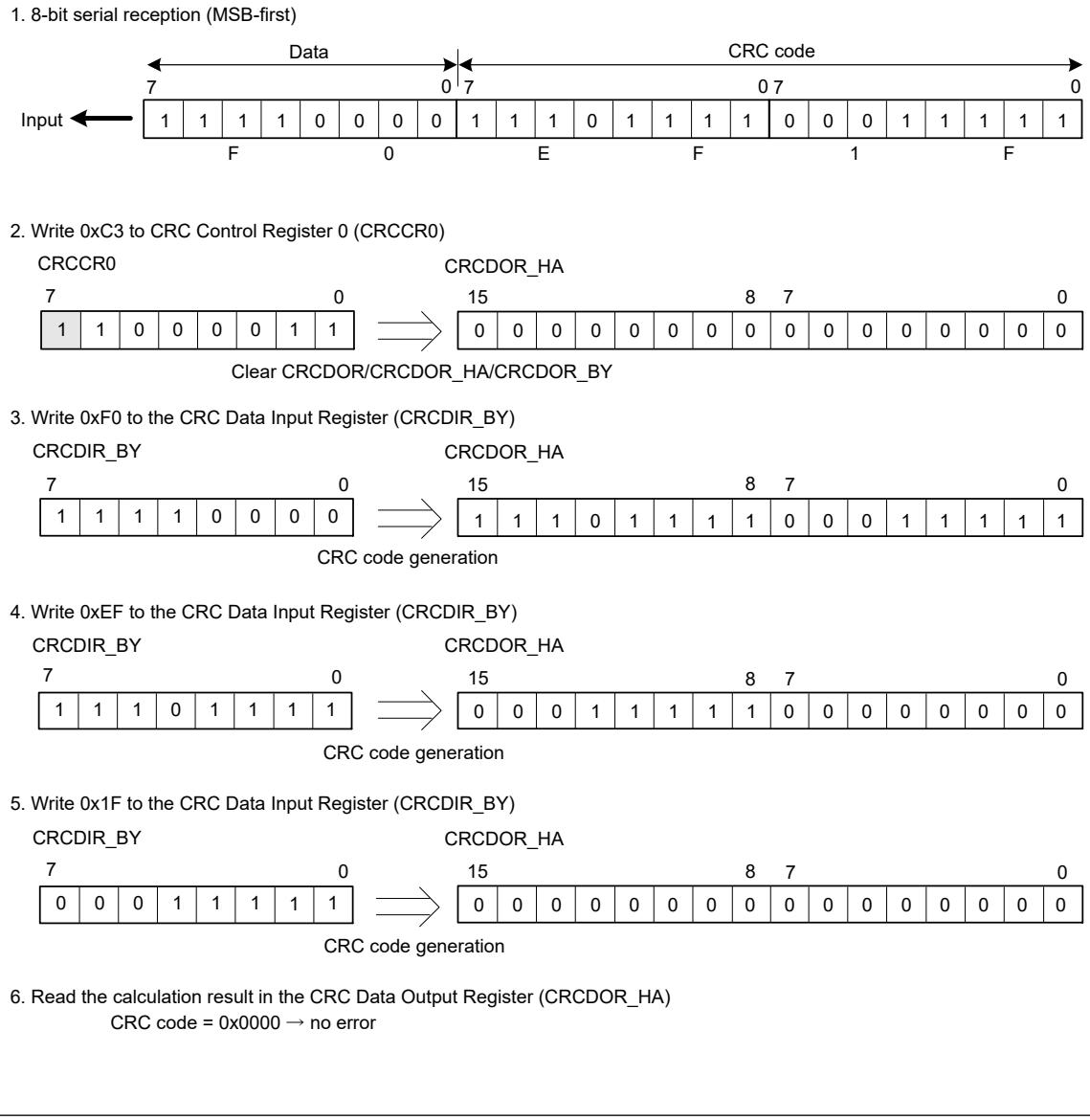


Figure 27.5 MSB-first data reception

27.3.2 CRC Snoop Function

The CRC snoop function monitors reads from and writes to a specific register and performs CRC calculation on the monitored data automatically. Because the CRC snoop function recognizes writes to and reads from a specific register address as a trigger to automatically perform CRC calculation, there is no need to write data to the CRCDIR_BY register. All I/O register specified in the section 27.2.5. CRCSAR : Snoop Address Register are subject to the CRC snoop. The CRC snoop is useful in monitoring writes to the SCIn.TDR (n = 0 to 2, 9) register, and reads from the SCIn.RDR (n = 0 to 2, 9) register.

To use this function, write the lower address 14 bits of a specific register to bits CRCSA13 to CRCSA0 in the CRCSAR register, and set CRCSEN bit in the CRCCR1 register to 1. Then, set the CRCSWR bit in the CRCCR1 register to 1 to enable snooping on writes to the target register, or set the CRCSWR bit in the CRCCR1 register to 0 to enable snooping on reads from the target register. It is possible that access to a target I/O register may be executed before the CRCSWR bit write completed. In this case, the data is not stored in the CRCDIR register. To avoid this issue, before accessing I/O register, read back the CRCSWR bit that was written to confirm that the write completed.

When both the CRCSEN and CRCSWR bits are set to 1, and data is written to a target register in a bus master module such as the CPU and DTC, the CRC calculator stores the data in the CRCDIR_BY register and performs CRC calculation.

Similarly, when the CRCSEN bit is set to 1, CRCSWR bit to 0, and data is read from a target register in a bus master module such as the CPU and DTC, the CRC calculator stores the data in the CRCDIR_BY register and performs CRC calculations.

When the CRC code is generated by using CRC-8, CRC-16, and CRC-CCITT generating polynomial, the target register is accessed in 1 byte (8 bits). Similarly, when the CRC code is generated by using CRC-32 and CRC-32C generating polynomial, the target register is accessed in words (32 bits).

27.4 Usage Notes

27.4.1 Settings for the Module-Stop State

The Module Stop Control Register C (MSTPCRC) can enable or disable CRC calculator operation. The CRC calculator is initially stopped after a reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

27.4.2 Note on Transmission

The transmission sequence for the CRC code differs based on whether the transmission is LSB-first or MSB-first. Figure 27.6 shows an LSB-first and MSB-first data transmission.

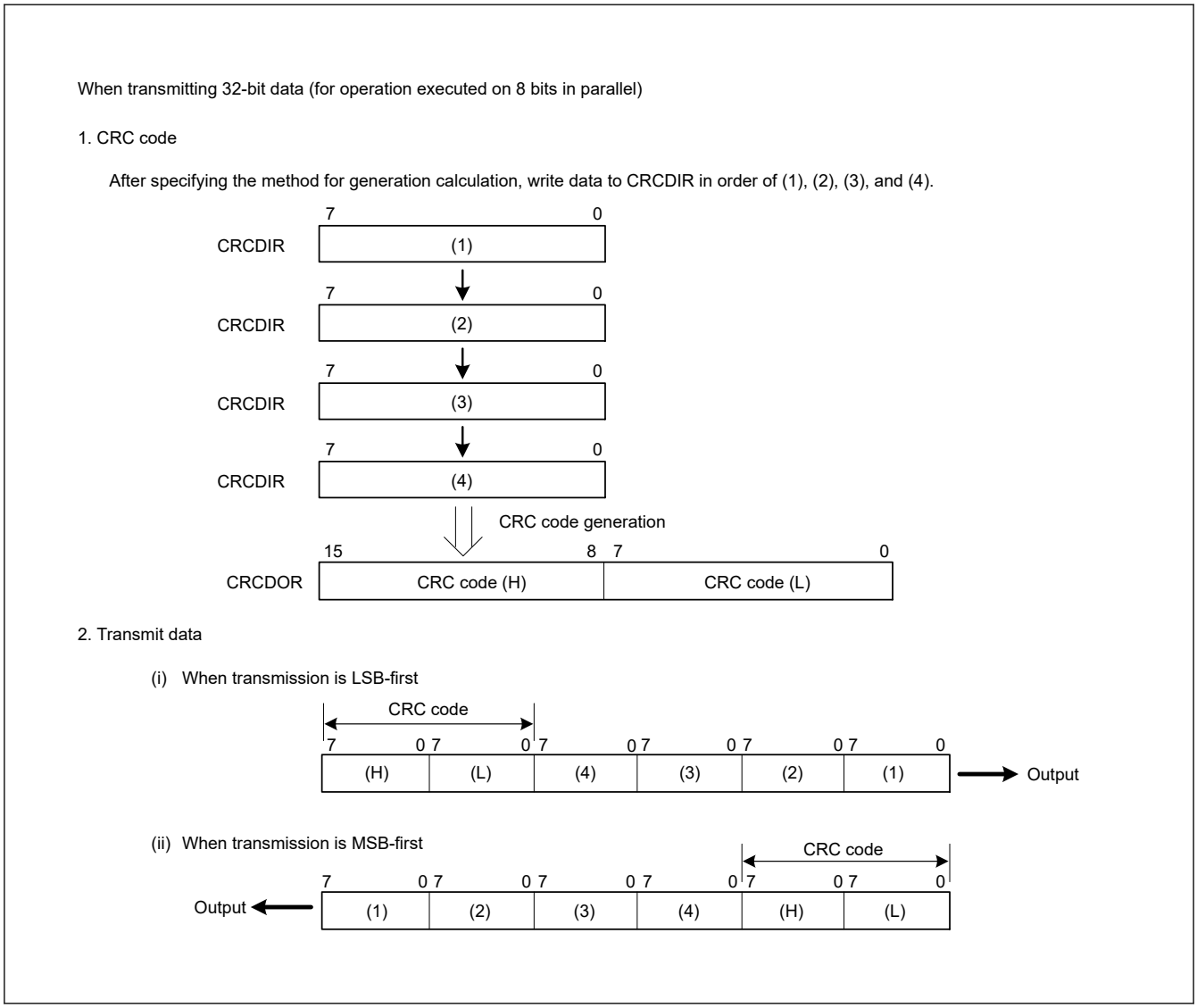


Figure 27.6 LSB-first and MSB-first data transmission

28. High-Speed Analog Comparator (ACMPHS)

28.1 Overview

The High-Speed Analog Comparator (ACMPHS) compares a reference input voltage with an analog input voltage. Comparator channels ACMPHS0 and ACMPHS1 are independent of each other.

The comparison result of the reference input voltage and analog input voltage can be read by software. The comparison result can also be output externally. The reference input voltage can be selected from an input to the CMPREFi (i = 0, 1) pin, input to the CMPIN0j (j = 0, 1, 2) pin or the internal reference voltage (Vref) generated internally in the MCU.

Table 28.1 lists the specifications of the ACMPHS. Figure 28.1 shows a block diagram of the ACMPHS. Table 28.2 lists the I/O pins of the ACMPHS.

Table 28.1 ACMPHS specifications

Parameter	Specifications
Number of channels	2 channels: ACMPHS0 and ACMPHS1
Analog input voltage	Comparator 0: ● Input from CMPIN0i (i = 1, 2, 3) pin ● External reference voltage CMPREF0 Comparator 1: CMPIN1
Reference voltage	Comparator 0: ● Input from CMPIN0i (i = 1, 2, 3) pin ● External reference voltage CMPREF0 ● Internal reference voltage (V_{ref}) Comparator 1: ● External reference voltage CMPREF1 ● Internal reference voltage (V_{ref})
Comparator output	● Comparison result ● Generation of ELC event output ● Monitor output from register
Interrupt request signal	● Interrupt request generated on valid edge detection from comparison result ● Selectable to rising edge, falling edge, or both edges
Digital filter function	● Selectable to one of two sampling frequencies ● Not using the filter function is selectable

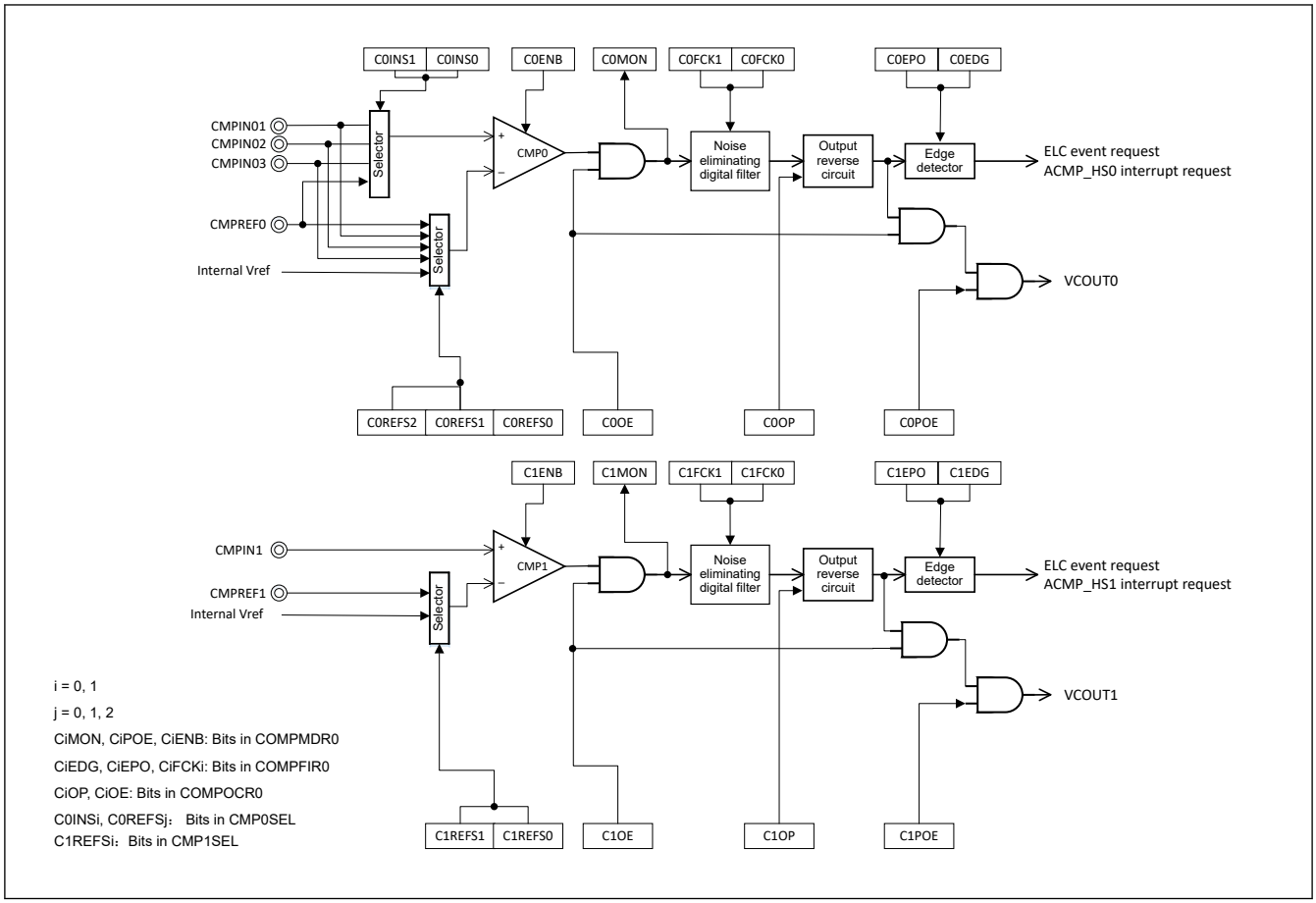


Figure 28.1 ACMPHS block diagram

Table 28.2 Input source configuration of the ACMPHS

Comparator	Reference voltage input pin	Analog voltage input pin	Output pin
ACMPHS0	CMPIN01	CMPIN01	VCOUT*1
	CMPIN02	CMPIN02	
	CMPIN03	CMPIN03	
	CMPREF0	CMPREF0	
ACMPHS1	CMPREF1	CMPIN1	

Note 1. ACMPHS0 and ACMPHS1 compare outputs are bundled with the VCOUT pin.

28.2 Register Descriptions

28.2.1 COMPMDR0 : ACMPHS Mode Setting Register 0

Base address: ACMPHS = 0x4008_5E00

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	C1MON	—	C1POE	C1ENB	C0MON	—	C0POE	C0ENB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	C0ENB	Enabling operation of comparator 0 0: The operation of comparator 0 is disabled. 1: The operation of comparator 0 is enabled.	R/W
1	C0POE	ACMPHS0 VCOUT Pin Output Enable*1 0: Output of the result from comparator 0 on the VCOUT0 is disabled. 1: Output of the result from comparator 0 on the VCOUT0 is enabled.	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	C0MON	Monitoring flag of comparator 0*2 0: The input voltage of comparator 0 is lower than the reference voltage of comparator 0, or comparator 0 is stopped. 1: The input voltage of comparator 0 is greater than the reference voltage of comparator 0.	R
4	C1ENB	Enabling operation of comparator 1 0: The operation of comparator 1 is disabled. 1: The operation of comparator 1 is enabled.	R/W
5	C1POE	ACMPHS1 VCOUT Pin Output Enable*1 0: Output of the result from comparator 1 on the VCOUT1 is disabled. 1: Output of the result from comparator 1 on the VCOUT1 is enabled.	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	C1MON	Monitoring flag of comparator 1*2 0: The input voltage of comparator 1 is lower than the reference voltage of comparator 1, or comparator 1 is stopped. 1: The input voltage of comparator 1 is greater than the reference voltage of comparator 1.	R

Note 1. ACMPHS0 and ACMPHS1 result outputs are bundled on the VCOUT pin.
Note 2. The initial value is 0 immediately after release from a reset. However, the value is undefined when both the C0ENB and C1ENB bits are set to 0 after operation of the comparator has been enabled.

The COMPMDR0 register is used to enable or disable the operation of the comparators and monitor the output of the comparators. Setting the CiENB bit to 0 is prohibited when COMPOCR0.CiOE = 1 (i = 0, 1).

The COMPMDR0 register can be set by an 8-bit memory manipulation instruction.

28.2.2 COMPFIR0 : ACMPHS Filter Control Register 0

Base address: ACMPHS = 0x4008_5E00

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	C1EDG	C1EPO	C1FCK[1:0]		C0EDG	C0EPO	C0FCK[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	C0FCK[1:0]	Selecting the filter for comparator 0*1 0 0: The filter for comparator 0 is disabled. 0 1: The filter for comparator 0 is enabled, and sampling proceeds at a cycle of PCLKB. 1 0: The filter for comparator 0 is enabled, and sampling proceeds at a cycle of PCLKB/8. 1 1: Setting prohibited	R/W
2	C0EPO	Switching the edge sense of comparator 0*1 0: Interrupt and ELC event request on rising edge 1: Interrupt and ELC event request on falling edge	R/W
3	C0EDG	Selecting the edge detection for comparator 0*1 0: Interrupt and ELC event request by one-edge detection 1: Interrupt and ELC event request by both-edge detection	R/W

Bit	Symbol	Function	R/W
5:4	C1FCK[1:0]	Selecting the filter for comparator 1[*1] 0 0: The filter for comparator 1 is disabled. 0 1: The filter for comparator 1 is enabled, and sampling proceeds at a cycle of PCLKB. 1 0: The filter for comparator 1 is enabled, and sampling proceeds at a cycle of PCLKB/8. 1 1: Setting prohibited	R/W
6	C1EPO	Switching the edge sense of comparator 1[*1] 0: Interrupt and ELC event request on rising edge 1: Interrupt and ELC event request on falling edge	R/W
7	C1EDG	Selecting the edge detection for comparator 1[*1] 0: Interrupt and ELC event request by one-edge detection 1: Interrupt and ELC event request by both-edge detection	R/W

Note 1. If bits CiFCK[1:0], CiEPO, and CiEDG (i = 0, 1) are changed, an ACMPLP interrupt request and an ELC event request can be generated. Change these bits only after setting event link to deselect. Also, be sure to clear the associated interrupt request flag.

The COMPFIR0 register is used to control the digital noise filter. The COMPFIR0 register can be set by an 8-bit memory manipulation instruction.

28.2.3 COMPOCR0 : ACMPHS Output Control Register 0

Base address: ACMPHS = 0x4008_5E00

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	C1OP	C1OE	—	—	C0OP	C0OE	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	C0OE	ACMPHS0 VCOUT Pin Output Enable[*1] 0: Output of the result from comparator 0 on the VCOUT0 is disabled. 1: Output of the result from comparator 0 on the VCOUT0 is enabled.	R/W
2	C0OP	ACMPHS0 VCOUT Output Polarity Selection[*1] 0: The output of comparator 0 is output to VCOUT0. 1: The inverted output of comparator 0 is output to VCOUT0.	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	C1OE	ACMPHS1 VCOUT Pin Output Enable[*1] 0: Output of the result from comparator 1 on the VCOUT1 is disabled. 1: Output of the result from comparator 1 on the VCOUT1 is enabled.	R/W
6	C1OP	ACMPHS1 VCOUT Output Polarity Selection[*1] 0: The output of comparator 1 is output to VCOUT1. 1: The inverted output of comparator 1 is output to VCOUT1.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. ACMPHS0 and ACMPHS1 result outputs are bundled on the VCOUT pin.

The COMPOCR0 register is used to control the logical sense of the comparators output, enable or disable the output, and enable or disable the interrupt output. Setting the CiOE bit to 1 (operation enabled) is prohibited in the following case (i = 0, 1).

- Operation of comparator i is disabled (COMPMDR0.CiENB = 0)
  The COMPOCR0 register can be set by an 8-bit memory manipulation instruction.

28.2.4 CMP0SEL : Comparator 0 Input Signal Selection Control Register

Base address: ACMPHS = 0x4008_5E00

Offset address: 0x0A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	C0INS[1:0]		—	—	—	C0REFS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	C0REFS[2:0]	Selecting the input signal on the negative side of comparator 0 0 0 0: The external reference voltage (CMPREF0) is selected. 0 0 1: External pin (CMPIN01) is selected. 0 1 0: External pin (CMPIN02) is selected. 0 1 1: External pin (CMPIN03) is selected. 1 0 0: Select internal reference voltage (V_{ref}) as ACMPHS0 reference voltage. Others: Setting prohibited	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
7:6	C0INS[1:0]	Selecting the input signal on the positive side of comparator 0 0 0: External pin (CMPIN01) is selected. 0 1: External pin (CMPIN02) is selected. 1 0: External pin (CMPIN03) is selected. 1 1: The external reference voltage (CMPREF0) is selected.	R/W

Note: Set the switching interval to at least 3 µs for stable operation when switching the analog input of comparator 0.

The CMP0SEL register is used to select the input signal on the negative side of comparator 0. The CMP0SEL register should only be rewritten while comparator operation is stopped (C0ENB = 0). The CMP0SEL register can be set by an 8-bit memory manipulation instruction.

28.2.5 CMP1SEL : Comparator 1 Input Signal Selection Control Register

Base address: ACMPHS = 0x4008_5E00

Offset address: 0x0B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	C1REFS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	C1REFS[1:0]	Selecting the input signal on the negative side of comparator 1 0 0: The external reference voltage (CMPREF1) is selected. 0 1: Select internal reference voltage (V_{ref}) as ACMPHS1 reference voltage. Others: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the switching interval to at least 3 µs for stable operation when switching the analog input of comparator 1.

The CMP1SEL register is used to select the input signal on the negative side of comparator 1. The CMP1SEL register should only be rewritten while comparator operation is stopped (C1ENB = 0). The CMP1SEL register can be set by an 8-bit memory manipulation instruction.

28.3 Operation

ACMPHS0 and ACMPHS1 are capable of independent operation. The setting methods and operations are the same for each of them. The procedure for setting the individual or interlocked operation of the comparators is described in Table 28.3.

Table 28.3 Procedure for setting the ACMPHS to Control the Comparators (i = 0, 1; j = 1, 2, 3)

Step number	Register	Bit	Setting
1	MSTPCRD	MSTPD28	0: Cancel the module-stop state.
2	Corresponding Port mn Pin Function Select Register PmnPFS	ASEL	1: Select the function of pins CMPREFi, CMPIN0j, CMPIN1.
3	CMP0SEL	C0INS	Select the positive-side input of comparator 0.
4	CMP0SEL	C0REFS[2:0]	Select the negative-side input of comparator 0.
5	CMP1SEL	C1REFS[1:0]	Select the negative-side input of comparator 1.
6	COMPMDR0	CiENB	1 (operation is enabled)
7	Wait for the comparator stabilization time (at least 3 µs).		
8	COMPFIR0	CiFCK[1:0]	Select whether the digital filter is to be used or not and the sampling clock.
		CiEPO, CiEDG	Select the edge detection condition for an interrupt request (rising edge, falling edge, or both edges).
9	COMPOCR0	CiOP	Set the output of comparator i (selecting the logical sense).
10	COMPOCR0	CiOE	Set the output of comparator i (enabling output).
11	COMPMDR0	CiPOE	Set the output of the VCOUTi pins (enabling output).*1
12	Corresponding Port mn Pin Function Select Register PmnPFS	PSEL, PMR	Select the VCOUT port function.
13	IELSRn	IR, IELS[4:0]	When using an interrupt: select the interrupt status flag, ICU event link select.*2
14	ELSRn	ELS[7:0]	When using an ELC: Select the Event Link Select.*3
15	Operation started		

Note 1. ACMPHS0 and ACMPHS1 result outputs are bundled on the VCOUT pin.
Note 2. After the setting of the comparator, a spurious interrupt may occur until operation becomes stable, so initialize the interrupt flag.
Note 3. After the setting of the comparator, a spurious interrupt may occur until operation becomes stable, so initialize the event link select.

Figure 28.2 shows an example of operation for comparator i (i = 0, 1; in normal output mode). The CiMON flag in the COMPMDR0 register is set to 1 when the voltage of the analog input is higher than the reference voltage, and the CiMON flag is set to 0 when the voltage of the analog input is lower than the reference voltage.

ACMPHSi outputs an interrupt to ICU. For details on the interrupt, see section 28.5. ACMPHS Interrupts.

ACMPHSi outputs an event signal to the ELC to activate other modules. For details on the ELC, see section 28.6. ELC Event Output.

Do not change the values of the registers during comparison.

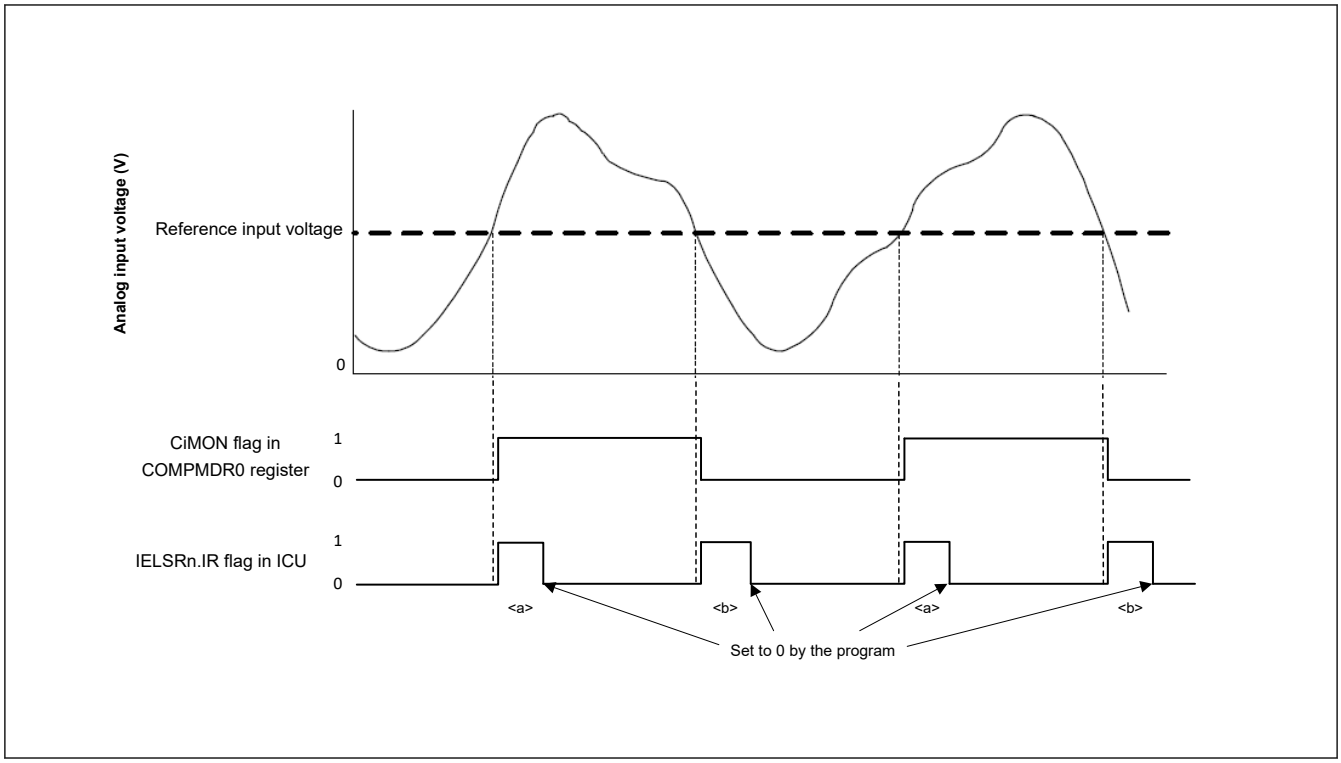


Figure 28.2 ACMPHS operation example

- CiFCK[1:0] = 00b (no sampling) and CiEDG = 1 (both edges)
- When CiEDG = 0 and CiEPO = 0 (rising edge), IELSRn.IR changes as shown by <a> only.
- When CiEDG = 0 and CiEPO = 1 (falling edge), IELSRn.IR changes as shown by <b> only.

28.4 Noise Filter

The ACMPHS contains a noise filter. The sampling clock can be selected in the COMPFIR.CiFCK[1:0] bits. The ACMPHSi signal (internal signal) output from ACMPHSi is sampled at every sampling clock cycle. When the level matches three times, the corresponding IELSRn.IR bit is set to 1 (interrupt requested) and an ELC event is output.

When using an interrupt in Software Standby mode, set the COMPFIR.CiFCK[1:0] bits to 00b (bypass).

Figure 28.3 shows the configuration of the digital filter for comparator i and edge detection. Figure 28.4 shows the digital filter for comparator i (i = 0, 1) and an example of interrupt operation.

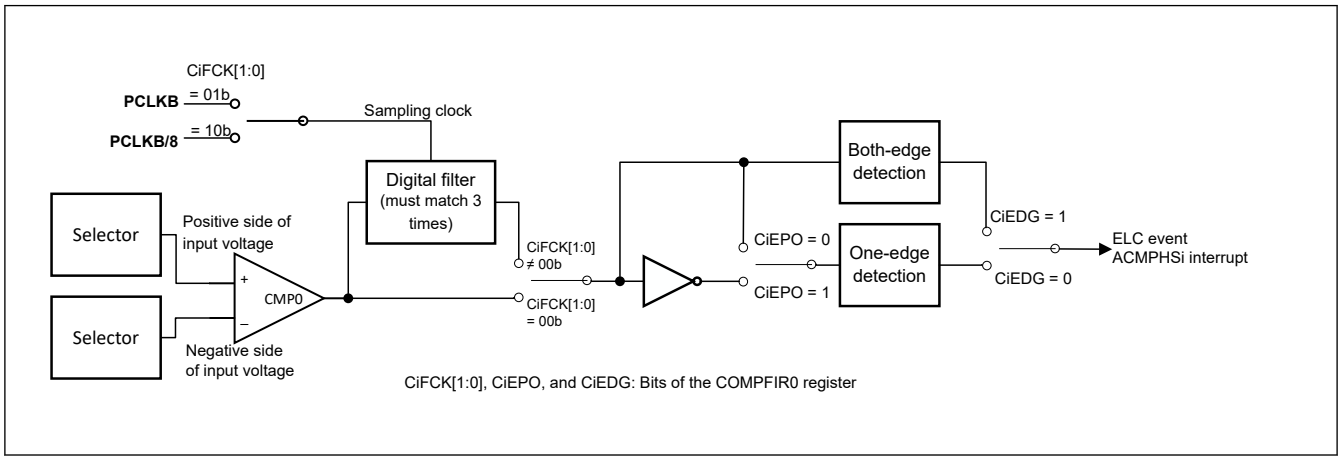


Figure 28.3 Noise filter and edge detection configuration

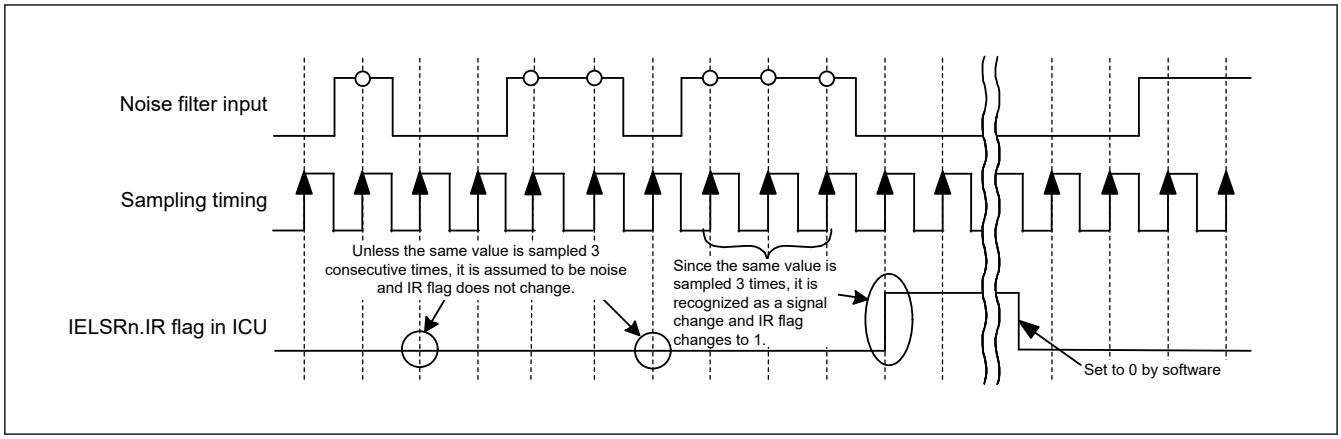


Figure 28.4 Noise filter and interrupt operation example

28.5 ACMPHS Interrupts

The ACMPHS generates interrupt requests from the ACMPHS0 and ACMPHS1 sources. To use the ACMPHSi (i = 0, 1) interrupt, select it in the IELSRn register in the ICU.

To use the ACMPHSi interrupt, select either single-edge detection or both-edge detection using the COMPFIR0.CiEDG bit. When single-edge detection is selected, select the polarity using the CiEPO bit.

The interrupt output can also be passed through the noise filter, which uses one of the two different sampling clocks, as selected in the COMPFIR0.CiFCK[1:0] bits. Set the COMPFIR0.CiFCK[1:0] bits to 01b or 10b to select the respective sampling clock. To use the ACMPHS0 interrupt request to release Software Standby mode or Snooze mode, set COMPFIR0.C0FCK[1:0] to 00b (no sampling). The ACMPHS1 interrupt request cannot be used to release Software Standby mode or Snooze mode.

28.6 ELC Event Output

The ELC uses the ACMPHS interrupt request signal as an ELC event signal, enabling link operation for the preset module. To use the ELC events of the ACMPHS, select them in the ELSRn register in the ELC. When using ELC event request, set the COMPFIR0.CiFCK[1:0] bits to 01b, 10b.

28.7 Interrupt Handling and ELC Linking

ACMPHSi outputs event signals to the ELC to initiate operations of other modules selected in advance. In the same way as for the interrupt sources, the conditions for generation of the event signals output from ACMPHSi to the ELC can be selected as a single-edge detection or both-edge detection by setting the COMPFIR0.CiEDG bit. When the single-edge detection is selected, the polarity can be selected in the CiEPO bit.

28.8 Comparator Pin Output

The result of comparison by a comparator can be output to an external pin. The CiOP and CiOE bits in the COMPOCR0 register can be used to set the logical sense of the output (non-inverted or inverted) and the CiPOE bit in the COMPMDR0 register can be used to enable or disable the output.

To output the ACMPHS comparison result to the VCOUT output pin, set section 17.2.5. PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 5, 9, n = 00 to 15).

For the correspondence between register settings and comparator output, see the following descriptions of the registers.

- section 28.2.1. COMPMDR0 : ACMPHS Mode Setting Register 0
- section 28.2.3. COMPOCR0 : ACMPHS Output Control Register 0

28.9 Usage Notes

28.9.1 Module-Stop Function Settings

The Module Stop Control Register can enable or disable the ACMPHS operation. The ACMPHS is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

29. 12-Bit A/D Converter (ADC12)

29.1 Overview

The MCU includes 12-bit successive approximation A/D converter (ADC12) unit. Up to 13 analog input channels, temperature sensor output, and internal reference voltage can be selected for conversion.

The ADC12 supports the following operating modes:

- Single scan mode to convert analog inputs of selected channels in ascending order of channel number
- Continuous scan mode to convert analog inputs of selected channels continuously in ascending order of channel number
- Group scan mode to divide analog inputs of channels into two groups (group A and B) and convert the analog inputs of selected channels for each group in ascending order of channel number.
- Built-in channel-dedicated sample-and-hold circuit (S/H)

In group scan mode, select two groups (group A and B). You can individually select the scan start conditions for each group (group A, B) and start scanning of each group at different times. In addition, when group A priority control operation is set, the ADC12 accepts group A scan start during group B A/D conversion, suspending group B conversion. This allows you to assign higher priority to A/D conversion start for group A.

In double trigger mode, the analog input of a selected channel is converted in single scan mode or group scan mode (group A), and data converted by the first and second A/D conversion start triggers are stored in different registers, providing duplexing of A/D converted data.

Self-diagnosis is performed once at the beginning of each scan, and one of the three reference voltage values generated in ADC12 is A/D converted.

The temperature sensor output and the internal reference voltage cannot be selected for conversion simultaneously. Perform A/D conversion independently for the temperature sensor output or the internal reference voltage.

The ADC12 also provides a compare function (window A and window B). The compare function specifies the upper reference value for window A and lower reference value for window B, and outputs an interrupt when the A/D converted value of the selected channel meets the comparison conditions.

The reference power supply pin (VREFH0), the analog block power supply pin (AVCC0), or the internal reference voltage can selected as the high-potential reference voltage. The reference power supply ground pin (VREFL0) or the analog block power supply ground pin (AVSS0) can be selected as the low-potential reference voltage. If the internal reference voltage is selected as the high-potential reference voltage, A/D conversion of the temperature sensor or internal reference voltage is prohibited.

Table 29.1 lists the ADC12 specifications and Table 29.2 lists the functions. Figure 29.1 shows a block diagram of ADC12 and Table 29.3 lists the I/O pins.

Table 29.1 ADC12 specifications (1 of 3)

Parameter	Specifications
Number of units	One unit
Input channels	Up to 13 channels (AN000 to AN002, AN005 to AN010, AN019 to AN022)*1
Extended analog function	Temperature sensor output and internal reference voltage
A/D conversion method	Successive approximation method
Resolution	12-bit
Conversion time	Normal conversion mode (ADACSR.ADSAC = 0): 0.7 µs/channel (when 12-bit A/D conversion clock PCLKD (ADCLK) operates at 64 MHz) Fast conversion mode (ADACSR.ADSAC = 1): 0.67 µs/channel (when 12-bit A/D conversion clock PCLKD (ADCLK) operates at 48 MHz)
A/D conversion clock	Peripheral module clock PCLKB and A/D conversion clock PCLKD (ADCLK) can be set with the following division ratios: PCLKB to PCLKD (ADCLK) frequency ratio = 1:1, 1:2, 1:4

Table 29.1 ADC12 specifications (2 of 3)

Parameter	Specifications
Data registers*2	● 13 registers for analog input ● One register for A/D-converted data duplication in double trigger mode ● Two registers for A/D-converted data duplication during extended operation in double trigger mode ● One register for temperature sensor output ● One register for internal reference voltage ● One register for self-diagnosis ● A/D conversion results are stored in A/D data registers ● 12-bit accuracy for A/D conversion results ● A/D-converted value addition mode, in which the sum of all A/D-converted results is stored in the A/D data registers as a value with the conversion accuracy bit count + extended bits ● Double-trigger mode (selectable in single scan and group scan modes): – The first unit of A/D-converted analog input data on one selected channel is stored in the data register for the channel, and the second unit is stored in the duplication register. ● Extended operation in double trigger mode (available for specific triggers): – A/D-converted analog input data on one selected channel is stored in the duplication register provided for the associated trigger.
Operating modes*3	● Single scan mode: – A/D conversion is performed only once on the analog inputs of arbitrarily selected channels, on the temperature sensor output, and on the internal reference voltage. ● Continuous scan mode: A/D conversion is performed repeatedly on the analog inputs of the selected channels. ● Group scan mode: – Analog inputs of selected channels, the temperature sensor output, and the internal reference voltage are divided into groups A and B. Then A/D conversion of the analog inputs selected on a group basis is performed once. – The scan start conditions can be independently selected for groups A and B, allowing A/D conversion of groups A and B to be started independently. ● Group scan mode (when group priority operation is selected): – If a priority group trigger is input during scanning of a lower-priority group, the scanning of the lower-priority group is stopped and then scanning of the priority group is started. The order of priority is group A > group B.
Conditions for A/D conversion start	● Software trigger ● Synchronous triggers from the Event Link Controller (ELC) ● Asynchronous triggering by the external trigger pins, ADTRG0 ● Triggers from GPT
Functions	● Variable sampling state count ● Channel-dedicated sample-and-hold circuit (S/H) (3 S/H units for A/D Converter) ● Self-diagnosis of A/D converter ● Selectable A/D-converted value addition mode or average mode ● Analog input disconnection detection function (discharge and precharge functions) ● Double-trigger mode (duplication of A/D conversion data) ● Automatic clear function for A/D data registers ● Digital comparison of values in the comparison register and data register, and comparison between values in the data registers
Interrupt sources	● In single scan mode (double trigger deselected), an A/D scan end interrupt request (ADC120_ADI) and ELC event signal (ADC120_ADI) can be generated on completion of single scan. – A compare interrupt request (ADC120_CMPAI/ADC120_CMPBI) can be generated in response to a match with a digital comparison condition. – A window compare ELC event signal (ADC120_WCMPM) can be generated in response to a match with a digital comparison condition. – A window compare ELC event signal (ADC120_WCMPUM) can be generated in response to a mismatch with a digital comparison condition. ● In single scan mode (double trigger selected), an A/D scan end interrupt request (ADC120_ADI) and ELC event signal (ADC120_ADI) is generated on completion of two scans. ● In continuous scan mode, an A/D scan end interrupt request (ADC120_ADI) and ELC event signal (ADC120_ADI) is generated on completion of all the selected channel scans. ● In group scan mode (double trigger deselected), an A/D scan end interrupt request (ADC120_ADI) and ELC event signal (ADC120_ADI) is generated on completion of group A scan, and an A/D scan end interrupt request for group B (ADC120_GBADI) can be generated on completion of group B scan. ● In group scan mode (double trigger selected), an A/D scan end interrupt request (ADC120_ADI) and ELC event signal (ADC120_ADI) is generated on completion of two group A scans, and an A/D scan end interrupt request for group B (ADC120_GBADI) can be generated on completion of group B scan. ● ADC120_ADI, ADC120_GBADI, ADC120_WCMPM, and ADC120_WCMPUM can activate the Data Transfer Controller (DTC).

Table 29.1 ADC12 specifications (3 of 3)

Parameter	Specifications
ELC interface	• An event is generated upon completion of group A scan in group-scan mode. • An event is generated upon completion of group B scan in group-scan mode. • An event is generated when all scans complete. • Scan can be started by a trigger from the ELC. • An event is generated according to conditions of the compare function window in single-scan mode.
Reference voltage	• VREFH0, AVCC0, or internal reference voltage (BGR) (external reference voltage or output voltage from reference voltage generation circuit) can be selected as the analog reference voltage. • VREFL0 or AVSS0 can be selected as the analog reference ground.
Module-stop function	Module-stop state can be set to reduce power consumption.*4

Note 1. AN000 to AN002, AN005 to AN010, AN019 to AN022 for LQFP/HWQFN 48-pin
AN000 to AN002, AN005 to AN007, AN019 to AN022 for LQFP/HWQFN 32-pin
AN000 to AN002, AN005, AN006, AN019 to AN022 for HWQFN 24-pin
Note 2. Changing the A/D conversion accuracy also changes the A/D conversion time. For details, see section 29.3.6. Analog Input Sampling and Scan Conversion Time.
Note 3. When selecting the temperature sensor output and the internal reference voltage, do not use continuous scan mode or group scan mode.
Note 4. For details, see section 10, Low Power Modes.

Table 29.2 ADC12 functions

Parameter			Function
Analog input channel			AN000 to AN002, AN005 to AN010, AN019 to AN022 Internal reference voltage Temperature sensor output
Conditions for A/D conversion start	Software	Software trigger	Enabled
	Asynchronous trigger (external trigger)	Trigger input pin	ADTRG0
	Synchronous trigger (trigger from ELC)	ELC trigger	ELC_AD00, ELC_AD01
	Synchronous trigger (trigger from GPT)	GPT trigger	GPTn_ADTRGA, GPTn_ADTRGB (n = 0 to 3)
Channel-dedicated sample-and-hold function	Target channel		AN000 to AN002
Interrupt			ADC120_ADI ADC120_GBADI ADC120_CMPAI ADC120_CMPBI
Output to ELC			ADC120_ADI ADC120_WCMPM ADC120_WCMPUM
Module-stop function settings*1 *2			MSTPCRD.MSTPD16 bit

Note 1. For details, see section 10, Low Power Modes.
Note 2. Wait 1 µs or longer to start A/D conversion after release from the module-stop state.

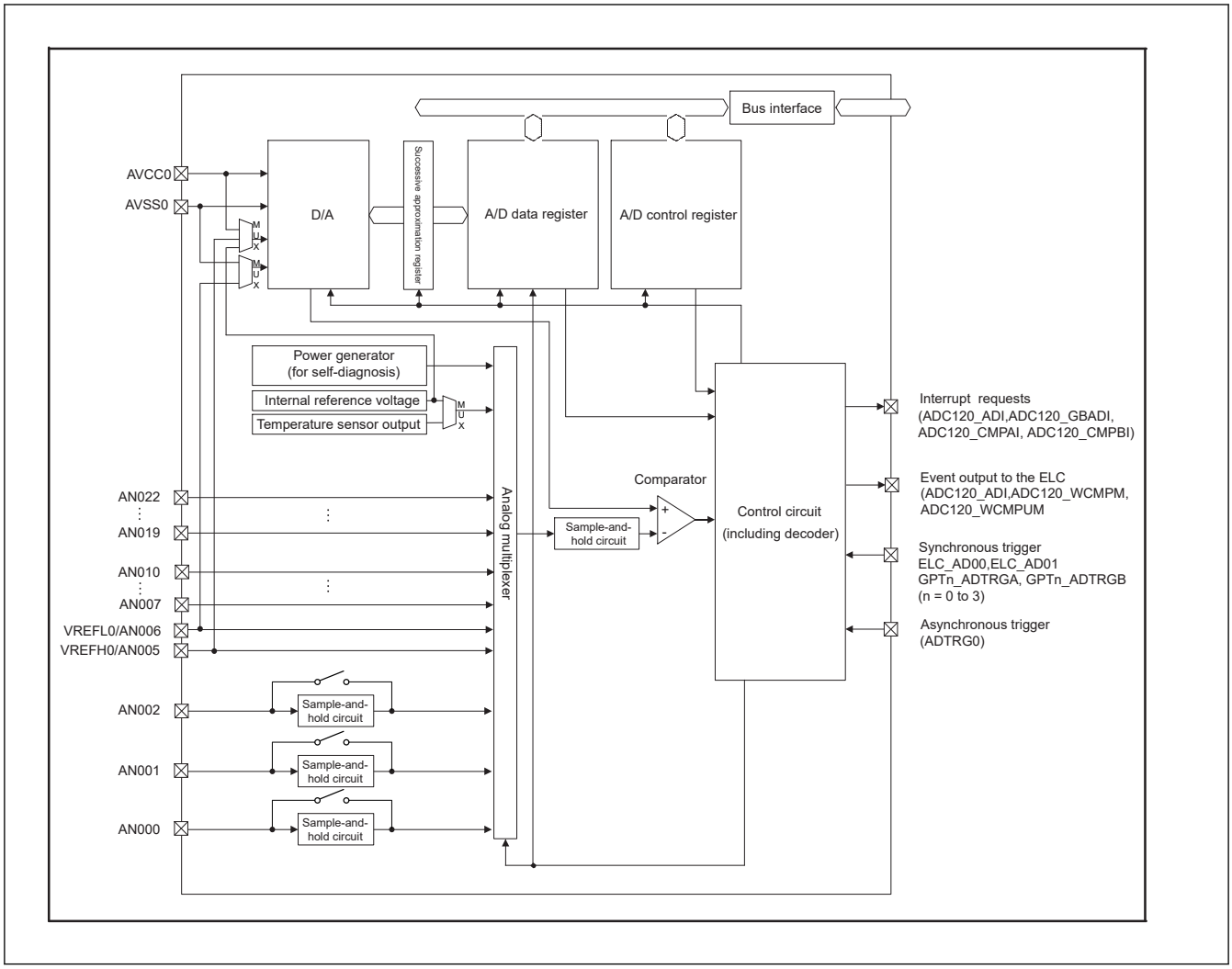


Figure 29.1 ADC12 block diagram

Table 29.3 lists the ADC12 I/O pins.

Table 29.3 ADC12 I/O pins

Pin name	I/O	Function
AVCC0	Input	Analog block power supply pin (Connect to VCC when ADC12 is not used.)
AVSS0	Input	Analog block power supply ground pin (Connect to VSS when ADC12 is not used.)
VREFH0	Input	Analog reference voltage supply pin
VREFL0	Input	Analog reference ground pin
AN000 to AN002, AN005 to AN010, AN019 to AN022	Input	Analog input pins 0 to 2, 5 to 10, 19 to 22
ADTRG0	Input	External trigger input pin for starting A/D conversion

29.2 Register Descriptions

29.2.1 ADDRn : A/D Data Registers n (n = 0 to 2, 5 to 10, 19 to 22)

Base address: ADC120 = 0x4005_C000

Offset address: 0x020 + 0x2 × n (n = 0 to 2, 5 to 10, 19 to 22)

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

Bit	Symbol	Function	R/W
15:0	ADDR [15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 29.4 and Table 29.5.	R

ADDRn registers are 16-bit read-only registers to store A/D conversion results.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

Table 29.4 shows the example of bit assignment for 12-bit accuracy.

Table 29.4 Example of bit assignment for 12-bit accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, these registers indicate the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, these registers indicate the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 29.5 shows example of the bit assignment for 12-bit accuracy.

Table 29.5 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.		Added Value 13 to 0: 14-bit sum of A/D conversion results													
Left-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

29.2.2 ADDBLDR : A/D Data Duplexing Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x018

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

Bit	Symbol	Function	R/W
15:0	ADDBLDR [15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 29.6 and Table 29.7.	R

ADDBLDR register is a 16-bit read-only register to store A/D conversion results in response to a second trigger in double-trigger mode.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

Table 29.6 shows the example of bit assignment for 12-bit accuracy.

Table 29.6 Example of bit assignment for 12-bit accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, this register indicates the mean of A/D-converted values on a

specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, this register indicates the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 29.7 shows example of the bit assignment for 12-bit accuracy.

Table 29.7 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.		Added Value 13 to 0: 14-bit sum of A/D conversion results													
Left-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

29.2.3 ADSHCR : A/D S&H Circuit Control Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x066

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

Bit	Symbol	Function	R/W
7:0	SSTSH[7:0]	Channel-Dedicated S&H Circuit Sampling Time Setting Set the sampling time (4 to 255 states).	R/W*1
10:8	SHANS[2:0]	Channel-Dedicated S&H Circuit Bypass Select Select whether to use or not use (bypass) AN000 to AN002 channel-dedicated S&H circuits. 0: Bypass the channel-dedicated S&H circuits. 1: Use the channel-dedicated S&H circuits.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 0x03 or less to the SSTSH[7:0] is changed to 0x04.

ADSHCR sets the parameters related to channel-dedicated S&H (sample-and-hold) circuits.

SSTSH[7:0] bits (Channel-Dedicated S&H Circuit Sampling Time Setting)

The SSTSH[7:0] bits set the sampling time for the channel-dedicated sample-and-hold circuits. If one state is one ADCLK (A/D conversion clock) cycle and the ADCLK clock is 64 MHz, one state is 15.6 ns. The initial value is 26 states. If the impedance of analog input signal source is too high to secure sufficient sampling time or if the ADCLK clock is slow, the sampling time can be adjusted.

The SSTSH[7:0] bits should be set while the ADCSR.ADST bit is 0. The sampling time must be set to a value that is 4 states or more and is 255 or less. Also, the sampling time needs to be 4us or more.

SHANS[2:0] bits (Channel-Dedicated S&H Circuit Bypass Select)

The SHANS[2:0] bits select whether to use or not use (bypass) AN000 to AN002 channel-dedicated sample-and-hold circuits. The SHANS[0] bit selects AN000, the SHANS[1] bit selects AN001, and the SHANS[2] bit selects AN002. The SHANS[2:0] bits should be set while the ADST bit in ADCSR is 0.

In the group priority operation, make the setting of bypassing the channel-dedicated S&H circuit when one of AN000 to AN002 is selected for group B.

In the group priority operation, group B performs scan suspension and restart. The sample-and-hold circuit cannot be held according to the timer because it performs sampling and hold in the restart process.

29.2.4 ADDBLDRn : A/D Data Duplexing Register n (n = A, B)

Base address: ADC120 = 0x4005_C000

Offset address: 0x084 (n = A)
0x086 (n = B)

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

Bit	Symbol	Function	R/W
15:0	ADDBLDR [15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 29.8 and Table 29.9.	R

ADDBLDRn registers are 16-bit read-only registers to store A/D conversion results in response to respective triggers during extended operation in double-trigger mode.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

Table 29.8 shows the example of bit assignment for 12-bit accuracy.

Table 29.8 Example of bit assignment for 12-bit accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, these registers indicate the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, these registers indicate the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 29.9 shows example of the bit assignment for 12-bit accuracy.

Table 29.9 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.		Added Value 13 to 0: 14-bit sum of A/D conversion results													
Left-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

29.2.5 ADTSDR : A/D Temperature Sensor Data Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x01A

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

Bit	Symbol	Function	R/W
15:0	ADTSDR [15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 29.10 and Table 29.11.	R

ADTSDR register is a 16-bit read-only register to store A/D conversion result of the temperature sensor output.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

Table 29.10 shows the example of bit assignment for 12-bit accuracy.

Table 29.10 Example of bit assignment for 12-bit accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, this register indicates the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, this register indicates the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 29.11 shows example of the bit assignment for 12-bit accuracy.

Table 29.11 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.	Added Value 13 to 0: 14-bit sum of A/D conversion results														
Left-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results															These bits are read as 0.

29.2.6 ADOCDR : A/D Internal Reference Voltage Data Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x01C

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

Bit	Symbol	Function	R/W
15:0	ADOCDR [15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 29.12 and Table 29.13.	R

ADOCDR register is a 16-bit read-only register to store A/D conversion result of the internal reference voltage.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

Table 29.12 shows the example of bit assignment for 12-bit accuracy.

Table 29.12 Example of bit assignment for 12-bit accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, this register indicates the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, this register indicates the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 29.13 shows example of the bit assignment for 12-bit accuracy.

Table 29.13 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.		Added Value 13 to 0: 14-bit sum of A/D conversion results													
Left-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

29.2.7 ADRD : A/D Self-Diagnosis Data Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x01E

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

Bit	Symbol	Function	R/W
11:0	AD[11:0]	Converted Value 11 to 0 12-bit A/D-converted value	R
13:12	—	These bits are read as 0.	R
15:14	DIAGST[1:0]	Self-Diagnosis Status For details on self-diagnosis, see section 29.2.16. ADCER : A/D Control Extended Register. 0 0: Self-diagnosis not executed after power-on. 0 1: Self-diagnosis was executed using the 0 V voltage. 1 0: Self-diagnosis was executed using the reference voltage*1 × 1/2. 1 1: Self-diagnosis was executed using the reference voltage *1.	R

Note: The example of the bit assignment for the right-justified data with 12-bit accuracy is indicated.
Note 1. The reference voltage refers to VREFH0, AVCC0, or internal reference voltage (BGR) by ADHVREFCNT setting.

ADRD is a 16-bit read-only register that holds the A/D conversion results based on the self-diagnosis of the ADC12. In addition to the AD[11:0] bits indicating the A/D-converted value, it includes the Self-Diagnosis Status bit (DIAGST[1:0]).

The settings of the A/D data register format and the A/D conversion accuracy determines the formats for data in this register.

The A/D-converted value addition and average modes cannot be applied to the A/D self-diagnosis function. For details on self-diagnosis, see section 29.2.16. ADCER : A/D Control Extended Register.

This section describes the data formats for each condition. The register diagram and the register bit table shown in this section indicate example of the bit assignment for the left and right-justified data with 12-bit accuracy.

Table 29.14 Bit assignment for each right-justified accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	DIAGST[1:0]		—		AD[11:0]											

Table 29.15 Bit assignment for each left-justified accuracy

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Left-justified data with 12-bit accuracy	AD[11:0]												—		DIAGST[1:0]	

29.2.8 ADCSR : A/D Control Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADST	ADCS[1:0]		—	—	ADHSC	TRGE	EXTRG	DBLE	GBADIE	—	DBLANS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	DBLANS[4:0]	Double Trigger Channel Select These bits select one analog input channel for double-trigger operation. The setting is only valid in double-trigger mode.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	GBADIE	Group B Scan End Interrupt and ELC Event Enable Group B scan only works in group scan mode. 0: Disable ADC120_GBADI interrupt generation on group B scan completion. 1: Enable ADC120_GBADI interrupt generation on group B scan completion.	R/W
7	DBLE	Double Trigger Mode Select 0: Deselect double-trigger mode. 1: Select double-trigger mode.	R/W
8	EXTRG	Trigger Select*1 0: Start A/D conversion by the synchronous trigger (ELC, GPT). 1: Start A/D conversion by the asynchronous trigger (ADTRG0).	R/W
9	TRGE	Trigger Start Enable 0: Disable A/D conversion to be started by the synchronous or asynchronous trigger 1: Enable A/D conversion to be started by the synchronous or asynchronous trigger	R/W
10	ADHSC	A/D Conversion Mode Select 0: High-speed A/D conversion mode 1: Low-power A/D conversion mode	R/W
12:11	—	These bits are read as 0. The write value should be 0.	R/W
14:13	ADCS[1:0]	Scan Mode Select 0 0: Single scan mode 0 1: Group scan mode 1 0: Continuous scan mode 1 1: Setting prohibited	R/W
15	ADST	A/D Conversion Start 0: Stop A/D conversion process. 1: Start A/D conversion process.	R/W

Note 1. To start A/D conversion using an external pin (asynchronous trigger):
After a high-level signal is input to the external pin (ADTRG0), write 1 to both the TRGE and EXTRG bits in the ADCSR register and drive the ADTRG0 pin low. With these settings, the scan conversion process starts on detection of the falling edge of ADTRG0. The pulse width of the low-level input must be at least PCLKB 1.5 clock cycles.

The ADCSR register sets double-trigger mode and A/D conversion start trigger, enables or disables scan end interrupt, selects the scan mode, and starts or stops A/D conversion.

DBLANS[4:0] bits (Double Trigger Channel Select)

The DBLANS[4:0] bits select one channel for A/D conversion data duplication in double-trigger mode. This can be selected by setting the binary value of the channel number to be duplicated. The A/D conversion results of the analog input of the channel selected in the DBLANS[4:0] bits are stored in A/D Data Register y when conversion is started by the first trigger, and in the A/D Data Duplexing Register when conversion is started by the second trigger.

In double-trigger mode, the channels selected in the ADANSA0 and ADANSA1 registers, are invalid, and the channel selected in the DBLANS[4:0] bits is A/D converted instead.

When double-trigger mode is used in group scan mode, double-trigger control is only applied to group A and not to group B. Therefore, multiple channel analog input can be selected for group B even in double-trigger mode.

Only set the DBLANS[4:0] bits when the ADST bit is 0. Do not set the DBLANS[4:0] bits at the same time that you write 1 to the ADST bit.

To enter A/D-converted value addition/average mode when in double-trigger mode, select the channel using the DBLANS[4:0] bits in the ADANSA0 and ADANSA1 registers.

A/D-converted data from the self-diagnosis function temperature sensor output and internal reference voltage cannot be used in double-trigger mode.

Table 29.16 Relationship between DBLANS bit settings and double-trigger enabled channels

DBLANS[4:0]	Duplication channel
0x00	AN000
0x01	AN001
0x02	AN002
0x05	AN005
0x06	AN006
0x07	AN007
0x08	AN008
0x09	AN009
0x0A	AN010
0x13	AN019
0x14	AN020
0x15	AN021
0x16	AN022

GBADIE bit (Group B Scan End Interrupt and ELC Event Enable)

The GBADIE bit enables or disables group B scan end interrupt (ADC120_GBADI) in group scan mode.

DBLE bit (Double Trigger Mode Select)

The DBLE bit selects or deselects double-trigger mode. Double-trigger mode can only be operated by the synchronous trigger (ELC, GPT) selected in the ADSTRGR.TRSA[5:0] bits.

Double-trigger operation is as follows:

1. The ADC120_ADI interrupt is not output on completion of the first conversion but on completion of the second conversion.
2. The A/D conversion results from the duplication channel (selected in DBLANS[4:0]) started by the first trigger are stored in A/D Data Register y and those started by the second trigger are stored in the A/D Data Duplexing Register.

When the DBLE bit is set (double-trigger mode is selected), the channels specified in the ADANSA0 and ADANSA1 registers are invalid. Double-trigger mode is deselected by setting DBLE to 0. Setting DBLE to 1 again enables the same double-trigger operation described in 1. and 2. for first time scanning with the first trigger.

Do not select double-trigger mode in continuous scan mode. Software triggering cannot be used in double-trigger mode. Always set the ADST bit to 0 before setting the DBLE bit. Do not set the DBLE bit at that same time as writing 1 to the ADST bit.

EXTRG bit (Trigger Select)

The EXTRG bit selects the synchronous or asynchronous trigger as the trigger for starting A/D conversion.

In group scan mode, the setting of this bit takes effect on the trigger selected for group A. For group B, A/D conversion is started by the selected synchronous trigger regardless of this bit setting.

TRGE bit (Trigger Start Enable)

The TRGE bit enables or disables A/D conversion by the synchronous and asynchronous triggers. In group scan mode, set this bit to 1.

ADHSC bit (A/D Conversion Mode Select)

The ADHSC bit selects either high speed mode or low current mode for A/D conversion. For details on how to rewrite the ADHSC bit, see section 29.8.9. ADHSC Bit Rewriting Procedure.

ADCS[1:0] bits (Scan Mode Select)

The ADCS[1:0] bits select the scan mode.

In single scan mode, A/D conversion is performed for the analog inputs of the channels selected in the ADANSA0 and ADANSA1 registers, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, the scan conversion stops.

In continuous scan mode, when the ADCSR.ADST bit is 1, A/D conversion is performed for the analog inputs of the channels selected with the ADANSA0 and ADANSA1 registers , in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, A/D conversion repeats from the first channel. If the ADCSR.ADST bit is set to 0 during continuous scan, A/D conversion stops even if scanning is in progress.

In group scan mode:

- Group A scanning is started by the synchronous trigger (ELC, GPT) selected in the TRSA[5:0] bits in the ADSTRGR register. A/D conversion is performed on group A analog inputs of the channels selected in the ADANSA0 and ADANSA1 registers, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, A/D conversion stops.
- Group B scanning is started by the synchronous trigger (ELC, GPT) selected in the ADSTRGR.TRSB[5:0] bits. A/D conversion is performed on group B analog inputs of the channels selected in the ADANSB0 and ADANSB1 registers, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, A/D conversion stops.

If the conversion processes in group A and group B occur at the same time, those conversions cannot be controlled separately. In this case, set group A Priority Control Setting bit (ADGSPCR.PGS) in the A/D Group Scan Priority Control Register (ADGSPCR) to 1 to assign a priority to group A conversion.

In group scan mode, select different channels and triggers for group A and group B.

When selecting temperature sensor output or internal reference voltage, select single scan mode, and perform A/D conversion after deselecting all analog input channels in the ADANSA0 and ADANSA1 registers. When A/D conversion of the temperature sensor output or internal reference voltage completes, A/D conversion stops.

Only set the ADCS[1:0] bits when the ADST bit is 0. Do not set the ADCS[1:0] bits at the same time that you write 1 to the ADST bit.

Table 29.17 Selectable targets for A/D conversion depending on scan and double-trigger mode settings

Scan mode setting	Double-trigger mode setting	Targets for A/D conversion				
		Self-diagnosis	Analog input (group A)	Analog input (group B)	Temperature sensor output	Internal reference voltage
Single scan	DBLE = 0	✓	✓	—	✓	✓
	DBLE = 1	—	✓ (1 ch only)	—	—	—
Continuous scan	DBLE = 0	✓	✓	—	—	—
	DBLE = 1	—	—	—	—	—
Group scan	DBLE = 0	✓	✓	✓	—	—
	DBLE = 1	—	✓ (1 ch only)	✓	—	—

Note: ✓: Selectable, —: Not selectable

ADST bit (A/D Conversion Start)

The ADST bit starts or stops the A/D conversion process. Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and the conversion target analog input.

[Setting conditions]

- 1 is written.
- The synchronous trigger (ELC, GPT) selected in the ADSTRGR.TRSA[5:0] bits is detected when ADCSR.EXTRG is 0 and ADCSR.TRGE is 1.
- The synchronous trigger (ELC, GPT) selected in the ADSTRGR.TRSB[5:0] bits is detected when ADCSR.TRGE is set to 1 in group scan mode.
- The asynchronous trigger is detected when the ADCSR.TRGE and ADCSR.EXTRG bits are set to 1 and the ADSTRGR.TRSA[5:0] bits are set to 0x00.

- When group priority operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), the ADGSPCR.GBRP bit is set to 1, and each time A/D conversion on the group with the lowest priority is started.

[Clearing conditions]

- 0 is written.
- The A/D conversion of all the selected channels, the temperature sensor output, and the internal reference voltage completes in single scan mode.
- Group A scan completes in group scan mode.
- Group B scan completes in group scan mode.
- When group priority operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), the ADGSPCR.GBRSCN bit is set to 1, and A/D conversion on the group with the lowest priority started by trigger completes.

Note: When group priority operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), do not set the ADST bit to 1.

Note: When group priority operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), do not set the ADST bit to 0. When forcing A/D conversion to terminate, follow the procedure for clearing the ADST bit.

29.2.9 ADANSA0 : A/D Channel Select Register A0

Base address: ADC120 = 0x4005_C000

Offset address: 0x004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ANSA10	ANSA9	ANSA8	ANSA7	ANSA6	ANSA5	—	—	ANSA2	ANSA1	ANSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	ANSA2 to ANSA0	A/D Conversion Channels Select Bit 2 (ANSA2) is associated with AN002 and bit 0 (ANSA0) is associated with AN000. 0: Do not select associated input channel. 1: Select associated input channel.	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
10:5	ANSA10 to ANSA5	A/D Conversion Channels Select Bit 10 (ANSA10) is associated with AN010 and bit 5 (ANSA5) is associated with AN005. 0: Do not select associated input channel. 1: Select associated input channel.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

ADANSA0 register selects analog input channels for A/D conversion. In group scan mode, this register selects group A channels.

Only set the ADANSA0 register when the ADCSR.ADST bit is 0.

ANSAn bits (A/D Conversion Channels Select) (n = 0 to 2, 5 to 10)

The ADANSA0 register selects any combination of analog input channels for A/D conversion. The channels and the number of channels can be arbitrarily set.

When performing A/D conversion of temperature sensor output or internal reference voltage, set the ADANSA0 register to 0x0000 to deselect all analog input channels.

In double trigger mode, the channels selected in the ADANSA0 register are invalid, and the channel selected in the ADCSR.DBLANS[4:0] bits is selected in group A instead.

When group scan mode is selected, do not select the channels specified in A/D Channel Select Register B0 (ADANSB0) and A/D Channel Select Register B1 (ADANSB1).

29.2.10 ADANSA1 : A/D Channel Select Register A1

Base address: ADC120 = 0x4005_C000

Offset address: 0x006

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	ANSA22	ANSA21	ANSA20	ANSA19	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
6:3	ANSA22 to ANSA19	A/D Conversion Channels Select Bit 6 (ANSA22) is associated with AN022 and bit 3 (ANSA19) is associated with AN019. 0: Do not select associated input channel. 1: Select associated input channel.	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W

ADANSA1 register selects analog input channels for A/D conversion. In group scan mode, this register selects group A channels.

Only set the ADANSA1 register when the ADCSR.ADST bit is 0.

ANSAn bits (A/D Conversion Channels Select) (n = 19 to 22)

The ADANSA1 register selects any combination of analog input channels for A/D conversion. The channels and the number of channels can be arbitrarily set.

When performing A/D conversion of temperature sensor output or internal reference voltage, set the ADANSA1 register to 0x0000 to deselect all analog input channels.

In double trigger mode, the channels selected in the ADANSA1 register are invalid, and the channel selected in the ADCSR.DBLANS[4:0] bits is selected in group A instead.

When group scan mode is selected, do not select the channels specified in A/D Channel Select Register B0 (ADANSB0) and A/D Channel Select Register B1 (ADANSB1).

29.2.11 ADANSB0 : A/D Channel Select Register B0

Base address: ADC120 = 0x4005_C000

Offset address: 0x014

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ANSB10	ANSB9	ANSB8	ANSB7	ANSB6	ANSB5	—	—	ANSB2	ANSB1	ANSB0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	ANSB2 to ANSB0	A/D Conversion Channels Select Bit 2 (ANSB2) is associated with AN002 and bit 0 (ANSB0) is associated with AN000. 0: Do not select associated input channel. 1: Select associated input channel.	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
10:5	ANSB10 to ANSB5	A/D Conversion Channels Select Bit 10 (ANSB10) is associated with AN010 and bit 5 (ANSB5) is associated with AN005. 0: Do not select associated input channel. 1: Select associated input channel.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

ADANSB0 selects analog input channels for A/D conversion in group B when group scan mode is selected. The ADANSB0 register is not used in any scan mode other than group scan mode.

Only set the ADANSB0 register when the ADCSR.ADST bit is 0.

ANSBn bits (A/D Conversion Channels Select) (n = 0 to 2, 5 to 10)

The ADANSB0 register selects any combination of analog input channels in group B for A/D conversion when group scan mode is selected. The ADANSB0 register is used for group scan mode only and not for any other modes.

Do not select channels specified in group A as selected in the ADANSA0 and ADANSA1 registers or the ADCSR.DBLANS[4:0] bits in double-trigger mode.

When performing A/D conversion of temperature sensor output or internal reference voltage, set the ADANSB0 register to 0x0000 to deselect all analog input channels.

29.2.12 ADANSB1 : A/D Channel Select Register B1

Base address: ADC120 = 0x4005_C000

Offset address: 0x016

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	ANSB22	ANSB21	ANSB20	ANSB19	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
6:3	ANSB22 to ANSB19	A/D Conversion Channels Select Bit 6 (ANSB22) is associated with AN022 and bit 3 (ANSB19) is associated with AN019. 0: Do not select associated input channel. 1: Select associated input channel.	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W

ADANSB1 selects analog input channels for A/D conversion in group B when group scan mode is selected. The ADANSB1 register is not used in any scan mode other than group scan mode.

Only set the ADANSB1 register when the ADCSR.ADST bit is 0.

ANSBn bits (A/D Conversion Channels Select) (n = 19 to 22)

The ADANSB1 register selects any combination of analog input channels in group B for A/D conversion when group scan mode is selected. The ADANSB1 register is used for group scan mode only and not for any other modes.

Do not select channels specified in group A as selected in the ADANSA0 and ADANSA1 registers or the ADCSR.DBLANS[4:0] bits in double-trigger mode.

When performing A/D conversion of temperature sensor output or internal reference voltage, set the ADANSB1 register to 0x0000 to deselect all analog input channels.

29.2.13 ADADS0 : A/D-Converted Value Addition/Average Channel Select Register 0

Base address: ADC120 = 0x4005_C000

Offset address: 0x008

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ADS10	ADS9	ADS8	ADS7	ADS6	ADS5	—	—	ADS2	ADS1	ADS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	ADS2 to ADS0	A/D-Converted Value Addition/Average Channel Select Bit 2 (ADS2) is associated with AN002 and bit 0 (ADS0) is associated with AN000. 0: Do not select associated input channel. 1: Select associated input channel.	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
10:5	ADS10 to ADS5	A/D-Converted Value Addition/Average Channel Select Bit 10 (ADS10) is associated with AN010 and bit 5 (ADS5) is associated with AN005. 0: Do not select associated input channel. 1: Select associated input channel.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

ADSn bits (A/D-Converted Value Addition/Average Channel Select) (n = 0 to 2, 5 to 10)

The ADSn bits determine which A/D-converted channels are subject to A/D-converted value addition/averaging. When an ADSn bit associated with a channel selected for A/D conversion is set to 1, A/D conversion of the analog input of the respective channel is performed successively 1, 2, 3, 4, or 16 times, as specified in the ADC[2:0] bits in the ADADC register.

When the ADADC.AVEE bit is 0, the value obtained by addition is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition is stored in the A/D data register.

The ADSn bits apply only to channels that are selected for A/D conversion in:

- The ANSAn bits in the ADANSA0 register or the DBLANS[4:0] bits in the ADCSR register
- The ANSBn bits in the ADANSB0 register

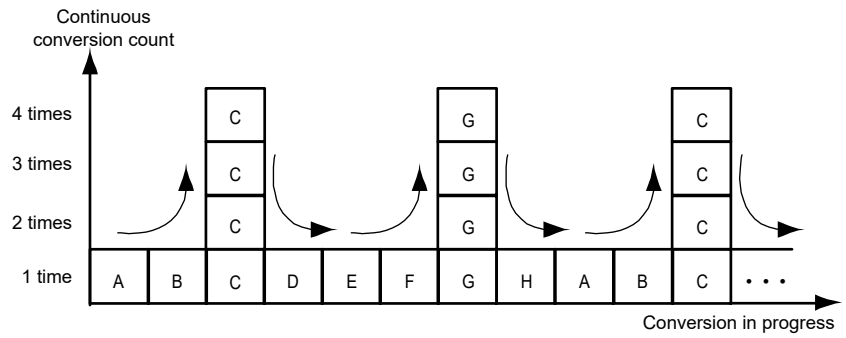
For channels on which the A/D conversion is performed and for which addition/average mode is not selected, a normal 1-time conversion is executed, and the conversion result is stored in the A/D data register.

Only set ADADS0 register bits when the ADCSR.ADST bit is 0.

Figure 29.2 shows a scanning operation sequence in which the ADADS0 register bits (channel c and g) are set to 1. In this figure:

- Addition mode is selected (ADADC.AVEE = 0)
- The number of conversions is set to 4 (ADADC.ADC[1:0] = 11b)
- The analog input channels (a to h) are selected by ADANSA0 register in continuous scan mode (ADCSR.ADCS[1:0] = 10b).

The conversion process begins with analog input A (channel a). The analog input C (channel c) conversion is performed successively 4 times and the added value is returned to A/D Data Register c (ADDRc). Next, the analog input D (channel d) conversion process is started. The analog input G (channel g) is performed successively 4 times and the added value is returned to A/D Data Register g (ADDRg). After conversion of analog input H (channel h), the conversion operation repeats in the same sequence starting with analog input A (channel a).



Note: A to H are analog inputs (channel a to h)

Figure 29.2 Scan conversion sequence with ADADC.ADC[2:0] = 011b, set 1 for analog inputs C and G by ADADS0/1

29.2.14 ADADS1 : A/D-Converted Value Addition/Average Channel Select Register 1

Base address: ADC120 = 0x4005_C000

Offset address: 0x00A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	ADS22	ADS21	ADS20	ADS19	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
6:3	ADS22 to ADS19	A/D-Converted Value Addition/Average Channel Select Bit 6 (ADS22) is associated with AN022 and bit 3 (ADS19) is associated with AN019. 0: Do not select associated input channel. 1: Select associated input channel.	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W

ADSn bits (A/D-Converted Value Addition/Average Channel Select) (n = 19 to 22)

The ADSn bits determine which A/D-converted channels are subject to A/D-converted value addition/averaging. When an ADSn bit associated with a channel selected for A/D conversion is set to 1, A/D conversion of the analog input of the respective channel is performed successively 1, 2, 3, 4, or 16 times, as specified in the ADC[2:0] bits in the ADADC register.

When the ADADC.AVEE bit is 0, the value obtained by addition is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition is stored in the A/D data register.

The ADSn bits apply only to channels that are selected for A/D conversion in:

- The ANSAn bits in the ADANSA1 register or the DBLANS[4:0] bits in the ADCSR register
- The ANSBn bits in the ADANSB1 register.

For channels on which the A/D conversion is performed and for which addition/average mode is not selected, a normal 1-time conversion is executed, and the conversion result is stored in the A/D data register.

Only set ADADS1 register when the ADCSR.ADST bit is 0.

29.2.15 ADADC : A/D-Converted Value Addition/Average Count Select Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x00C

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

Bit	Symbol	Function	R/W
2:0	ADC[2:0]	Addition/Average Count Select 0 0 0: 1-time conversion (no addition, same as normal conversion) 0 0 1: 2-time conversion (1 addition) 0 1 0: 3-time conversion (2 additions) 0 1 1: 4-time conversion (3 additions) 1 0 1: 16-time conversion (15 additions) Others: Setting prohibited	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	AVEE	Average Mode Select 0: Enable addition mode 1: Enable average mode	R/W

ADADC sets the addition or average mode and addition count for A/D conversion. Table 29.18 lists the settable combinations of ADADC register.

Table 29.18 Settable combinations of ADADC register

Mode select (AVEE)	Conversion time				
	1-time	2-time	3-time	4-time	16-time
Addition mode (AVEE = 0)	✓	✓	✓	✓	✓
Average mode (AVEE = 1)	—	✓	—	✓	—

Note: ✓: Selectable, —: Not selectable

ADC[2:0] bits (Addition/Average Count Select)

The ADC[2:0] bits set the addition count in all channels for which A/D conversion and addition/average mode are selected, including the channel selected in double trigger mode with the ADCSR.DBLANS[4:0] bits. The count also applies to A/D conversion of the temperature sensor output and the internal reference voltage.
When self-diagnosis is executed (ADCER.DIAGM = 1), do not set the ADC[2:0] bits to any value other than 000b.

AVEE bit (Average Mode Select)

The AVEE bit selects addition or average mode in all channels for which A/D conversion and addition/average mode are selected, including the channels selected in double-trigger mode in the ADCSR.DBLANS[4:0] bits, temperature sensor output, internal reference voltage.

29.2.16 ADCER : A/D Control Extended Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x00E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADRFMT	—	—	—	DIAGM	DIAGLD	DIAGVAL[1:0]		—	—	ACE	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 0. The write value should be 0.	R/W
5	ACE	A/D Data Register Automatic Clearing Enable 0: Disable automatic clearing 1: Enable automatic clearing	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
9:8	DIAGVAL[1:0]	Self-Diagnosis Conversion Voltage Select 0 0: Setting prohibited when self-diagnosis is enabled 0 1: 0 volts 1 0: Reference voltage*1 × 1/2 1 1: Reference voltage*1	R/W
10	DIAGLD	Self-Diagnosis Mode Select 0: Select rotation mode for self-diagnosis voltage 1: Select mixed mode for self-diagnosis voltage	R/W
11	DIAGM	Self-Diagnosis Enable 0: Disable ADC12 self-diagnosis 1: Enable ADC12 self-diagnosis	R/W
14:12	—	These bits are read as 0. The write value should be 0.	R/W
15	ADRFMT	A/D Data Register Format Select 0: Select right-justified for the A/D data register format 1: Select left-justified for the A/D data register format	R/W

Note 1. The reference voltage refers to VREFH0, AVCC0, or internal reference voltage (BGR) by ADHVREFCNT setting.

ACE bit (A/D Data Register Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all 0) of the ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, or ADOCDR register after any of these registers is read by the CPU or DTC. Automatic clearing of the A/D data registers enables detection of failures that are not updated in the A/D data registers. For details, see section 29.3.7. Usage Example of A/D Data Register Automatic Clearing Function.

DIAGVAL[1:0] bits (Self-Diagnosis Conversion Voltage Select)

The DIAGVAL[1:0] bits select the voltage value used in self-diagnosis fixed voltage mode. For details, see the DIAGLD bit description.

Do not execute self-diagnosis by setting the DIAGLD bit to 1 when the DIAGVAL[1:0] bits are set to 00b.

DIAGLD bit (Self-Diagnosis Mode Select)

The DIAGLD bit selects whether the three voltage values are rotated, or the fixed voltage is used in self-diagnosis.

Setting the DIAGLD bit to 0 selects conversion of the voltages in rotation mode, where 0 V, the reference voltage × 1/2, and the reference voltage are converted, in that order. After reset and when self-diagnosis voltage rotation mode is selected, self-diagnosis is executed from 0 V. The self-diagnosis voltage value does not return to 0 V when scan conversion completes. When scan conversion is restarted, rotation starts at the voltage value following the previous value.

Setting the DIAGLD bit to 1 selects fixed voltage, in which the fixed voltage specified in the ADCER.DIAGVAL[1:0] bits is converted. If fixed mode is switched to rotation mode, rotation starts at the fixed voltage value.

Only set the DIAGLD bit when the ADCSR.ADST bit is 0.

DIAGM bit (Self-Diagnosis Enable)

The DIAGM bit enables or disables self-diagnosis.

Self-diagnosis is used to detect a failure of the ADC12. In self-diagnosis mode, one of the three voltage values (0 V, the reference voltage × 1/2, or the reference voltage) is converted. When conversion completes, information on the converted voltage and the conversion result is stored into the A/D Self-Diagnosis Data Register (ADRD). The ADRD register can be read to determine whether the conversion result falls within the normal or abnormal range.

Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. In double trigger mode (ADCSR.DBLE = 1), self-diagnosis (DIAGM = 0) is deselected. When self-diagnosis is selected in group scan mode, self-diagnosis is executed separately for group A and group B.

Only set the DIAGM bit when the ADCSR.ADST bit is 0.

ADRFMT bit (A/D Data Register Format Select)

The ADRFMT bit specifies flush-right or flush-left for data to be stored in the ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR, ADCMPDR0/1, ADWINLLB, ADWINULB, or ADRD register.

Only set the ADRFMT bit when the ADCSR.ADST bit is 0.

29.2.17 ADSTRGR : A/D Conversion Start Trigger Select Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x010

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

Bit	Symbol	Function	R/W
5:0	TRSB[5:0]	A/D Conversion Start Trigger Select for Group B Select the A/D conversion start trigger for group B in group scan mode.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
13:8	TRSA[5:0]	A/D Conversion Start Trigger Select Select the A/D conversion start trigger in single scan mode and continuous scan mode. In group scan mode, the A/D conversion start trigger for group A is selected.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

TRSB[5:0] bits (A/D Conversion Start Trigger Select for Group B)

The TRSB[5:0] bits select the trigger to start scanning of the analog input selected in group B. The TRSB[5:0] bits must only be set in group scan mode and are not used in any other scan mode. For the scan conversion start trigger for group B, setting a software trigger or an asynchronous trigger is prohibited. In group scan mode, set the TRSB[5:0] bits to a value other than 0x00 and set the ADCSR.TRGE bit to 1.

When group A is given priority in group scan mode, setting the ADGSPCR.GBRP bit to 1 allows group B to continuously operate in single scan mode. When setting the ADGSPCR.GBRP bit to 1, set the TRSB[5:0] bits to 0x3F. The issuance period for a conversion trigger must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by the trigger might have no effect.

When the GPT module is selected as an A/D conversion start trigger, a delay for synchronization processing occurs. For details, see section 29.3.6. Analog Input Sampling and Scan Conversion Time.

Table 29.19 lists the A/D conversion startup sources selected in the TRSB[5:0] bits.

Table 29.19 Selection of A/D conversion start sources in the TRSB[5:0] bits (1 of 2)

Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
Trigger source deselected state	—	1	1	1	1	1	1
ELC_AD00	ELC	0	0	1	0	0	1
ELC_AD01	ELC	0	0	1	0	1	0
ELC_AD00, ELC_AD01	ELC	0	0	1	0	1	1
GPT0_ADTRGA	GPT0	0	1	0	0	0	1
GPT0_ADTRGB	GPT0	0	1	0	0	1	0
GPT1_ADTRGA	GPT1	0	1	0	0	1	1
GPT1_ADTRGB	GPT1	0	1	0	1	0	0

Table 29.19 Selection of A/D conversion start sources in the TRSB[5:0] bits (2 of 2)

Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
GPT2_ADTRGA	GPT2	0	1	0	1	0	1
GPT2_ADTRGB	GPT2	0	1	0	1	1	0
GPT3_ADTRGA	GPT3	0	1	0	1	1	1
GPT3_ADTRGB	GPT3	0	1	1	0	0	0
GPT0_ADTRGA or GPT0_ADTRGB	GPT0	0	1	1	0	0	1
GPT1_ADTRGA or GPT1_ADTRGB	GPT1	0	1	1	0	1	0
GPT2_ADTRGA or GPT2_ADTRGB	GPT2	0	1	1	0	1	1
GPT3_ADTRGA or GPT3_ADTRGB	GPT3	0	1	1	1	0	0

TRSA[5:0] bits (A/D Conversion Start Trigger Select)

The TRSA[5:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode, or the trigger to start scanning of group A analog inputs in group scan mode. When scanning is executed in group scan mode or double trigger mode, software trigger or asynchronous trigger is prohibited.

- When using a synchronous trigger (ELC, GPT), set the TRGE bit in the ADCSR register to 1 and set the EXTRG bit in the ADCSR register to 0.
- When using the asynchronous trigger (ADTRG0), set the TRGE bit in the ADCSR register to 1 and set the EXTRG bit in the ADCSR register to 1.
- Software trigger (ADCSR.ADST) is enabled regardless of the settings of the ADCSR.TRGE bit, the ADCSR.EXTRG bit, or the TRSA[5:0] bits.

The issuance period for a conversion trigger must be more than or equal to the actual scan conversion time (tSCAN). If the issuance period is less than tSCAN, A/D conversion by a trigger might have no effect.

When the GPT module is selected as an A/D conversion start trigger, a delay for synchronization processing occurs. For details, see section 29.3.6. Analog Input Sampling and Scan Conversion Time.

Table 29.20 lists the A/D conversion start sources selected in the TRSA[5:0] bits.

Table 29.20 Selection of A/D activation sources in the TRSA[5:0] bits (1 of 2)

Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
Trigger source deselected state	—	1	1	1	1	1	1
ADTRG0	Input pin for the trigger	0	0	0	0	0	0
ELC_AD00	ELC	0	0	1	0	0	1
ELC_AD01	ELC	0	0	1	0	1	0
ELC_AD00, ELC_AD01	ELC	0	0	1	0	1	1
GPT0_ADTRGA	GPT0	0	1	0	0	0	1
GPT0_ADTRGB	GPT0	0	1	0	0	1	0
GPT1_ADTRGA	GPT1	0	1	0	0	1	1
GPT1_ADTRGB	GPT1	0	1	0	1	0	0
GPT2_ADTRGA	GPT2	0	1	0	1	0	1

Table 29.20 Selection of A/D activation sources in the TRSA[5:0] bits (2 of 2)

Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
GPT2_ADTRGB	GPT2	0	1	0	1	1	0
GPT3_ADTRGA	GPT3	0	1	0	1	1	1
GPT3_ADTRGB	GPT3	0	1	1	0	0	0
GPT0_ADTRGA or GPT0_ADTRGB	GPT0	0	1	1	0	0	1
GPT1_ADTRGA or GPT1_ADTRGB	GPT1	0	1	1	0	1	0
GPT2_ADTRGA or GPT2_ADTRGB	GPT2	0	1	1	0	1	1
GPT3_ADTRGA or GPT3_ADTRGB	GPT3	0	1	1	1	0	0

29.2.18 ADEXICR : A/D Conversion Extended Input Control Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x012

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

Bit	Symbol	Function	R/W
0	TSSAD	Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select 0: Do not select addition/average mode for temperature sensor output. 1: Select addition/average mode for temperature sensor output.	R/W
1	OCSAD	Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select 0: Do not select addition/average mode for internal reference voltage. 1: Select addition/average mode for internal reference voltage.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	TSSA	Temperature Sensor Output A/D Conversion Select 0: Disable A/D conversion of temperature sensor output 1: Enable A/D conversion of temperature sensor output	R/W
9	OCSA	Internal Reference Voltage A/D Conversion Select 0: Disable A/D conversion of internal reference voltage 1: Enable A/D conversion of internal reference voltage	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

TSSAD bit (Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select)

When the TSSAD bit is set to 1, A/D conversion of the temperature sensor output is selected and performed successively the number of times specified in the ADC[2:0] bits in ADADC. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is returned to the A/D Temperature Sensor Data Register (ADTSDR). When the ADADC.AVEE bit is 1, the mean value is returned to ADTSDR.

Only set the TSSAD bit while the ADCSR.ADST bit is 0.

OCSAD bit (Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select)

When the OCSAD bit is set to 1, A/D conversion of the internal reference voltage is selected and performed successively the number of times specified in the ADC[2:0] bits in ADADC. When the ADADC.AVEE bit is 0, the value obtained

by addition (integration) is returned to the A/D Internal Reference Voltage Data Register (ADOCDR). When the ADADC.AVEE bit is 1, the mean value is returned to ADOCDR.

Only set the OCSAD bit while the ADCSR.ADST bit is 0.

TSSA bit (Temperature Sensor Output A/D Conversion Select)

The TSSA bit selects A/D conversion of the temperature sensor output. When executing the A/D conversion of the temperature sensor output:

1. Set all bits in the ADANSA0/1 and ADANSB0/1 registers, the ADCSR.DBLE bit, and the ADEXICR.OCSA bit to 0.
2. Execute the A/D conversion in single scan mode.

Only set the TSSA bit when the ADCSR.ADST bit is 0.

When executing A/D conversion of the temperature sensor output, the ADDISCR register is set to 0x0F and the A/D converter executes discharge (15 ADCLK) before executing sampling. The required sampling time is 5 µs or more.

The A/D converter executes discharge each time A/D conversion is executed on the temperature sensor output.

OCSA bit (Internal Reference Voltage A/D Conversion Select)

The OCSA bit selects A/D conversion of the internal reference voltage. When executing A/D conversion of the internal reference voltage:

1. Set all bits in the ADANSA0/1 and ADANSB0/1 registers, the ADCSR.DBLE bit, and the ADEXICR.TSSA bits to 0.
2. Execute the A/D conversion in single scan mode.

Only set the OCSA bit when the ADCSR.ADST bit is 0.

When executing A/D conversion of the internal reference voltage, the ADDISCR register are set to 0x0F and the A/D converter executes discharge (15 ADCLK) before executing sampling. The required sampling time is 5 µs or more.

The A/D converter executes discharge each time A/D conversion is executed on the internal reference voltage.

29.2.19 ADSSTRn/ADSSTRL/ADSSTRT/ADSSTRO : A/D Sampling State Register n (n = 0 to 2, 5 to 10)

Base address: ADC120 = 0x4005_C000

Offset address: 0x0E0 + 0x1 × n (n = 0 to 2, 5 to 10)
0x0DD (ADSSTRL)
0x0DE (ADSSTRT)
0x0DF (ADSSTRO)

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

Bit	Symbol	Function	R/W
7:0	SST[7:0]	Sampling Time Setting These bits set the sampling time in the range from 5 to 255 states.	R/W

The ADSSTRn register sets the sampling time for analog input.

The sampling time can be adjusted if the impedance of the analog input signal source is too high to secure sufficient sampling time, or if the ADCLK clock is slow. The set value indicates the time for one ADCLK cycle, and the required sampling time is specified by the voltage conditions. For details, see section 35.4. ADC12 Characteristics.

The lower limit of the sampling time setting depends on the frequency ratio:

- If the frequency ratio of PCLKB to PCLKD (ADCLK) = 1:1, the sampling time must be set to a value of more than 5 states
- If the frequency ratio of PCLKB to PCLKD (ADCLK) = 1:2 or 1:4, the sampling time must be set to a value of more than 6 states.

Table 29.21 shows the relationship between the A/D Sampling State Register and the associated channels. For details, see section 29.3.6. Analog Input Sampling and Scan Conversion Time.

Only set the SST[7:0] bits when the ADCSR.ADST bit is 0.

Table 29.21 Relationship between A/D sampling state register and associated channels

Bit name	Associated channels
ADSSTRn.SST[7:0] bits (n = 0 to 2, 5 to 10)*1	AN0n (n = 00 to 02, 05 to 10)*2
ADSSTRL.SST[7:0] bits	AN0n (n = 19 to 22)
ADSSTRT.SST[7:0] bits	Temperature sensor output*2
ADSSTRO.SST[7:0] bits	Internal reference voltage*2

Note 1. When the self-diagnosis function is selected, the sampling time set in the ADSSTR0.SST[7:0] bits is applied.
Note 2. When the temperature sensor output or the internal reference voltage is converted, set the sampling time to more than 5 µs. Because the maximum SST[7:0] value is 255 states, the ADCLK frequency must be 51 MHz or less to achieve 5 µs sampling time.

29.2.20 ADDISCR : A/D Disconnection Detection Control Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x07A

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

Bit	Symbol	Function	R/W
3:0	ADNDIS[3:0]	Disconnection Detection Assist Setting 0x0: The disconnection detection assist function is disabled 0x1: Setting prohibited Others: The number of states for the discharge or precharge period.	R/W
4	PCHG	Precharge/discharge select 0: Discharge 1: Precharge	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The ADDISCR register selects either precharge or discharge, and the period of precharge or discharge for the A/D disconnection detection assist function. Only set the ADDISCR register when the ADCSR.ADST bit is 0. When the temperature sensor output or internal reference voltage is converted, the A/D converter executes discharge automatically.

This operation is achieved by setting the ADDISCR register to 0x0F (15 ADCLK) when ADEXICR.OCSA or TSSA is set to 1. After executing discharge, the A/D converter executes sampling.

Disable the disconnection detection assist function if any of the following functions are used:

- The temperature sensor
- The internal reference voltage
- A/D self-diagnosis

ADNDIS[3:0] bits (Disconnection Detection Assist Setting)

The ADNDIS[3:0] bits specify the period of precharge or discharge. When ADNDIS[3:0] = 0000b, the disconnection detection assist function is disabled. Setting the ADNDIS[3:0] bits to 0001b is prohibited. Except when ADNDIS[3:0] = 0000b or 0001b, the specified value indicates the number of states for the period of precharge or discharge. When the ADNDIS[3:0] bits are set to any values other than 0000b or 0001b, the disconnection detection assistance function is enabled.

PCHG bit (Precharge/discharge select)

The PCHG bit selects either precharge or discharge.

29.2.21 ADACSR : A/D Conversion Operation Mode Select Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x07E

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	ADSAC	Successive Approximation Control Setting 0: Normal conversion mode (default) 1: Fast conversion mode	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

The ADACSR register selects the A/D conversion operation mode.

ADSAC bit (Successive Approximation Control Setting)

The ADSAC bit selects either normal conversion mode or fast conversion mode for A/D conversion.

When the ADSAC bit is 0, the ADCLK maximum frequency is 64 MHz. At high-speed A/D conversion mode (ADCSR.ADHSC = 0), the conversion time of successive approximation is 31.5 ADCLK. At Low-Power A/D conversion mode (ADCSR.ADHSC = 1), the conversion time of successive approximation is 40.5 ADCLK.

When the ADSAC bit is 1, the ADCLK maximum frequency is 48 MHz. At high-speed A/D conversion mode (ADCSR.ADHSC = 0), the conversion time of successive approximation is 21.5 ADCLK. At Low-Power A/D conversion mode (ADCSR.ADHSC = 1), the conversion time of successive approximation is 27.5 ADCLK.

For details, see section 29.3.6. Analog Input Sampling and Scan Conversion Time

29.2.22 ADGSPCR : A/D Group Scan Priority Control Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x080

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

Bit	Symbol	Function	R/W
0	PGS	Group Priority Operation Setting*1 0: Operate without group priority control. 1: Operate with group priority control.	R/W
1	GBRSCN	Lower-Priority Group Restart Setting (enabled only when PGS = 1 and reserved when PGS = 0.) 0: Disable rescanning of the group that was stopped in group priority operation 1: Enable rescanning of the group that was stopped in group priority operation.	R/W
14:2	—	These bits are read as 0. The write value should be 0.	R/W
15	GBRP	Single Scan Continuous Start*2 (enabled only when PGS = 1 and reserved when PGS = 0.) 0: Single scan is not continuously activated. 1: Single scan for the group with the lower-priority is continuously activated.	R/W

Note 1. The ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode) before setting PGS to 1. Operation is not guaranteed if these bits are set to any other value.

Note 2. When the GBRP bit is set to 1, single scan is performed continuously for the group with the lower-priority regardless of the setting in the GBRSCN bit.

PGS bit (Group Priority Operation Setting)

The PGS bit controls group priority operation in group scan mode. Set the PGS bit to 1 to enable group priority operation.

The ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode) before setting the PGS bit to 1. Operation is not guaranteed if the bits are set to any other value.

When the PGS bit is set to 0, a clear operation must be performed by software as described in section 29.8.3. Constraints on Stopping A/D Conversion. When the PGS bit is set to 1, use the settings described in section 29.3.4.3. Group Priority Operation.

GBRSCN bit (Lower-Priority Group Restart Setting)

The GBRSCN bit controls the restarting of scan operation in group priority operation.

When the GBRSCN bit is set to 1, if the scan operation of a lower-priority group is stopped by a trigger input of a priority group, the lower-priority group scanning is restarted on completion of the priority group scanning. If a trigger of a lower-priority group is input during scanning of the priority group, the lower-priority group scanning is started on completion of the priority group scanning.

When the GBRSCN bit is set to 0, triggers input during scanning are ignored. Set the GBRSCN bit while the ADCSR.ADST bit is 0.

GBRP bit (Single Scan Continuous Start)

The GBRP bit is set when a single scan operation is to be performed continuously on the group with the lower-priority.

Setting the GBRP bit to 1 starts a single scan of the group with the lower-priority. On completion of the scan, another single scan of the group with the lower-priority is started automatically. If scanning has been stopped during group priority operation, single scan of the group with the lower-priority is automatically restarted on completion of the A/D conversion of the priority group.

Before setting the GBRP bit to 1, disable input of a trigger for the lower-priority group. If the GBRP bit is set to 1, rescanning is performed only on the group with the lower-priority even if the GBRSCN bit is set to 0.

29.2.23 ADCMPCR : A/D Compare Function Control Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x090

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPAIE	WCMPE	CMPBIE	—	CMPAE	—	CMPBE	—	—	—	—	—	—	—	CMPAB[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CMPAB[1:0]	Window A/B Composite Conditions Setting These bits are valid when both window A and window B are enabled (CMPAE = 1 and CMPBE = 1). 0 0: Output ADC120_WCMPM when window A OR window B comparison conditions are met. Otherwise, output ADC120_WCMPUM. 0 1: Output ADC120_WCMPM when window A EXOR window B comparison conditions are met. Otherwise, output ADC120_WCMPUM. 1 0: Output ADC120_WCMPM when window A AND window B comparison conditions are met. Otherwise, output ADC120_WCMPUM. 1 1: Setting prohibited.	R/W
8:2	—	These bits are read as 0. The write value should be 0.	R/W
9	CMPBE	Compare Window B Operation Enable 0: Disable compare window B operation. Disable ADC120_WCMPM and ADC120_WCMPUM outputs. 1: Enable compare window B operation.	R/W

Bit	Symbol	Function	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	CMPAE	Compare Window A Operation Enable 0: Disable compare window A operation. Disable ADC120_WCMPM and ADC120_WCMPUM outputs. 1: Enable compare window A operation.	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	CMPBIE	Compare B Interrupt Enable 0: Disable ADC120_CMPBI interrupt when comparison conditions (window B) are met. 1: Enable ADC120_CMPBI interrupt when comparison conditions (window B) are met.	R/W
14	WCMPE	Window Function Setting 0: Disable window function Window A and window B operate as a comparator to compare the single value on the lower side with the A/D conversion result. 1: Enable window function Window A and window B operate as a comparator to compare the two values on the upper and lower sides with the A/D conversion result.	R/W
15	CMPAIE	Compare A Interrupt Enable 0: Disable ADC120_CMPAI interrupt when comparison conditions (window A) are met. 1: Enable ADC120_CMPAI interrupt when comparison conditions (window A) are met.	R/W

CMPAB[1:0] bits (Window A/B Composite Conditions Setting)

The CMPAB[1:0] bits are valid when both window A and window B are enabled (CMPAE = 1 and CMPBE = 1) in single scan mode. These bits specify the compare function match/mismatch event output conditions and monitoring conditions of ADWINMON.MONCOMB. Only set the CMPAB[1:0] bits while the ADCSR.ADST bit is 0.

CMPBE bit (Compare Window B Operation Enable)

The CMPBE bit enables or disables the compare window B operation. Set the CMPBE bit while the ADCSR.ADST bit is 0.

Set this bit to 0 before setting the following registers:

- A/D Channel Select Registers A0, A1, B0, B1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1)
- OCSA, or TSSA bits in the A/D Conversion Extended Input Control Register (ADEXICR)
- CMPCHB[5:0] bits in the Window B Channel Select Register (ADCMPBNSR)

CMPAE bit (Compare Window A Operation Enable)

The CMPAE bit enables or disables the compare window A operation. Set the CMPAE bit while the ADCSR.ADST bit is 0.

Set this bit to 0 before setting the following registers:

- A/D Channel Select Registers A0, A1, B0, B1 (ADANSA0, ADANSA1, ADANSB0, ADANSB1)
- TSSB, OCSA, or TSSA bits in the A/D Conversion Extended Input Control Register (ADEXICR)
- Window A Channel Select Registers 0 and 1 (ADCMPANSR0 and ADCMPANSR1)
- Window A Extended Input Select Register (ADCMPANSER)

CMPBIE bit (Compare B Interrupt Enable)

The CMPBIE bit enables or disables the ADC120_CMPBI interrupt output when the comparison conditions (window B) are met.

WCMPE bit (Window Function Setting)

The WCMPE bit enables or disables the window function. Set the WCMPE bit while the ADCSR.ADST bit is 0.

CMPAIE bit (Compare A Interrupt Enable)

The CMPAIE bit enables or disables the ADC120_CMPAI interrupt output when the comparison conditions (window A) are met.

29.2.24 ADCMPANSR0 : A/D Compare Function Window A Channel Select Register 0

Base address: ADC120 = 0x4005_C000

Offset address: 0x094

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CMPCHA10	CMPCHA9	CMPCHA8	CMPCHA7	CMPCHA6	CMPCHA5	—	—	CMPCHA2	CMPCHA1	CMPCHA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMPCHA2 to CMPCHA0	Compare Window A Channel Select Bit 2 (CMPCHA2) is associated with AN002 and bit 0 (CMPCHA0) is associated with AN000. 0: Disable compare function for associated input channel 1: Enable compare function for associated input channel	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
10:5	CMPCHA10 to CMPCHA5	Compare Window A Channel Select Bit 10 (CMPCHA10) is associated with AN010 and bit 5 (CMPCHA5) is associated with AN005. 0: Disable compare function for associated input channel 1: Enable compare function for associated input channel	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

CMPCHAn bits (Compare Window A Channel Select) (n = 0 to 2, 5 to 10)

The compare function is enabled by writing 1 to the CMPCHAn bits with the same number as the A/D conversion channel selected in the ADANSA0.ANSAn bits and the ADANSB0.ANSBn bits.

Set the CMPCHAn bits while the ADCSR.ADST bit is 0.

29.2.25 ADCMPANSR1 : A/D Compare Function Window A Channel Select Register 1

Base address: ADC120 = 0x4005_C000

Offset address: 0x096

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	CMPCHA22	CMPCHA21	CMPCHA20	CMPCHA19	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
6:3	CMPCHA22 to CMPCHA19	Compare Window A Channel Select Bit 6 (CMPCHA22) is associated with AN022 and bit 3 (CMPCHA19) is associated with AN019. 0: Disable compare function for associated input channel 1: Enable compare function for associated input channel	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W

CMPCHAn bits (Compare Window A Channel Select) (n = 19 to 22)

The compare function is enabled by writing 1 to the CMPCHAn bits with the same number as the A/D conversion channel selected in the ADANSA1.ANSA bits and the ADANSB1.ANSB bits.

Set the CMPCHAn bits while the ADCSR.ADST bit is 0.

29.2.26 ADCMPANSER : A/D Compare Function Window A Extended Input Select Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x092

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

Bit	Symbol	Function	R/W
0	CMPTSA	Temperature Sensor Output Compare Select 0: Exclude the temperature sensor output from the compare Window A target range. 1: Include the temperature sensor output in the compare Window A target range.	R/W
1	CMPOCA	Internal Reference Voltage Compare Select 0: Exclude the internal reference voltage from the compare Window A target range. 1: Include the internal reference voltage in the compare Window A target range.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

CMPTSA bit (Temperature Sensor Output Compare Select)

The compare window A function is enabled by setting the CMPTSA bit to 1 when the ADEXICR.TSSA bit is 1. Set the CMPTSA bit when the ADCSR.ADST bit is 0.

CMPOCA bit (Internal Reference Voltage Compare Select)

The compare window A function is enabled by setting the CMPOCA bit to 1 when the ADEXICR.OCSA bit is 1. Set the CMPOCA bit when the ADCSR.ADST bit is 0.

29.2.27 ADCMPLR0 : A/D Compare Function Window A Comparison Condition Setting Register 0

Base address: ADC120 = 0x4005_C000

Offset address: 0x098

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CMPLCHA10	CMPLCHA9	CMPLCHA8	CMPLCHA7	CMPLCHA6	CMPLCHA5	—	—	CMPLCHA2	CMPLCHA1	CMPLCHA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMPLCHA2 to CMPLCHA0	Compare Window A Comparison Condition Select These bits set comparison conditions for channels to which Window A comparison conditions are applied. Bit 2 (CMPLCHA2) is associated with AN002 and bit 0 (CMPLCHA0) is associated with AN000. Comparison conditions are shown in Figure 29.3. 0: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value > A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADCMPDR0 value, or ADCMPDR1 value < A/D-converted value 1: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): ADCMPDR0 value < A/D-converted value < ADCMPDR1 value	R/W

Bit	Symbol	Function	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
10:5	CMPLCHA10 to CMPLCHA5	Compare Window A Comparison Condition Select These bits set comparison conditions for channels to which Window A comparison conditions are applied. Bit 10 (CMPLCHA10) is associated with AN010 and bit 5 (CMPLCHA5) is associated with AN005. Comparison conditions are shown in Figure 29.3. 0: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value > A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADCMPDR0 value, or ADCMPDR1 value < A/D-converted value 1: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): ADCMPDR0 value < A/D-converted value < ADCMPDR1 value	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

CMPLCHAn bits (Compare Window A Comparison Condition Select) (n = 0 to 2, 5 to 10)

The CMPLCHAn bits specify the comparison conditions for channels to which Window A comparison conditions are applied. These bits can be set for each analog input to be compared. When the comparison result of each analog input meets the set condition, the ADCMPSR0.CMPSTCHAn flag sets to 1 and a compare interrupt (ADC120_CMPAI) is generated.

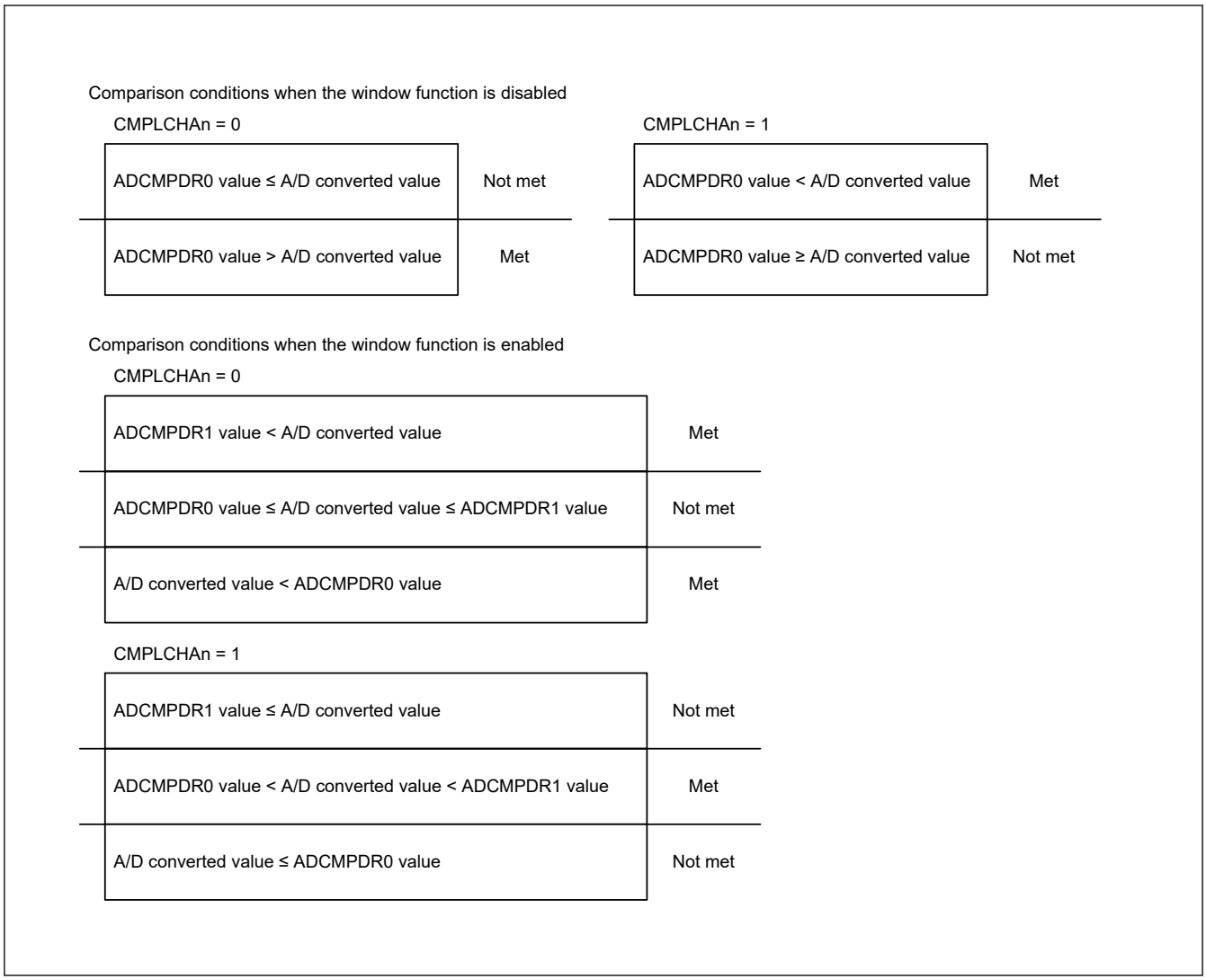


Figure 29.3 Explanation of comparison conditions for compare function Window A

29.2.28 ADCMPLR1 : A/D Compare Function Window A Comparison Condition Setting Register 1

Base address: ADC120 = 0x4005_C000

Offset address: 0x09A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	CMPLCHA22	CMPLCHA21	CMPLCHA20	CMPLCHA19	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
6:3	CMPLCHA22 to CMPLCHA19	Compare Window A Comparison Condition Select These bits set comparison conditions for channels to which Window A comparison conditions are applied. Bit 6 (CMPLCHA22) is associated with AN022 and bit 3 (CMPLCHA19) is associated with AN019. Comparison conditions are shown in Figure 29.3. 0: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value > A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADCMPDR0 value, or ADCMPDR1 value < A/D-converted value 1: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): ADCMPDR0 value < A/D-converted value < ADCMPDR1 value	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W

CMPLCHAn bits (Compare Window A Comparison Condition Select) (n = 19 to 22)

The CMPLCHAn bits specify the comparison conditions for analog channels to which window A comparison conditions are applied. These bits can be set for each analog input to be compared. When the comparison result of each analog input meets the set condition, the ADCMPSR1.CMPSTCHAn bit is set to 1 and a compare interrupt (ADC120_CMPAI) is generated.

29.2.29 ADCMPLER : A/D Compare Function Window A Extended Input Comparison Condition Setting Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x093

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

Bit	Symbol	Function	R/W
0	CMPLTSA	Compare Window A Temperature Sensor Output Comparison Condition Select Comparison conditions are shown in Figure 29.3. 0: When window function is disabled (ADCMPCR.WCMPE = 0) : ADCMPDR0 value > A/D-converted value Compare Window A Temperature Sensor Output Comparison Condition Select When window function is enabled (ADCMPCR.WCMPE = 1) : Compare Window A Temperature Sensor Output Comparison Condition A/D-converted value < ADCMPDR0 value, or A/D-converted value > ADCMPDR1 value 1: When window function is disabled (ADCMPCR.WCMPE = 0) : ADCMPDR0 value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1) : ADCMPDR0 value < A/D-converted value < ADCMPDR1 value	R/W
1	CMPLOCA	Compare Window A Internal Reference Voltage Comparison Condition Select Comparison conditions are shown in Figure 29.3. 0: When window function is disabled (ADCMPCR.WCMPE = 0) : ADCMPDR0 value > A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADCMPDR0 value, or A/D-converted value > ADCMPDR1 value 1: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): ADCMPDR0 value < A/D-converted value < ADCMPDR1 value	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

CMPLTSA bit (Compare Window A Temperature Sensor Output Comparison Condition Select)

The CMPLTSA bit specifies comparison conditions when the temperature sensor output is the target for the Window A comparison condition. When the temperature sensor output comparison result meets the set condition, the ADCMPSER.CMPSTTSA flag sets to 1 and a compare interrupt (ADC120_CMPAI) is generated.

CMPLOCA bit (Compare Window A Internal Reference Voltage Comparison Condition Select)

The CMPLOCA bit specifies comparison conditions when the internal reference voltage is the target for the Window A comparison condition. When the internal reference voltage comparison result meets the set condition, the ADCMPSER.CMPSTOCA flag sets to 1 and a compare interrupt (ADC120_CMPAI) is generated.

29.2.30 ADCMPDRn : A/D Compare Function Window A Lower-Side/Upper-Side Level Setting Register n (n = 0, 1)

Base address: ADC120 = 0x4005_C000

Offset address: 0x09C + (0x2 × n)

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

The ADCMPDRy (y = 0, 1) register specifies the reference data when the compare window A function is used. ADCMPDR0 sets the lower reference for window A, and ADCMPDR1 sets the upper reference for window A.

ADCMPDRy are read/write registers.

ADCMPDRy are writable even during A/D conversion. The reference data can be dynamically changed by rewriting register values during A/D conversion*1.

Set these registers so that the upper reference is not less than the lower reference (ADCMPDR1 ≥ ADCMPDR0). ADCMPDR1 are not used when the window function is disabled.

Note 1. The lower and the upper references are changed when each register is written. For example, when the upper reference value is changed and the lower reference value is being changed, the MCU compares the upper reference (after rewrite), and the lower reference (before rewrite) with the A/D conversion result. See Figure 29.4. If the comparison during the rewriting of these two references is erroneous, then rewrite these reference

values when both ADCSR.ADST and the target Compare Window Operation Enable bit (ADCMPCR.CMPAE or ADCMPCR.CMPBE) are 0.

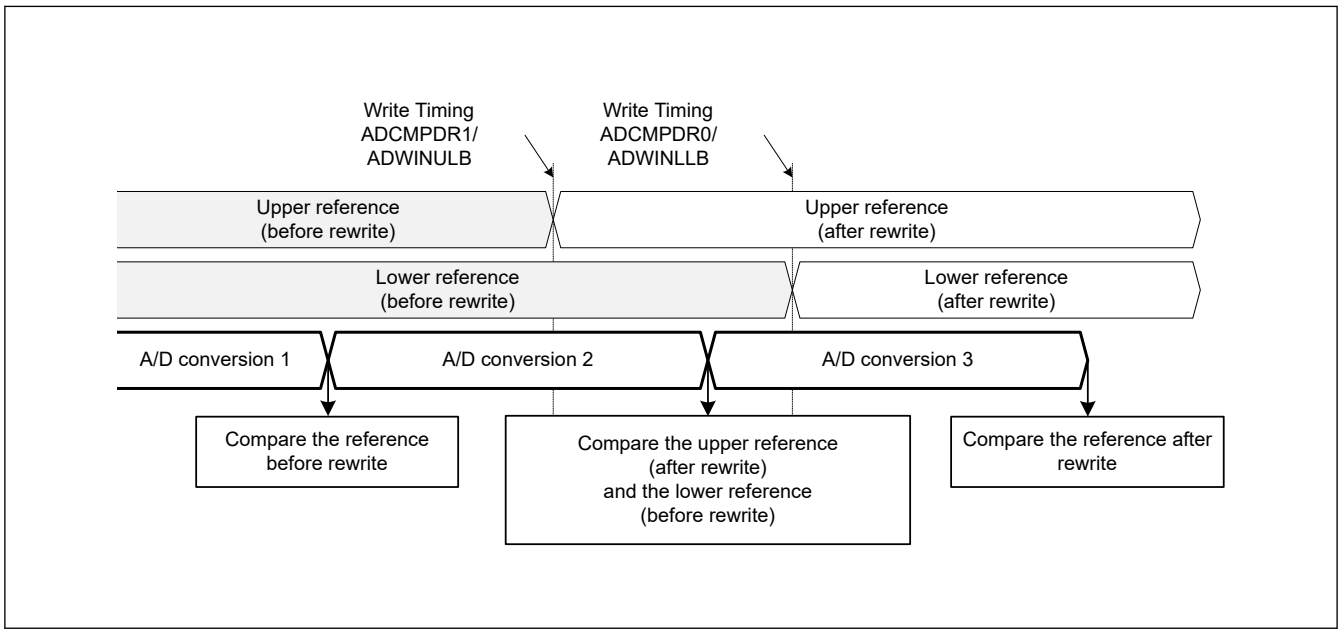


Figure 29.4 Comparison between upper and lower references before and after a rewrite

The ADCMPDRy registers use different formats depending on the following conditions:

- The value of A/D Data Register Format Select bit (flush-right or flush-left)
- The value of A/D-Converted Value Addition/Average Channel Select bits (A/D-converted value addition mode selected or not selected).

The data formats for each condition are shown as follows:

1. When A/D-converted value addition mode is not selected
   - Flush-right data with 12-bit accuracy — Lower 12 bits ([11:0]) are valid
   - Flush-left data with 12-bit accuracy — Upper 12 bits ([15:4]) are valid
2. When A/D-converted value addition mode is selected (other than 16-time conversion)
   - Flush-right data with 12-bit accuracy — Lower 14 bits ([13:0]) are valid
   - Flush-left data with 12-bit accuracy — Upper 14 bits ([15:2]) are valid
3. When A/D-converted value addition mode is selected (16-time conversion)
   - All bits ([15:0]) are valid

29.2.31 ADWINnLB : A/D Compare Function Window B Lower-Side/Upper-Side Level Setting Register n (n = L, U)

Base address: ADC120 = 0x4005_C000

Offset address: 0x0A8 (n = L)
0x0AA (n = U)

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

The ADWINULB and ADWINLLB registers specify the reference data when the compare window B function is used. ADWINLLB sets the lower reference for window B, and ADWINULB sets the upper reference for window B.

ADWINnLB are read/write registers.

ADWINnLB are writable even during A/D conversion. The reference data can be dynamically changed by rewriting register values during A/D conversion*1.

Set these registers so that the upper reference is not less than the lower reference (ADWINULB ≥ ADWINLLB). ADWINULB are not used when the window function is disabled.

Note 1. The lower and the upper references are changed when each register is written. For example, when the upper reference value is changed and the lower reference value is being changed, the MCU compares the upper reference (after rewrite), and the lower reference (before rewrite) with the A/D conversion result. See Figure 29.5. If the comparison during the rewriting of these two references is erroneous, then rewrite these reference values when both ADCSR.ADST and the target Compare Window Operation Enable bit (ADCMPCR.CMPAE or ADCMPCR.CMPBE) are 0.

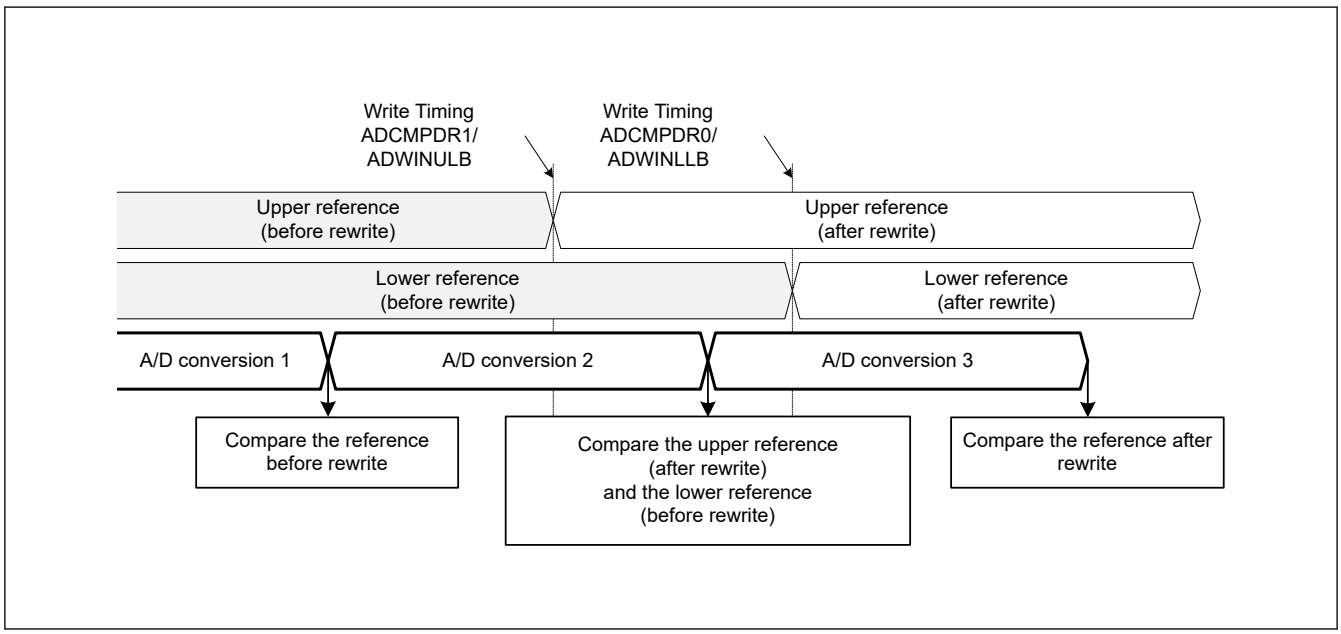


Figure 29.5 Comparison between upper and lower references before and after a rewrite

The ADWINnLB registers use different formats depending on the following conditions:

- The value of A/D Data Register Format Select bit (flush-right or flush-left)
- The value of A/D-Converted Value Addition/Average Channel Select bits (A/D-converted value addition mode selected or not selected).

The data formats for each condition are shown as follows:

1. When A/D-converted value addition mode is not selected
   - Flush-right data with 12-bit accuracy — Lower 12 bits ([11:0]) are valid
   - Flush-left data with 12-bit accuracy — Upper 12 bits ([15:4]) are valid
2. When A/D-converted value addition mode is selected (other than 16-time conversion)
   - Flush-right data with 12-bit accuracy — Lower 14 bits ([13:0]) are valid
   - Flush-left data with 12-bit accuracy — Upper 14 bits ([15:2]) are valid
3. When A/D-converted value addition mode is selected (16-time conversion)
   - All bits ([15:0]) are valid

29.2.32 ADCMPSR0 : A/D Compare Function Window A Channel Status Register 0

Base address: ADC120 = 0x4005_C000

Offset address: 0x0A0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CMPSTCHA10	CMPSTCHA9	CMPSTCHA8	CMPSTCHA7	CMPSTCHA6	CMPSTCHA5	—	—	CMPSTCHA2	CMPSTCHA1	CMPSTCHA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CMPSTCHA2 to CMPSTCHA0	Compare Window A Flag When Window A operation is enabled (ADCMPCR.CMPAE = 1b), these bits indicate the comparison result of channels to which Window A comparison conditions are applied. Bit 2 (CMPSTCHA2) is associated with AN002 and bit 0 (CMPSTCHA0) is associated with AN000. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
10:5	CMPSTCHA10 to CMPSTCHA5	Compare Window A Flag When Window A operation is enabled (ADCMPCR.CMPAE = 1b), these bits indicate the comparison result of channels to which Window A comparison conditions are applied. Bit 10 (CMPSTCHA10) is associated with AN010 and bit 5 (CMPSTCHA5) is associated with AN005. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

CMPSTCHAn flags (Compare Window A Flag) (n = 0 to 2, 5 to 10)

The CMPSTCHAn flags indicate the comparison results for channels to which Window A comparison conditions are applied. When a comparison condition set in ADCMPLR0.CMPLCHA is met at the end of A/D conversion, the associated CMPSTCHAn flag sets to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt request (ADC120_CMPAI) is generated when this flag sets to 1.

Writing 1 to the CMPSTCHAn flags is invalid.

[Setting condition]

- The condition set in ADCMPLR0.CMPLCHA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

29.2.33 ADCMPSR1 : A/D Compare Function Window A Channel Status Register 1

Base address: ADC120 = 0x4005_C000

Offset address: 0x0A2

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	CMPSTCHA22	CMPSTCHA21	CMPSTCHA20	CMPSTCHA19	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6:3	CMPSTCHA22 to CMPSTCHA19	Compare Window A Flag When Window A operation is enabled (ADCMPCR.CMPAE = 1), these bits indicate the comparison result of channels to which Window A comparison conditions are applied. Bit 6 (CMPSTCHA22) is associated with AN022 and bit 3 (CMPSTCHA19) is associated with AN019. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W

CMPSTCHAn flags (Compare Window A Flag) (n = 19 to 22)

The CMPSTCHAn flags indicate the comparison results for channels to which Window A comparison conditions are applied. When the comparison condition set in ADCMPLR1.CMPLCHA is met at the end of A/D conversion, the associated CMPSTCHAn flag sets to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt request (ADC120_CMPAI) is generated when this flag sets to 1.

Writing 1 to the CMPSTCHAn flags is invalid.

[Setting condition]

- The condition set in ADCMPLR1.CMPLCHA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

29.2.34 ADCMPSER : A/D Compare Function Window A Extended Input Channel Status Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x0A4

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

Bit	Symbol	Function	R/W
0	CMPSTTSA	Compare Window A Temperature Sensor Output Compare Flag When Window A operation is enabled (ADCMPCR.CMPAE = 1), this bit indicates the temperature sensor output comparison result. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
1	CMPSTOCA	Compare Window A Internal Reference Voltage Compare Flag When Window A operation is enabled (ADCMPCR.CMPAE = 1), this bit indicates the internal reference voltage comparison result. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

The ADCMPSER register stores compare results of compare function window A.

CMPSTTSA flag (Compare Window A Temperature Sensor Output Compare Flag)

The CMPSTTSA flag indicates the temperature sensor output comparison result. When the comparison condition set in ADCMPLER.CMPLTSA is met at the end of A/D conversion, this flag sets to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt request (ADC120_CMPAI) is generated when this flag sets to 1.

Writing 1 to the CMPSTTSA flag is invalid.

[Setting condition]

- The condition set in ADCMPLER.CMPLTSA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

CMPSTOCA flag (Compare Window A Internal Reference Voltage Compare Flag)

The CMPSTOCA flag indicates the internal reference voltage comparison result. When the comparison condition set in ADCMPLER.CMPLOCA is met at the end of A/D conversion, this flag sets to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt request (ADC120_CMPAI) is generated when this flag sets to 1.

Writing 1 to the CMPSTOCA flag is invalid.

[Setting condition]

- The condition set in ADCMPLER.CMPLOCA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

29.2.35 ADCMPBNSR : A/D Compare Function Window B Channel Select Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x0A6

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

Bit	Symbol	Function	R/W
5:0	CMPCHB[5:0]	Compare Window B Channel Select These bits select channels to be compared with the compare Window B conditions.	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W

CMPCHB[5:0]	Channel
0x00	AN000
0x01	AN001
0x02	AN002
0x05	AN005
⋮	⋮
0x0A	AN010
0x13	AN019
0x14	AN020
0x15	AN021
0x16	AN022
0x20	Temperature sensor
0x21	Internal reference voltage
0x3F	No selection
Others	Setting prohibited

Bit	Symbol	Function	R/W
7	CMPLB	Compare Window B Comparison Condition Setting This bit sets comparison conditions for channels for Window B. The comparison conditions are shown in Figure 29.6. 0: When window function is disabled (ADCMPCR.WCMPE = 0): ADWINLLB value > A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADWINLLB value, or ADWINULB value < A/D-converted value 1: When window function is disabled (ADCMPCR.WCMPE = 0): ADWINLLB value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): ADWINLLB value < A/D-converted value < ADWINULB value	R/W

CMPCHB[5:0] bits (Compare Window B Channel Select)

The CMPCHB[5:0] bits specify the channels to be compared with the compare Window B conditions from AN000 to AN002, AN005 to AN010, AN019 to AN022, the temperature sensor, the internal reference voltage. The compare Window B function is enabled by specifying the hexadecimal number of the A/D conversion channel selected in the ADANSA0, ADANSA1, ADANSB0, ADANSB1 registers.

Set the CMPCHB[5:0] bits while the ADCSR.ADST bit is 0.

CMPLB bit (Compare Window B Comparison Condition Setting)

The CMPLB bit specifies the comparison conditions for channels for Window B. When the comparison result of an analog input meets the set condition, the associated ADCMPBSR.CMPSTB flag sets to 1 and a compare interrupt request (ADC120_CMPBI) is generated.

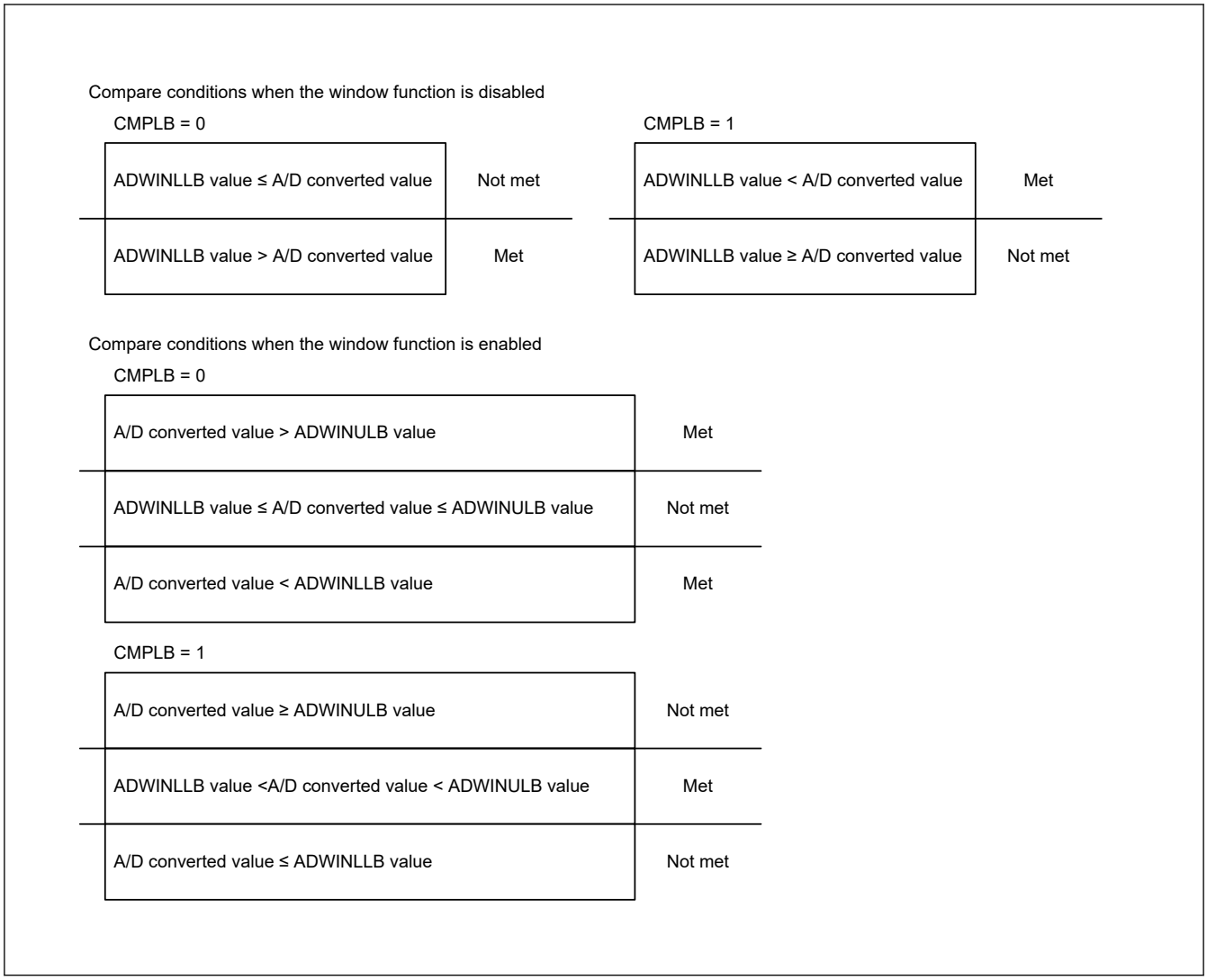


Figure 29.6 Explanation of compare conditions for compare function Window B

29.2.36 ADCMPBSR : A/D Compare Function Window B Status Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x0AC

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

Bit	Symbol	Function	R/W
0	CMPSTB	Compare Window B Flag When Window B operation is enabled (ADCMPCR.CMPBE = 1), this bit indicates the comparison result of channels to which Window B comparison conditions are applied, temperature sensor output, and internal reference voltage. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

CMPSTB flag (Compare Window B Flag)

The CMPSTB flag indicates the comparison result of channels to which Window B comparison conditions are applied, the temperature sensor output, and internal reference voltage. When the comparison condition set in ADCMPBNSR.CMPLB is

met at the end of A/D conversion, this flag sets to 1. When the ADCMPCR.CMPBIE bit is 1, a compare interrupt request (ADC120_CMPBI) is generated when this flag sets to 1.

Writing 1 to the CMPSTB flag is invalid.

[Setting condition]

- The condition set in ADCMPBNSR.CMPLB is met when ADCMPCR.CMPBE = 1.

[Clearing condition]

- Writing 0 after reading 1.

29.2.37 ADWINMON : A/D Compare Function Window A/B Status Monitor Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x08C

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

Bit	Symbol	Function	R/W
0	MONCOMB	Combination Result Monitor This bit indicates the combination result. This bit is valid when both Window A and Window B operations are enabled. 0: Window A/B composite conditions are not met. 1: Window A/B composite conditions are met.	R
3:1	—	These bits are read as 0.	R
4	MONCMPA	Comparison Result Monitor A 0: Window A comparison conditions are not met. 1: Window A comparison conditions are met.	R
5	MONCMPB	Comparison Result Monitor B 0: Window B comparison conditions are not met. 1: Window B comparison conditions are met.	R
7:6	—	These bits are read as 0.	R

MONCOMB bit (Combination Result Monitor)

The read-only MONCOMB bit indicates the combined result of comparison condition results A and B based on the combination condition set in the ADCMPCR.CMPAB[1:0] bits.

[Setting condition]

- The combined result meets the combination condition set in the ADCMPCR.CMPAB[1:0] bits when ADCMPCR.CMPAE = 1 and ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The combined result does not meet the combination condition set in the ADCMPCR.CMPAB[1:0] bits.
- ADCMPCR.CMPAE = 0 or ADCMPCR.CMPBE = 0.

MONCMPA bit (Comparison Result Monitor A)

The read-only MONCMPA bit is read as 1 when the A/D-converted value of the Window A target channel meets the condition set in ADCMPLR0/ADCMPLR1 and ADCMPLER. Otherwise, it is read as 0.

[Setting condition]

- The A/D-converted value meets the condition set in the ADCMPLR0/ADCMPLR1 and ADCMPLER registers when ADCMPCR.CMPAE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set in the ADCMPLR0/ADCMPLR1 and ADCMPLER registers when ADCMPCR.CMPAE = 1.
- ADCMPCR.CMPAE = 0 (automatically cleared when the ADCMPCR.CMPAE value changes from 1 to 0).

MONCMPB bit (Comparison Result Monitor B)

The read-only MONCMPB bit is read as 1 when the A/D-converted value of the Window B target channel meets the condition set in the ADCMPBNSR.CMPLB bit. Otherwise, it is read as 0.

[Setting condition]

- The A/D-converted value meets the condition set in ADCMPBNSR.CMPLB when ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set in ADCMPBNSR.CMPLB when ADCMPCR.CMPBE = 1.
- ADCMPCR.CMPBE = 0 (automatically cleared when the ADCMPCR.CMPBE value changes from 1 to 0).

29.2.38 ADHVREFCNT : A/D High-Potential/Low-Potential Reference Voltage Control Register

Base address: ADC120 = 0x4005_C000

Offset address: 0x08A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ADSLP	—	—	LVSEL	—	—	HVSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	HVSEL[1:0]	High-Potential Reference Voltage Select 0 0: AVCC0 is selected as the high-potential reference voltage 0 1: VREFH0 is selected as the high-potential reference voltage 1 0: Internal reference voltage is selected as the high-potential reference voltage 1 1: No reference voltage pin is selected (internal node discharge)	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	LVSEL	Low-Potential Reference Voltage Select 0: AVSS0 is selected as the low-potential reference voltage. 1: VREFL0 is selected as the low-potential reference voltage.	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	ADSLP	Sleep 0: Normal operation 1: Standby state	R/W

HVSEL[1:0] bits (High-Potential Reference Voltage Select)

The HVSEL[1:0] bits specify the high-potential reference voltage as VREFH0, AVCC0, or internal reference voltage (BGR).

When setting the register, make sure that HVSEL[1:0] = 11b is set.

Before selecting the internal reference voltage (HVSEL[1:0] = 10b), set HVSEL[1:0] = 11b to discharge the path of the high-potential reference voltage. After the discharge completes, set HVSEL[1:0] = 10b and start the A/D conversion.

When the internal reference voltage is selected as the high-potential reference voltage (HVSEL[1:0] = 10b), A/D conversion is possible for analog channels, but A/D conversion of the internal reference voltage and the temperature sensor output is prohibited. When the internal reference voltage is selected as the high-potential reference voltage (HVSEL[1:0] = 10b), can only work in low-current mode (ADCSR.ADHSC = 1).

LVSEL bit (Low-Potential Reference Voltage Select)

The LVSEL bit specifies the low-potential reference voltage as AVSS0 or VREFL0.

ADSLP bit (Sleep)

The ADSLP bit transitions the A/D converter to the standby state. Set the ADSLP bit to 1 only when modifying the ADCSR.ADHSC bit. In other cases, setting the ADSLP bit to 1 is prohibited.

After the ADSLP bit is set to 1, wait at least 5 µs before setting this bit to 0. Additionally, after the ADSLP bit is set to 0, wait at least 1 µs, then start the A/D conversion.

For the ADHSC bit rewriting procedure, see section 29.8.9. ADHSC Bit Rewriting Procedure.

29.3 Operation

29.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.

Scan conversion is performed in any of the three operating modes and two conversion modes:

- Single scan mode
- Continuous scan mode
- Group scan mode
- High-speed A/D conversion mode (include fast/normal conversion mode selected by ADACSR.ADSAC)
- Low-power A/D conversion mode (include fast/normal conversion mode selected by ADACSR.ADSAC)

In single scan mode, one or more specified channels are scanned once. In continuous scan mode, one or more specified channels are scanned repeatedly until software sets the ADCSR.ADST bit to 0. In group scan mode, the selected channels in group A, B are scanned once after scan starts in response to the respective synchronous trigger.

In single scan mode and continuous scan mode, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n. In group scan mode, A/D conversion is performed for the ANn channels in group A selected in the ADANSA0 and ADANSA1 registers, and for the ANn channels in group B selected in the ADANSB0 and ADANSB1 registers, starting from the channel with the smallest number n.

When self-diagnosis is selected, it is executed once at the beginning of each scan and one of the three reference voltages is converted.

It is prohibited to simultaneously select both temperature sensor output and internal reference voltage. If the internal reference voltage is selected as the reference voltage on the high potential side, A/D conversion of the temperature sensor or the internal reference voltage is also prohibited. When temperature sensor output or internal reference voltage is selected for A/D conversion, single scan mode should be used.

Double trigger mode can be used with single scan mode or group scan mode. With double trigger mode enabled (ADCSR.DBLE = 1), A/D conversion data of a channel selected in the ADCSR.DBLANS[4:0] bits is duplicated only if the conversion is started by the synchronous trigger (ELC, GPT) selected in the ADSTRGR.TRSA[5:0] bits. In group scan mode, only group A can use double trigger mode.

In the extended operation of double trigger mode, the A/D conversion operation is generated from the synchronous trigger combination selected in the ADSTRGR.TRSA[5:0] bits. In addition to normal double trigger mode operation, A/ D conversion data with odd number trigger (ELC_AD00) is stored in A/D Data Duplexing Register A (ADDBLDRA), and A/D conversion data with even number trigger (ELC_AD01) is stored in A/D Data Duplexing Register B (ADDBLDRB). In the extended operation of double trigger mode, when one of the trigger combinations occurs at the same time, the data duplexing register settings for the specified triggers do not work, and A/D conversion data is stored in A/D Data Duplexing Register B (ADDBLDRB).

The ADC12 ignores a synchronous trigger that occurs during the A/D conversion started by another synchronous trigger.

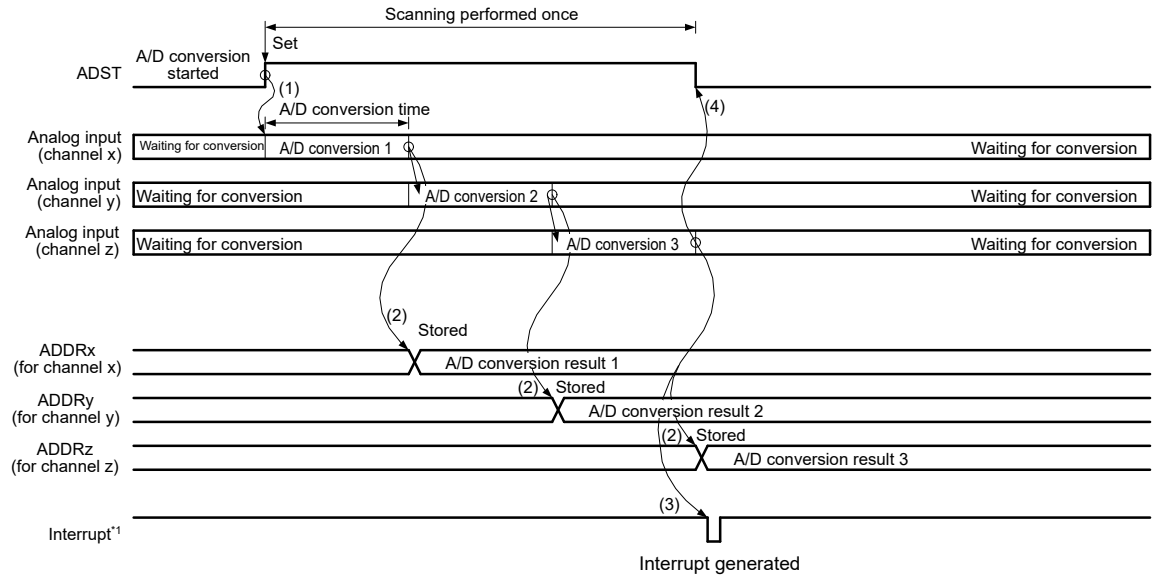
When any of the AN000 to AN002 channels are set as a channel-dedicated sample-and-hold circuit in the SHANS[2:0] bits in ADSHCR, the target analog input specified is sampled and held before the first A/D conversion of each scan.

29.3.2 Single Scan Mode

29.3.2.1 Basic Operation without channel-dedicated sample-and-hold circuits

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (ELC, GPT), or an asynchronous trigger input, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
2. Each time A/D conversion of a single channel is completed, the A/D conversion result is stored in the associated A/D data register (ADDRy).
3. When A/D conversion of all the selected channels is completed, an ADC120_ADI interrupt request is generated .
4. The ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when A/D conversion of all the selected channels is completed. The ADC12 then enters a wait state.



Note: x, y, and z (x < y < z) are the selected channel numbers by ADANSA0/1.
Note 1. ADC120_ADI

Figure 29.7 Example basic operation in single scan mode when the analog inputs (channel x to z) are selected

29.3.2.2 Basic Operation with channel-dedicated sample-and-hold circuits

When the channel-dedicated sample-and-hold circuit is used, sample-and-hold operation is first performed, and then A/D conversion is performed once on the analog input of all the specified channels. The channels whose dedicated sample-and-hold circuit is to be used can be selected in the SHANS[2:0] bits in ADSHCR.

The operation is as follows:

1. Analog input sampling of all channels whose dedicated sample-and-hold circuit is to be used starts when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (ELC, GPT), or asynchronous trigger input.
2. After sample-and-hold operation, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.

3. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D data register (ADDRy).
4. When A/D conversion of all the selected channels completes, an ADC120_ADI interrupt request is generated (no register setting).
5. The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then the ADC12 enters a wait state.

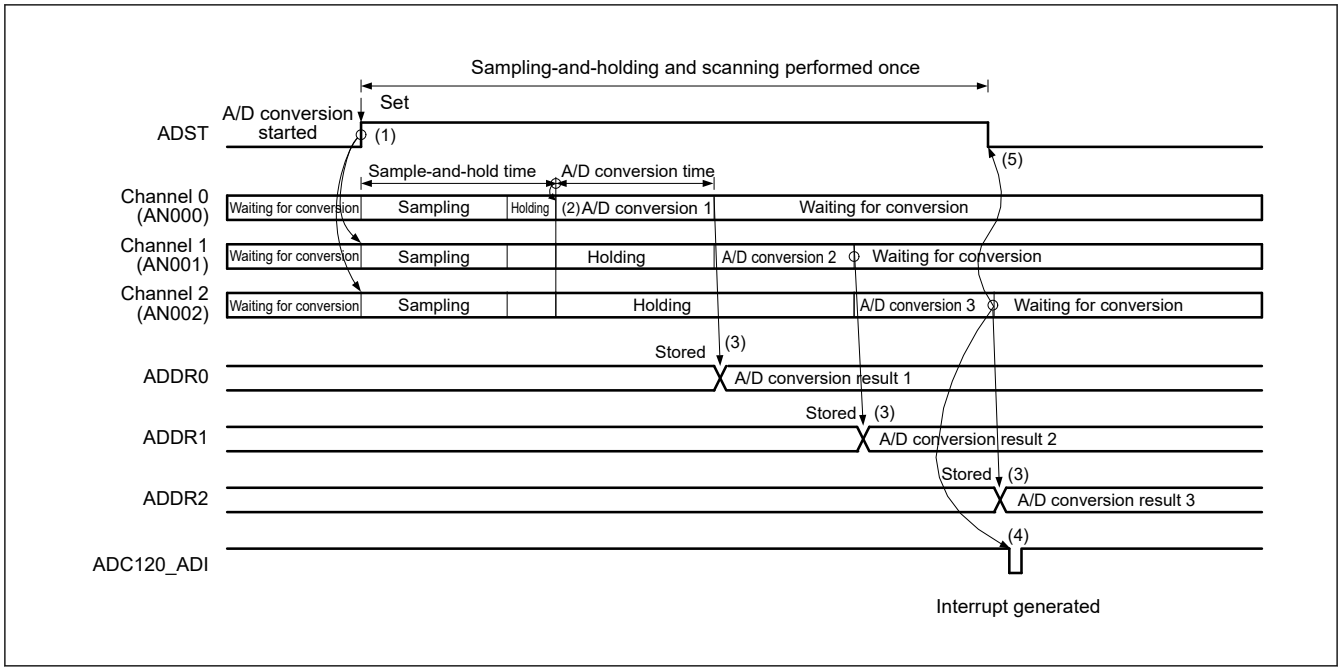
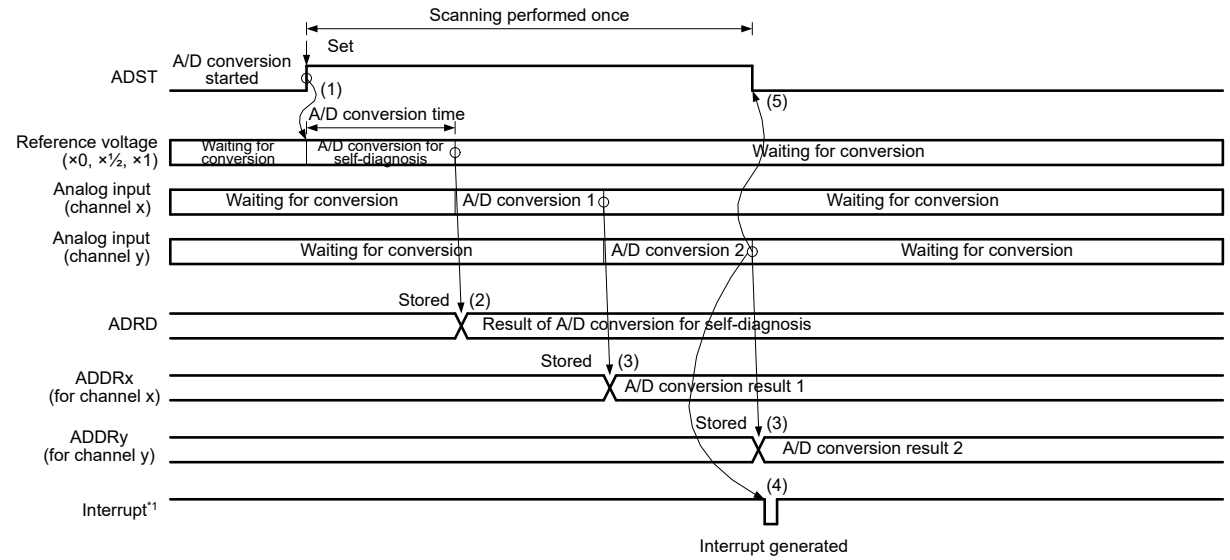


Figure 29.8 Example operation in single scan mode when the channel-dedicated sample-and-hold circuits are used and AN000 to AN002 are selected

29.3.2.3 Channel Selection and Self-Diagnosis without channel-dedicated sample-and-hold circuits

When channels and self-diagnosis are selected, A/D conversion is first performed for the reference voltage (×0, ×1/2, or ×1) , then A/D conversion is performed once on the analog input of the selected channels as follows:

1. A/D conversion for self-diagnosis is first started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger input, a synchronous trigger input (ELC, GPT), or an asynchronous trigger input.
2. When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel is completed, the A/D conversion result is stored in the associated A/D data register (ADDRy).
4. When A/D conversion of all the selected channels is completed, an ADC120_ADI interrupt request is generated .
5. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when A/D conversion of all the selected channels is completed. The ADC12 then enters a wait state.



Note: x and y (x < y) are the selected channel numbers by ADANSA0/1.
Note 1. ADC120_ADI

Figure 29.9 Example basic operation in single scan mode when the analog inputs (channel x and y) are selected with self- diagnosis

29.3.2.4 Channel Selection and Self-Diagnosis with channel-dedicated sample-and-hold circuits

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used, the sample-and-hold operation is performed first, and then A/D conversion is performed once for the reference voltage VREFH0 (unit 0) or VREFH (unit 1) (×0, ×1/2, or ×1) supplied to the ADC12. After that, A/D conversion is performed only once on the analog input of the selected channels.

The operation is as follows:

1. Analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (ELC, GPT), or asynchronous trigger input.
2. After sample-and-hold operation, A/D conversion for self-diagnosis starts.
3. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored into the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
4. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
5. When A/D conversion of all the selected channels completes, an ADC120_ADI interrupt request is generated (no register setting).
6. The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then the ADC12 enters a wait state.

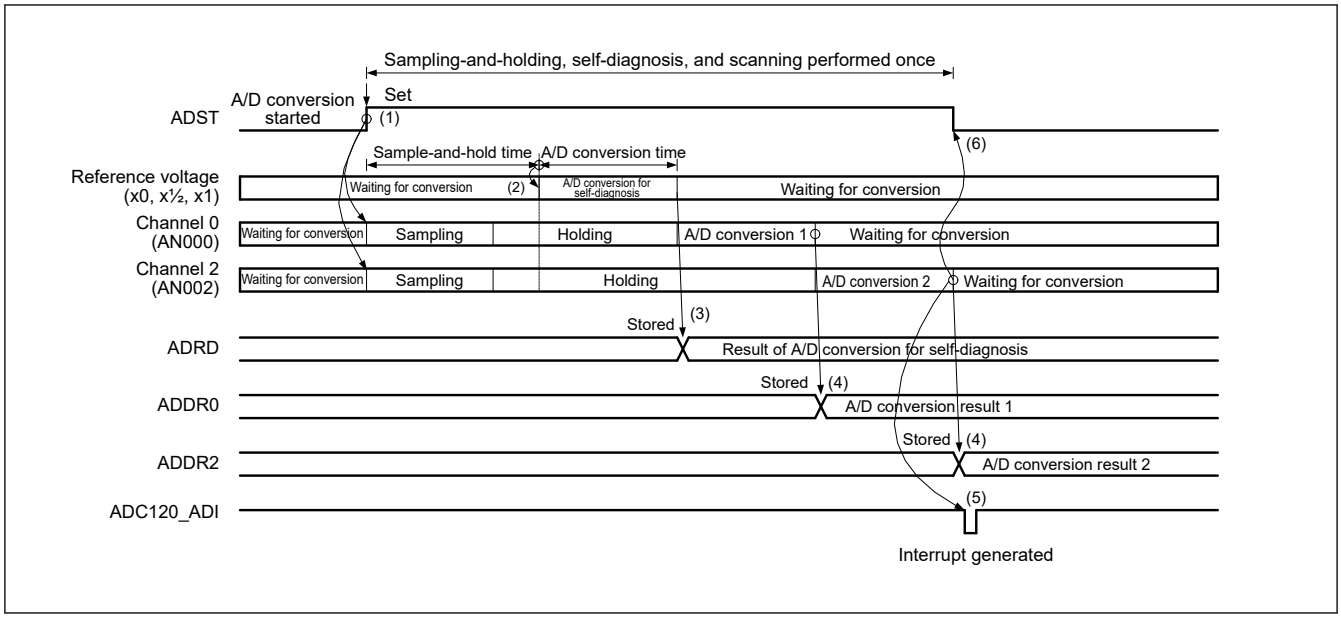


Figure 29.10 Example operation in single scan mode when channel-dedicated sample-and-hold circuits are used, AN000 and AN002 are selected with self-diagnosis

29.3.2.5 A/D Conversion of Temperature Sensor Output or Internal Reference Voltage

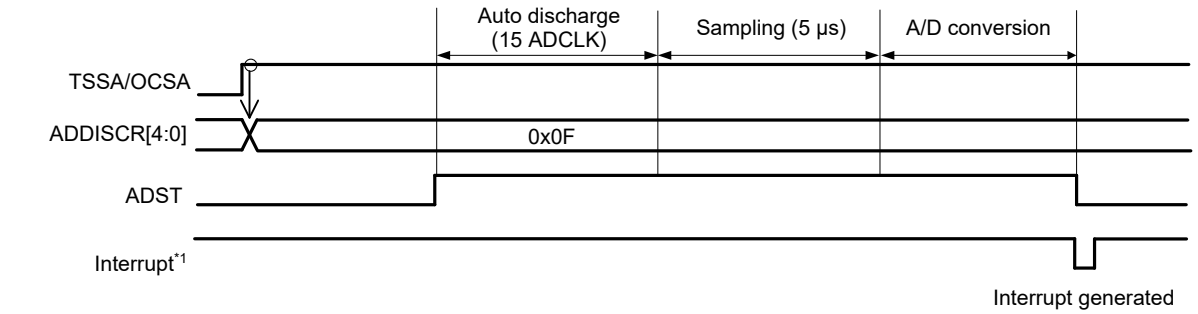
A/D conversion is performed on the temperature sensor output or the internal reference voltage in single scan mode as described in this section.

When selecting A/D conversion of the temperature sensor output or the internal reference voltage, deselect all analog input channels by setting the ADANSA0 and ADANSA1 registers to all 0's and the ADCSR.DBLE bit to 0.

When selecting A/D conversion of temperature sensor output, set the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) to 0 (deselected). When selecting A/D conversion of internal reference voltage, set the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) to 0 (deselected).

The operation is as follows:

1. Set the sampling time to 5 µs or longer. Take note of the sampling state register settings (ADSSTRT/ADSSTRO) and ADCLK frequency.
2. After switching to A/D conversion of internal reference voltage or temperature sensor output, set the ADST bit to 1 to start conversion.
3. On completion of A/D conversion, the result is stored in the Temperature Sensor Data Register (ADTSDR) or A/D Internal Reference Voltage Data Register (ADOCDR), and an ADC120_ADI interrupt request is generated.
4. The ADST bit remains 1 during A/D conversion and is automatically set to 0 on completion of A/D conversion. The ADC12 then enters a wait state.



Note 1. ADC120_ADI

Figure 29.11 Example basic operation in single scan mode when temperature sensor output or internal reference voltage is selected

29.3.2.6 A/D Conversion in Double-Trigger Mode

When double trigger mode is selected in single scan mode, two rounds of single scan operation started by a synchronous trigger (ELC, GPT) are performed in sequence.

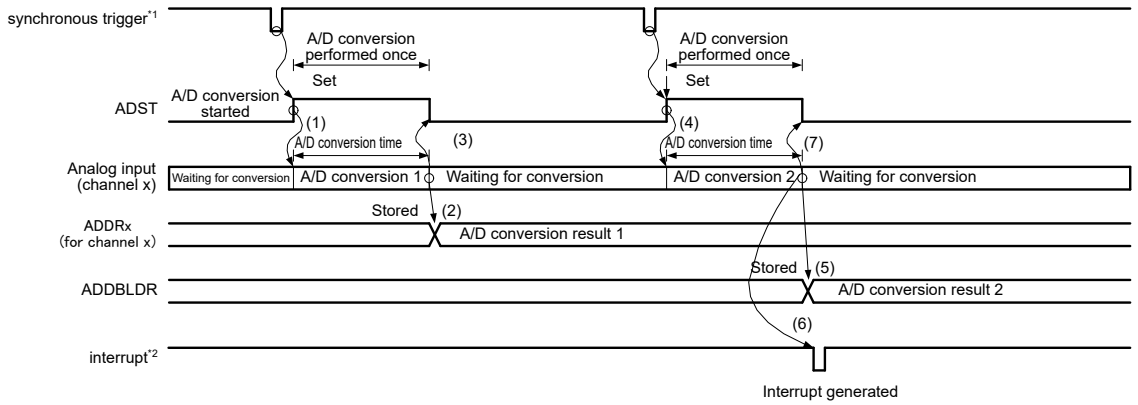
Deselect self-diagnosis and set the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) to 0.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated in the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid.

In double trigger mode, select a synchronous trigger (ELC, GPT) with the ADSTRGR.TRSA[5:0] bits. Additionally, set the ADCSR.EXTRG bit to 0 and the ADCSR.TRGE bit to 1. Do not use a software trigger.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a synchronous trigger input (ELC, GPT), A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
2. Each time A/D conversion of a single channel is completed, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
3. The ADCSR.ADST bit is automatically set to 0 and the ADC12 enters a wait state. An ADC120_ADI interrupt request is not generated .
4. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger input, A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
5. When A/D conversion is completed, the result is stored in the A/D Data Duplexing Register (ADDBLDR), which is exclusively used in double-trigger mode.
6. An ADC120_ADI interrupt request is generated.
7. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when A/D conversion is completed. Then the ADC12 enters a wait state.



Note: x is the selected channel number by ADANSA0/1.
Note 1. ELC_AD00/ELC_AD01
Note 2. ADC120_ADI

Figure 29.12 Example operation in single scan mode when double-trigger mode is selected and the analog input (channel x) is duplicated

29.3.2.7 Extended Operations When Double-Trigger Mode Is Selected

When double trigger mode is selected in single scan mode, and a synchronous trigger (ELC_AD00/ELC_AD01) is selected as the trigger for the start of A/D conversion, two rounds of single scan operation are performed.

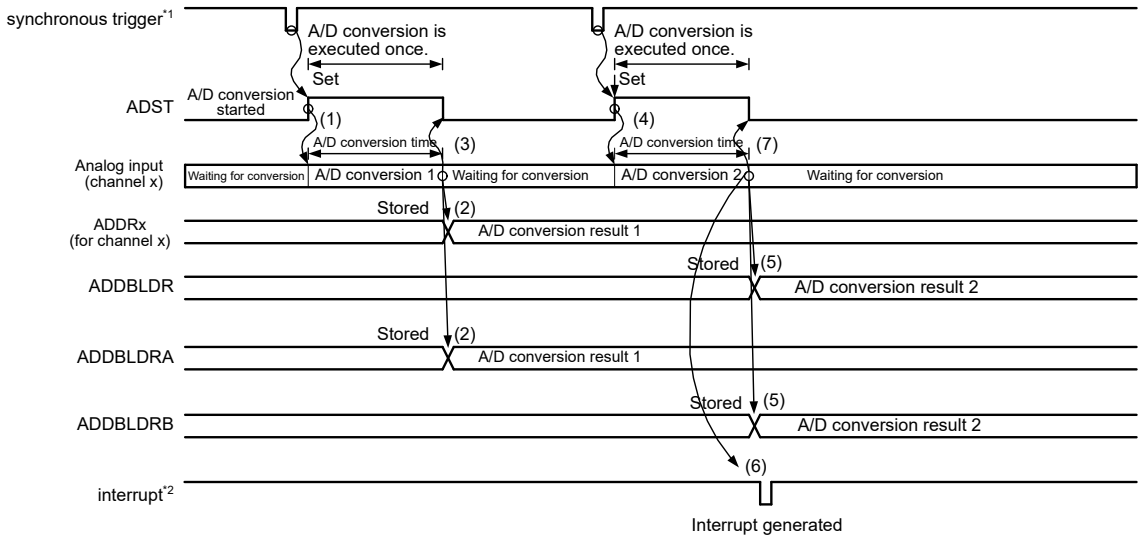
Deselect self-diagnosis and set the temperature sensor output A/D conversion select bit (ADEXICR.TSSA), and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) to 0.

Duplication of A/D conversion data is enabled by setting the channel number to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid.

In extended double trigger mode, select a synchronous trigger combination ELC_AD00/ELC_AD01 by setting the ADSTRGR.TRSA[5:0] bits to 0x0B, set the ADCSR.EXTRG bit to 0, and set the ADCSR.TRGE bit to 1. Do not use a software trigger.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a synchronous trigger input (ELC_AD00/ ELC_AD01), A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
2. When A/D conversion completes, the A/D conversion result is stored in the associated A/D Data Register (ADDRy) and in A/D Data Duplexing Register A (ADDBLDRA) or A/D Data Duplexing Register B (ADDBLDRB) when the ELC_ADi0 or ELC_ADi1 trigger is input respectively (i = 0).
3. The ADCSR.ADST bit is automatically set to 0 and the ADC12 enters a wait state. An ADC120_ADI interrupt request is not generated.
4. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger (ELC_AD00/ELC_AD01), A/ D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
5. When A/D conversion completes, the A/D conversion result is stored in the A/D Data Duplexing Register (ADDBLDR) and in A/D Data Duplexing Register A (ADDBLDRA) or A/D Data Duplexing Register B (ADDBLDRB) when the ELC_ADi0 or ELC_ADi1 trigger is input respectively (i = 0).
6. An ADC120_ADI interrupt request is generated.
7. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when A/D conversion completes. The ADC12 then enters a wait state.



Note: x is the selected channel number by ADANSA0/1.
Note 1. ELC_AD00/ELC_AD01
Note 2. ADC120_ADI

Figure 29.13 Example extended operation in double trigger mode with duplication selected for the analog input (channel x) and ELC_AD00/ELC_AD01

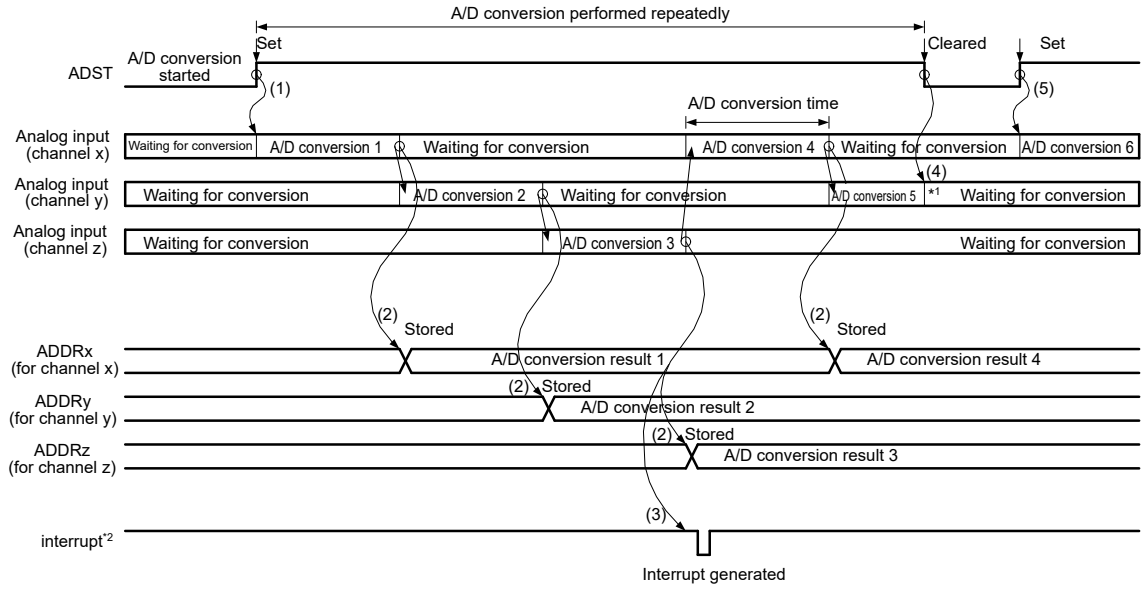
29.3.3 Continuous Scan Mode

29.3.3.1 Basic Operation without channel-dedicated sample-and-hold circuits

In continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels. In this mode, deselect the temperature sensor output A/D conversion and the internal reference voltage A/D conversion by setting the ADEXICR.TSSA and ADEXICR.OCSA bits to 0.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (ELC, GPT), or an asynchronous trigger input, A/D conversion is performed for ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
2. Each time A/D conversion of a single channel is completed, the A/D conversion result is stored in the associated A/D Data Register (ADDRy).
3. When A/D conversion of all the selected channels is completed, an ADC120_ADI interrupt request is generated. The ADC12 sequentially starts A/D conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
4. The ADCSR.ADST bit is not automatically cleared, and steps 2. and 3. are repeated as long as ADCSR.ADST remains 1 (A/ D conversion start). When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
5. When the ADCSR.ADST bit is later set to 1 (A/D conversion start), A/D conversion starts again for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.



Note: x, y, and z (x < y < z) are the selected channel numbers by ADANSA0/1.
Note 1. Data for A/D conversion 5 is ignored.
Note 2. ADC120_ADI

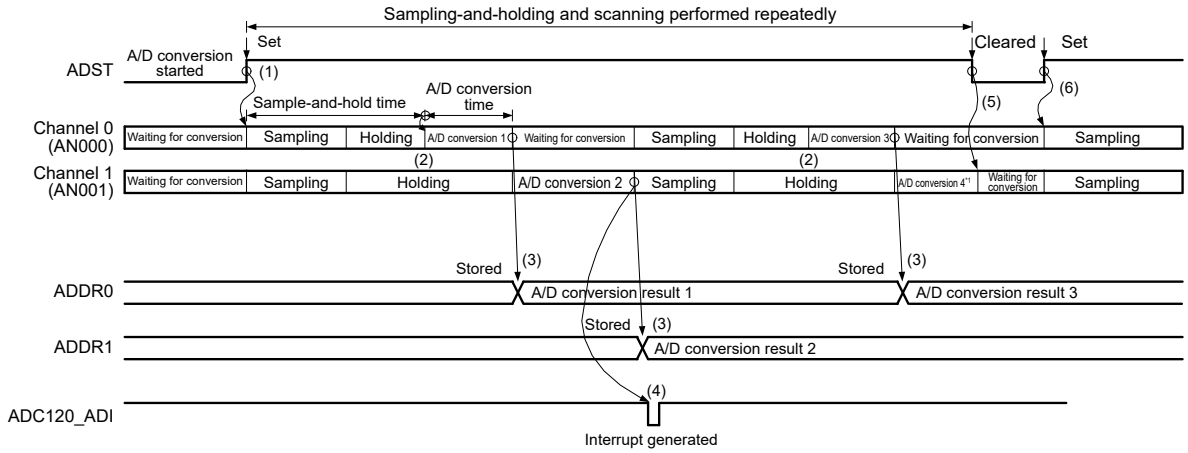
Figure 29.14 Example basic operation in continuous scan mode when the analog inputs (channel x to z) are selected

29.3.3.2 Basic Operation with channel-dedicated sample-and-hold circuits

When the channel-dedicated sample-and-hold circuit is used, sample-and-hold operation is first performed, and then A/D conversion is repeated on the analog input of all the specified channels. The channels whose dedicated sample-and-hold circuit is to be used can be selected in the SHANS[2:0] bits in ADSHCR.

The operation is as follows:

1. Analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (ELC, GPT), or asynchronous trigger input.
2. After sample-and-hold operation, A/D conversion is performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
4. When A/D conversion of all the selected channels completes, an ADC120_ADI interrupt request is generated (no register setting). At the same time, analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used.
5. The ADST bit is not automatically cleared, and steps 2 to 4 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
6. When the ADST bit is then set to 1 (A/D conversion start), analog input sampling starts again for all channels whose dedicated sample-and-hold circuit is to be used.



Note 1. Data for A/D conversion 4 is ignored.

Figure 29.15 Example operation in continuous scan mode when channel-dedicated sample-and-hold circuits are used and AN000 and AN001 are selected

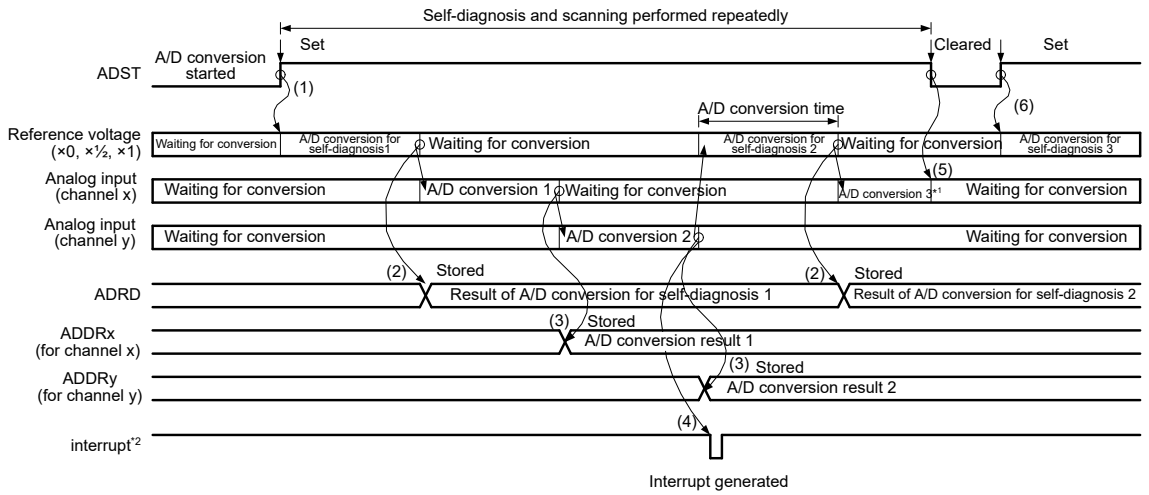
29.3.3.3 Channel Selection and Self-Diagnosis without channel-dedicated sample-and-hold circuits

When channels and self-diagnosis are selected at the same time, A/D conversion is first performed for the reference voltage (×0, ×1/2, or ×1) supplied to the ADC12, and A/D conversion is performed on the analog input of the selected channels. This sequence is repeated as described in the section that follows.

In continuous scan mode, deselect the temperature sensor output A/D conversion and the internal reference voltage A/D conversion by setting the ADEXICR.TSSA and ADEXICR.OCSA bits to 0.

The operation is as follows:

1. A/D conversion for self-diagnosis is first started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger input, a synchronous trigger input (ELC, GPT), or an asynchronous trigger input.
2. When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel is completed, the A/D conversion result is stored in the corresponding A/D Data Register (ADDRy).
4. When A/D conversion of all the selected channels is completed, an ADC120_ADI interrupt request is generated. At the same time, the ADC12 starts A/D conversion for self-diagnosis and then on the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
5. The ADCSR.ADST bit is not automatically cleared, and steps 2. to 4. are repeated as long as the ADCSR.ADST bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
6. When the ADST bit is later set to 1 (A/D conversion start), the A/D conversion for self-diagnosis is started again.



Note: x and y (x < y) are the selected channel numbers by ADANSA0/1.
Note 1. Data for A/D conversion 3 is ignored.
Note 2. ADC120_ADI

Figure 29.16 Example basic operation in continuous scan mode when the analog inputs (channel x and y) are selected with self-diagnosis

29.3.3.4 Channel Selection and Self-Diagnosis with channel-dedicated sample-and-hold circuits

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used, sample-and-hold operation is first performed, and then A/D conversion is performed for the reference voltage VREFH0 (unit 0) or VREFH (unit 1) (×0, ×1/2, or ×1) supplied to the ADC12, and then A/D conversion is performed on the analog input of the selected channels, and this sequence is repeated.

The operation is as follows:

1. Analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (ELC, GPT), or asynchronous trigger input.
2. After sample-and-hold operation, A/D conversion for self-diagnosis starts.
3. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
4. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
5. When A/D conversion of all the selected channels completes, an ADC120_ADI interrupt request is generated (no register setting). At the same time, analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used.
6. The ADST bit is not automatically cleared, and steps 2 to 5 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
7. When the ADST bit is then set to 1 (A/D conversion start), analog input sampling starts again for all channels whose dedicated sample-and-hold circuit is to be used.

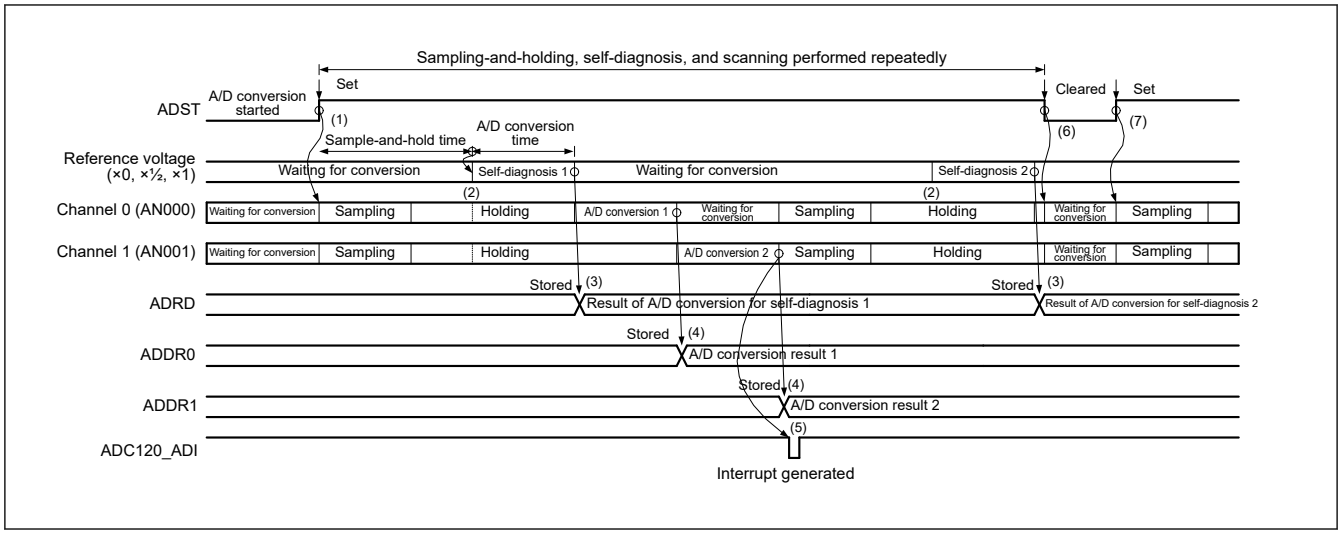


Figure 29.17 Example operation in continuous scan mode when channel-dedicated sample-and-hold circuits are used and AN000 and AN001 are selected with self-diagnosis

29.3.4 Group Scan Mode

29.3.4.1 Basic Operation

In group scan mode, A/D conversion is performed once on the analog input of all the specified channels in group A and B after scanning is started by a synchronous trigger (ELC, GPT). The scan operation of each group is similar to the scan operation in single scan mode.

The synchronous triggers can be selected in the ADSTRGR.TRSA[5:0] bits for group A and in the ADSTRGR.TRSB[5:0] bits for group B. Use different triggers for group A and B to prevent simultaneous A/D conversion of the two groups. Do not use a software trigger.

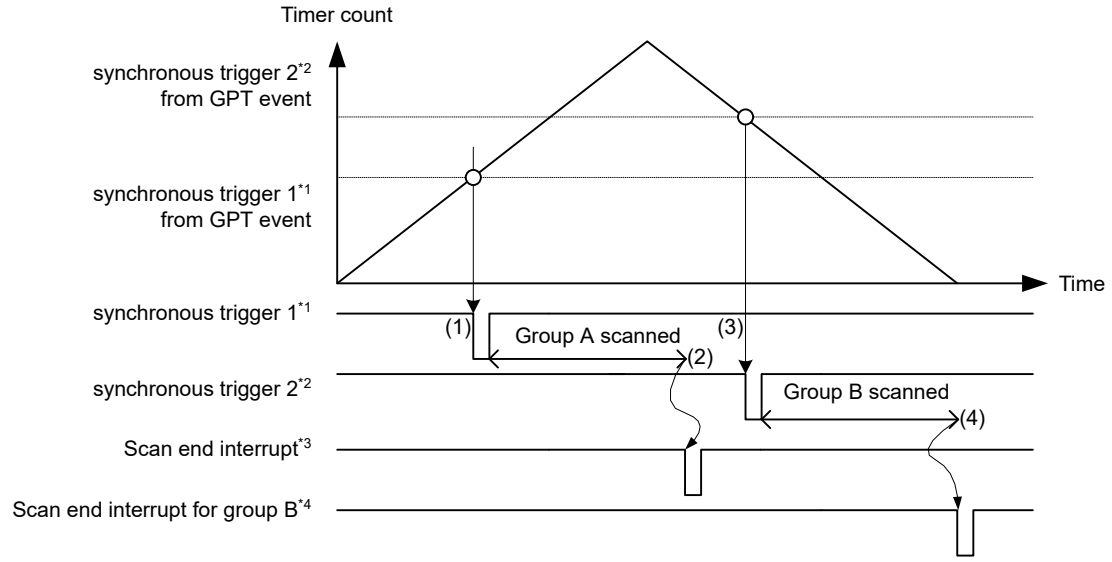
The group A channels to be A/D-converted are selected using the ADANSA0 and ADANSA1 registers. The group B channels to be A/D-converted are selected using the ADANSB0 and ADANSB1 registers. Group A and B cannot use the same channels.

In group scan mode, deselect the temperature sensor output A/D conversion and the internal reference voltage A/D conversion by setting the ADEXICR.TSSA and (ADEXICR.OCSA) bits to 0. When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed for Group A and B.

The following sequence describes operation in group scan mode using a synchronous trigger from the ELC or GPT. In this example, the ELC_AD00 trigger from the ELC is used to start conversion of group A and the ELC_AD01 trigger from the ELC is used to start conversion of group B. In addition, ELC_AD00 and ELC_AD01 are selected for the GPT event in the associated ELC.ELSRn registers.

The operation is as follows:

1. Scanning of group A is started by ELC_AD00.
2. When group A scanning completes, an ADC120_ADI interrupt is generated (no register setting).
3. Scanning of group B is started by ELC_AD01.
4. When group B scanning completes, an ADC120_GBADI interrupt is generated if the ADCSR.GBADIE bit is 1 (ADC120_GBADI interrupt when scanning completion is enabled).



Note 1. ELC_AD00
Note 2. ELC_AD01
Note 3. ADC120_ADI
Note 4. ADC120_GBADI

Figure 29.18 Example basic operation in group scan mode when synchronous triggers from the ELC are used

29.3.4.2 A/D Conversion in Double-Trigger Mode

When double trigger mode is selected in group scan mode, two rounds of single scan operation started by a synchronous trigger (ELC, GPT) are performed as a sequence for group A. For group B, single scan operation started by a synchronous trigger (ELC, GPT) is performed once.

In group scan mode, the synchronous trigger can be selected in the ADSTRGR.TRSA[5:0] bits for group A and in the ADSTRGR.TRSB[5:0] bits for group B. Use different triggers for group A, B to prevent simultaneous A/D conversion of the two groups. Do not use a software trigger or an asynchronous trigger.

When an ELC_AD00/ELC_AD01 is selected as group A synchronous triggers by setting the ADSTRGR.TRSA[5:0] bits to 0x0B, operation proceeds in extended double trigger mode.

The group A channel to be A/D-converted is selected using the DBLANS[4:0] bits in the ADCSR register, while the group B channels to be A/D-converted are selected using the ADANSB0 and ADANSB1 registers. Group A, B cannot use the same channels.

In group scan mode, deselect the temperature sensor output A/D conversion and the internal reference voltage A/D conversion by setting the ADEXICR.TSSA and ADEXICR.OCSA bits to 0.

Self-diagnosis cannot be selected when double trigger mode is selected in group scan mode.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated in the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1.

The following sequence describes operation in group scan mode with double trigger mode selected and using a synchronous trigger from the ELC or GPT. In this example, the ELC_AD00 trigger is used to start conversion of group A and the ELC_AD01 trigger is used to start conversion of group B. In addition, ELC_AD00 and ELC_AD01 are selected for the GPT event in the associated ELC.ELSRn registers.

The operation is as follows:

1. Scanning of group B is started by the ELC_AD00 trigger from the ELC.
2. When group B scanning completes, an ADC120_GBADI interrupt is generated if the GBADIE bit in ADCSR is 1 (ADC120_GBADI interrupt when scanning completion is enabled).
3. The first scan of group A is started by the first ELC_AD01 trigger.

4. When the first scan of group A completes, the conversion result is stored in the associated A/D Data Register y (ADDRy); an ADC120_ADI interrupt request is not generated.
5. The second scan of group A is started by the second ELC_AD01 trigger.
6. When the second scan of group A completes, the conversion result is stored in ADDBLDR. An ADC120_ADI interrupt is generated.

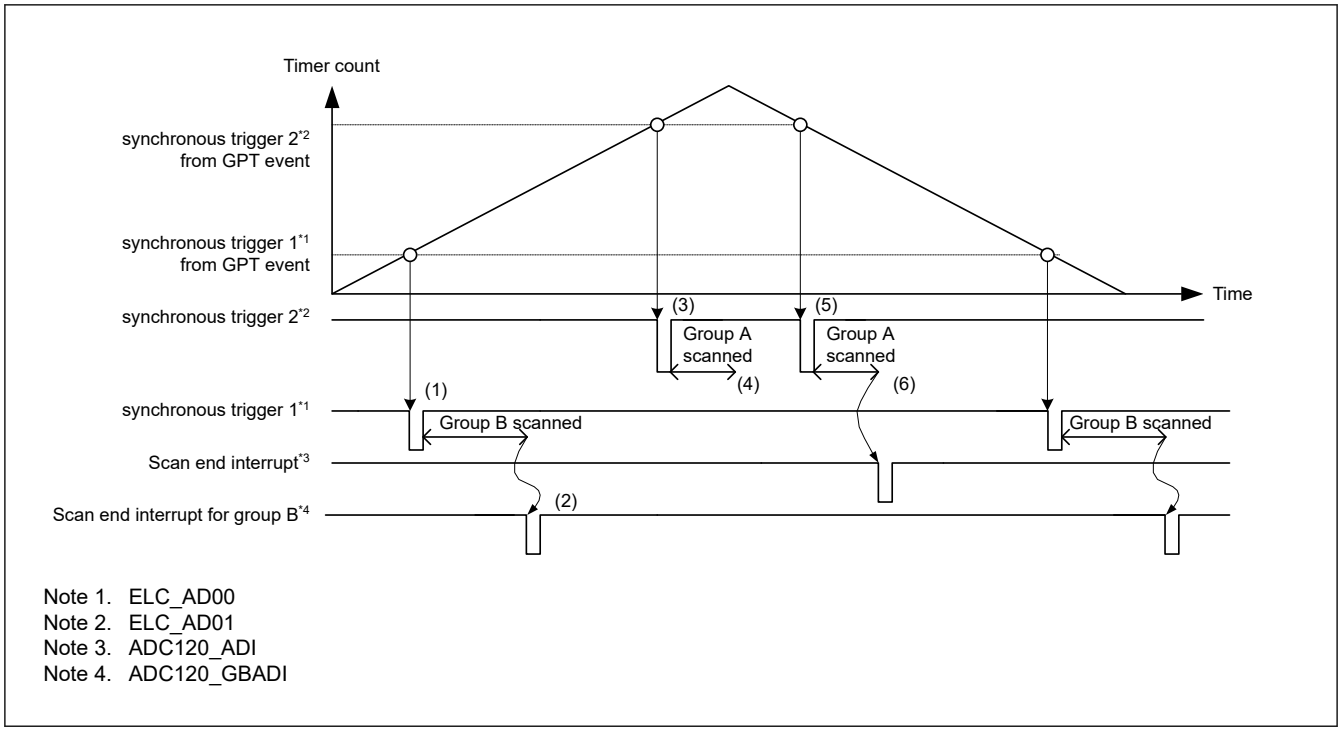


Figure 29.19 Example basic operation in group scan mode with double-trigger mode when synchronous triggers from the ELC are used

29.3.4.3 Group Priority Operation

Group priority operation is performed by setting the ADGSPCR.PGS bit to 1 in group-scan mode. The priority of groups is group A > group B.

When setting the PGS bit in the ADGSPCR register to 1, follow the procedure described in Figure 29.20. If the procedure is not followed, A/D conversion operation and stored data are not guaranteed.

As the basic operation in group-scan mode, a trigger input generated during A/D conversion of group A, B is ignored, and the A/D conversion operation of each group is similar to the operation in single-scan mode.

In group priority operation, if a trigger for a priority group is input during scanning of a lower-priority group, A/D conversion for the lower-priority group is stopped and A/D conversion for the priority group is performed.

If the setting of the ADGSPCR.GBRSCN bit is 0, the lower-priority group enters a wait state when A/D conversion for the priority group completes. A trigger input of the lower-priority group generated during A/D conversion is ignored.

If the setting of the ADGSPCR.GBRSCN bit is 1, A/D conversion for the lower-priority group automatically restarts upon completion of A/D conversion for the priority group. A trigger input of the lower-priority group generated during A/D conversion on the priority group takes effect, and A/D conversion for the lower-priority group is automatically performed upon completion of A/D conversion on the priority group.

Table 29.22 summarizes operations in response to the input of a trigger during A/D conversion with the settings of the ADGSPCR.GBRSCN bit.

If the setting of the ADGSPCR.GBRP bit is 1, A/D conversion operation for the lowest-priority group is to continuously perform single scans.

For the trigger settings in group-scan mode, select a synchronous trigger for group A by using the ADSTRGR.TRSA[5:0] bits, a synchronous trigger for group B by using the ADSTRGR.TRSB[5:0] bits. Each trigger must be different from each other. Set the ADSTRGR.TRSB[5:0] bits to 0x3F when setting the ADGSPCR.GBRP bit to 1.

The channels to be scanned must be selected in the registers shown in section 29.3.4. Group Scan Mode.

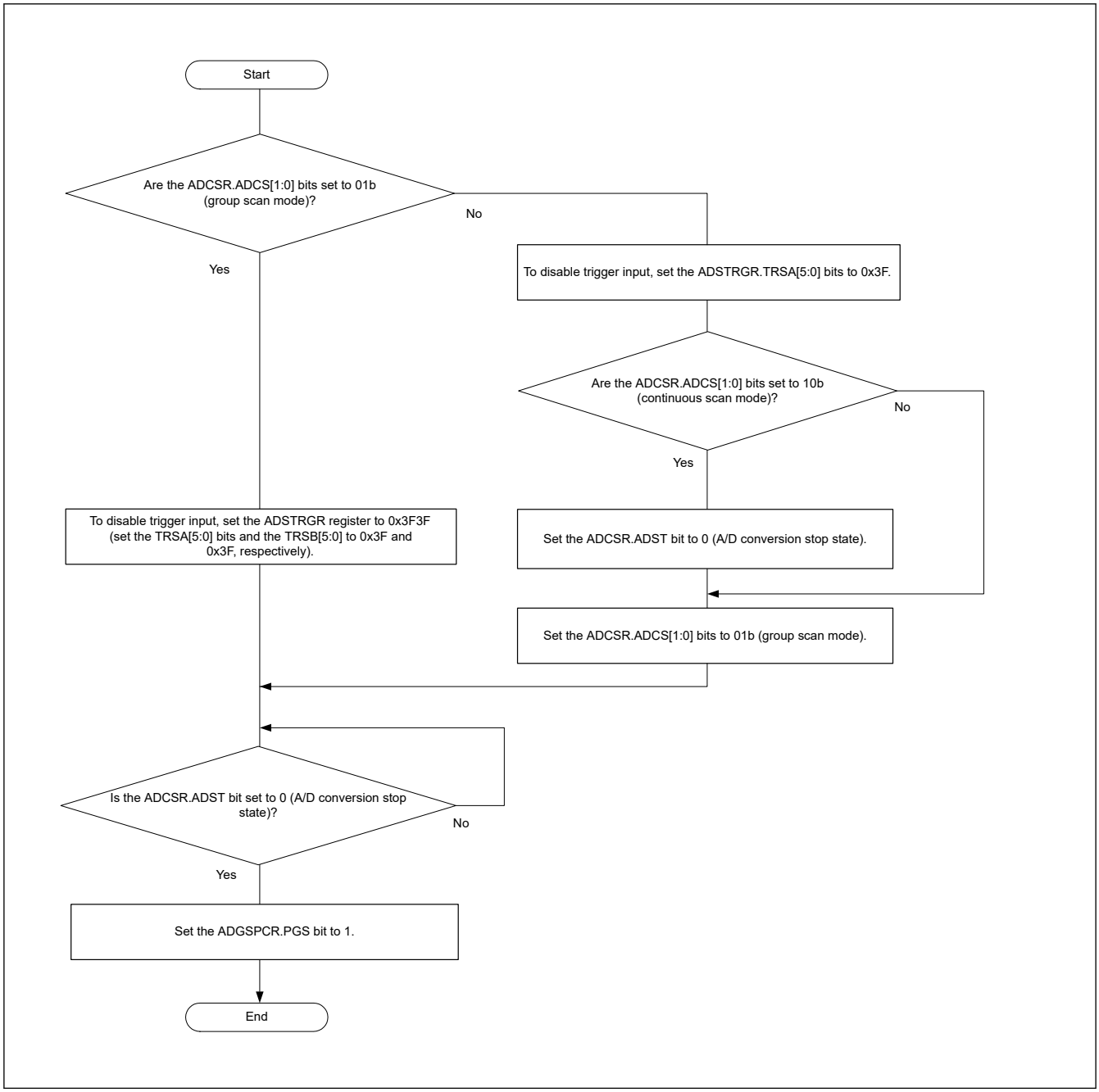


Figure 29.20 Flowchart for ADGSPCR.PGS bit setting

Table 29.22 Control of A/D conversion operations according to ADGSPCR.GBRSCN bit setting (1 of 2)

A/D conversion operation	Trigger input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group A is in progress	Input of trigger for group A	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group B	Trigger input is ineffective.	A/D conversion for group B is performed after A/D conversion for group A completes.

Table 29.22 Control of A/D conversion operations according to ADGSPCR.GBRSCN bit setting (2 of 2)

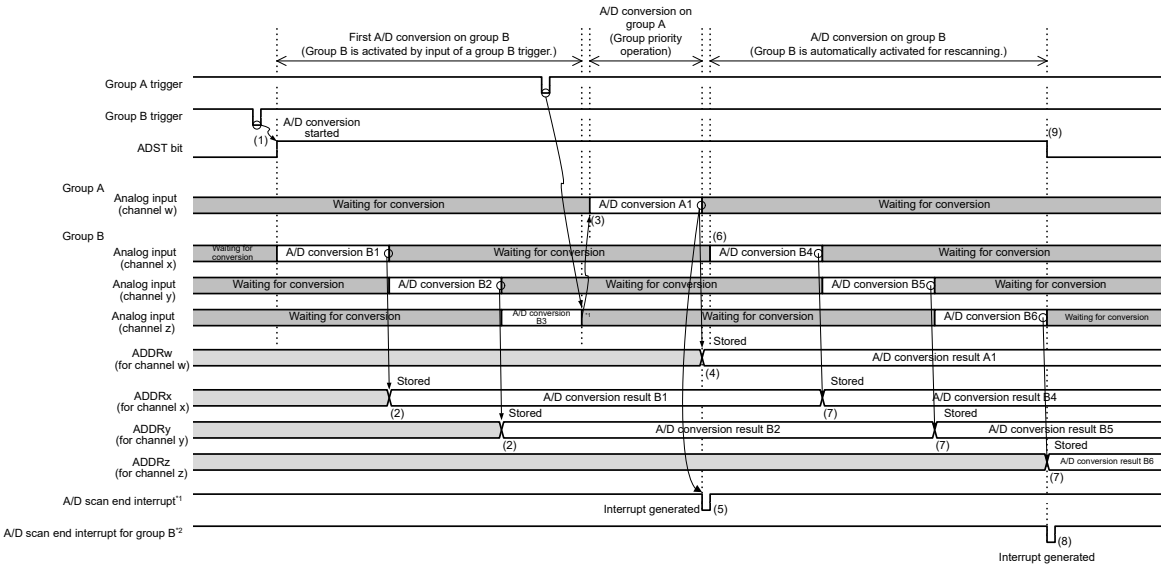
A/D conversion operation	Trigger input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group B is in progress	Input of trigger for group A	A/D conversion for group B is discontinued and A/D conversion for group A starts.	• A/D conversion for group B is discontinued and A/D conversion for group A starts. • A/D conversion for group B starts after A/D conversion for group A completes.
	Input of trigger for group B	Trigger input is ineffective.	Trigger input is ineffective.

(1) Group priority operation for two groups (when ADGSPCR.PGS = 1)

Operation examples 1-1 to 1-3 show group priority operations in group-scan mode (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0).

Operation example 1-1: “Group A trigger input during group B scan” when rescanning is enabled

1. When input of a trigger for group B sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order from the channel with the smallest number n.
2. On completion of A/D conversion for each channel in group B, the result is stored in the corresponding A/D Data Register y (ADDRy).
3. When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1. Then A/D conversion for the group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order from the channel with the smallest number n. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).
4. On completion of A/D conversion on the channels, the result is stored in the corresponding A/D Data Register y (ADDRy).
5. An ADC120_ADI interrupt request is generated.
6. If the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning of the group that was stopped in group priority operation), A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order from the channel with the smallest number n while the ADCSR.ADST remains 1.
7. On completion of A/D conversion on the channels, the result is stored in the corresponding A/D Data Register y (ADDRy).
8. If the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan), a group B scan end interrupt request is generated.
9. When A/D conversion for all the channels completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state.



Note: w, x, y, and z (x < y < z) are the selected channel numbers by ADANSA0/1 or ADANSB0/1 .
Note 1. ADC120_ADI
Note 2. ADC120_GBADI

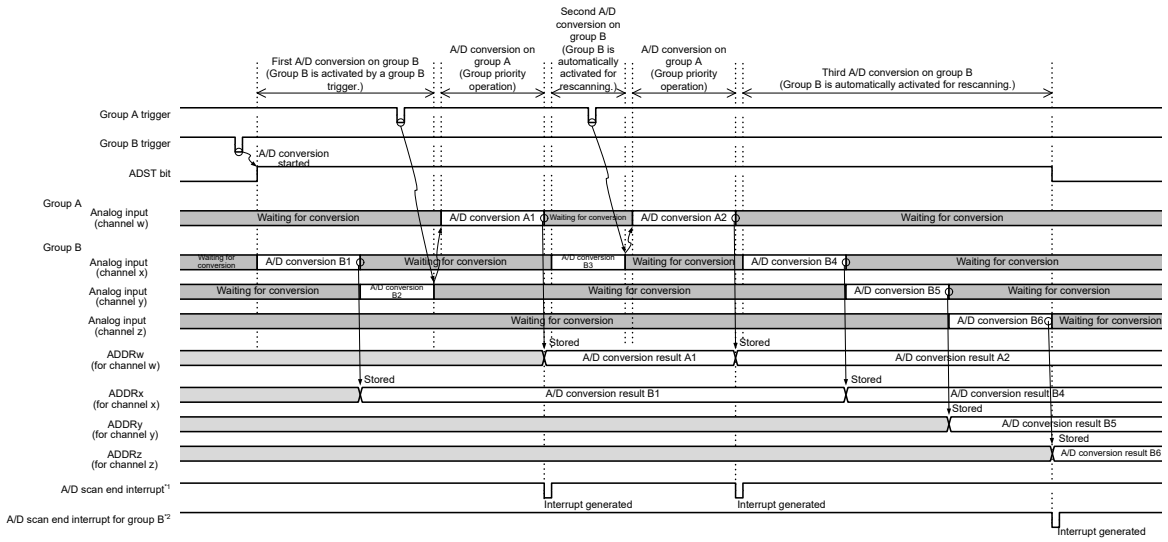
Figure 29.21 Example of group priority operation 1-1: Group A trigger input during group B scanning when rescanning is enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0)

Operation example 1-2: “Group A trigger input during rescanning of group B” when rescanning is enabled

Figure 29.22 shows the operation when a group A trigger is input during rescanning operation for group B.

Even during rescanning operation, when a trigger for group A is input, A/D conversion on group B stops and A/D conversion for group A starts. A/D conversion for group B starts after A/D conversion for group A completes.

Operations for setting the ADCSR.ADST bit, storing the A/D conversion result in the corresponding A/D Data Register y (ADDRy), and generating interrupt requests are the same as those in operation example 1-1.



Note: w, x, y, and z (x < y < z) are the selected channel numbers by ADANSA0/1 or ADANSB0/1.
Note 1. ADC120_ADI
Note 2. ADC120_GBADI

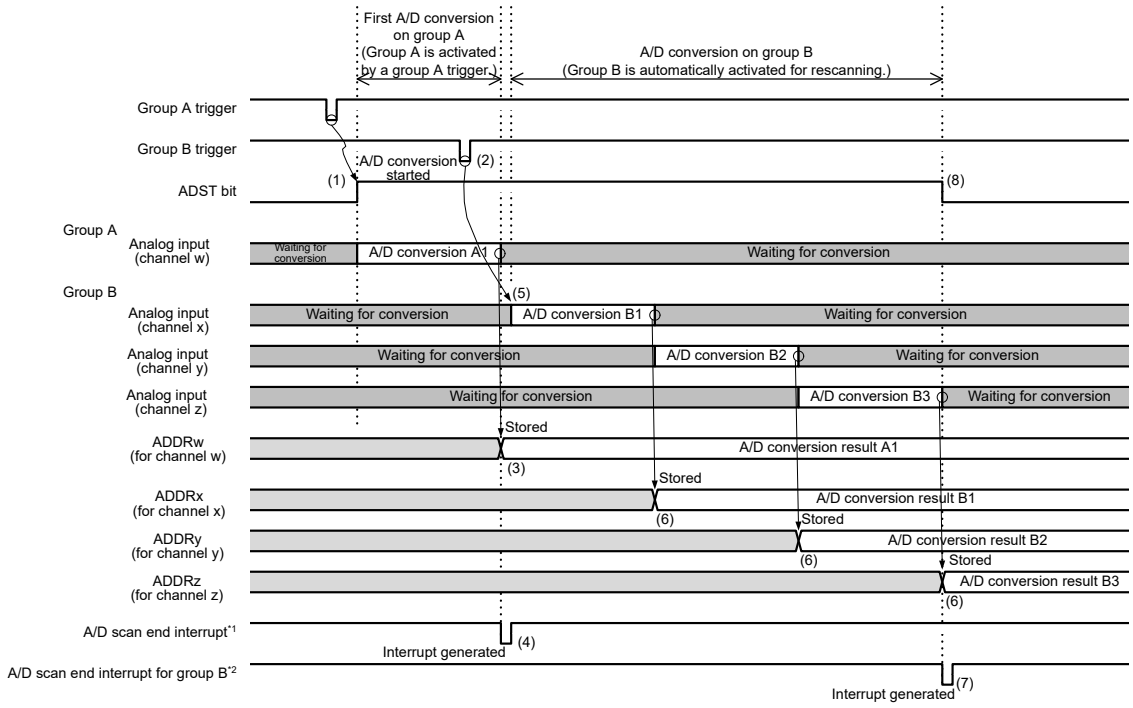
Figure 29.22 Example of group priority operation 1-2: Group A trigger input during rescanning of group B when rescanning is enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0)

Operation example 1-3: “Group B trigger input during group A scan” when rescanning is enabled

The following describes the operation when the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning of the group that was stopped in group priority operation) and a trigger for group B is input during scanning operation for group A.

If the setting of the ADGSPCR.GBRSCN bit is 0, any trigger for group B that is input during scanning operation for group A is invalid.

1. When input of a trigger for group A sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order from the channel with the smallest number n.
2. When a trigger for group B is input during A/D conversion for group A, group B is ready for A/D conversion.
3. On completion of A/D conversion for each channel in group A, the result is stored in the corresponding A/D Data Register y (ADDRy).
4. An ADC120_ADI interrupt request is generated.
5. When A/D conversion for group A completes, while the ADCSR.ADST bit remains 1, A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order from the channel with the smallest number n.
   (As with the case of operation example 1-1, if a trigger for group A is input during A/D conversion for group B, A/D conversion for group A starts. Then A/D conversion for group B starts upon completion of A/D conversion for group A.)
6. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
7. Upon completion of A/D conversion for group B, a group B scan end interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan).
8. When A/D conversion for all the channels completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state.



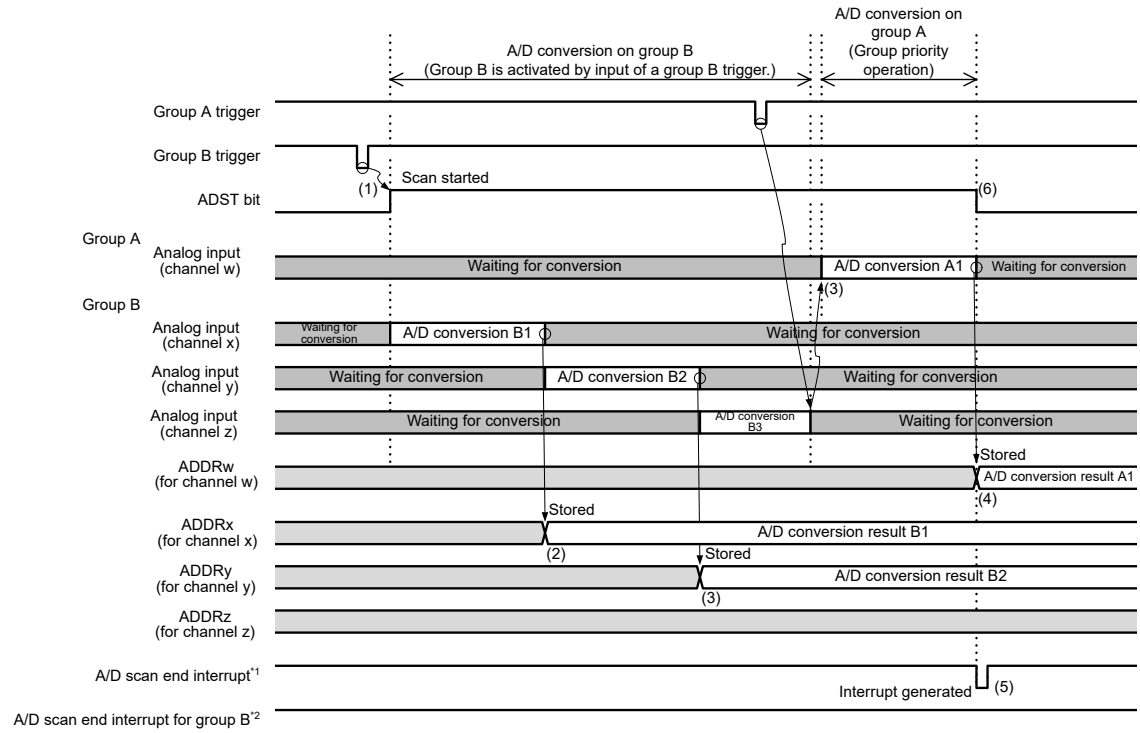
Note: w, x, y, and z (x < y < z) are the selected channel numbers by ADANSA0/1 or ADANSB0/1.
Note 1. ADC120_ADI
Note 2. ADC120_GBADI

Figure 29.23 Example of group priority operation 1-3: Group B trigger input during group A scan when rescanning is enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0)

Operation example 1-4 shows the group priority operation in group-scan mode (when ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0).

Operation example 1-4: “Group A trigger input during group B scan” when rescanning is disabled

1. When input of a trigger for group B sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order from the channel with the smallest number n.
2. On completion of A/D conversion for each channel in group B, the result is stored in the corresponding A/D Data Register y (ADDRy).
3. When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1, and then A/D conversion for the group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order from the channel with the smallest number n. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).
4. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
5. On completion of A/D conversion for group A, an ADC120_ADI interrupt request is generated.
6. When A/D conversion for group A completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state. A/D conversion for group B is not performed until a trigger for group B is input the next time.



Note: w, x, y, and z (x < y < z) are the selected channel numbers by ADANSA0/1 or ADANSB0/1.
Note 1. ADC120_ADI
Note 2. ADC120_GBADI

Figure 29.24 Group priority operation example 1-4: “Group A trigger is input during group B scan” when rescanning is disabled (when ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0)

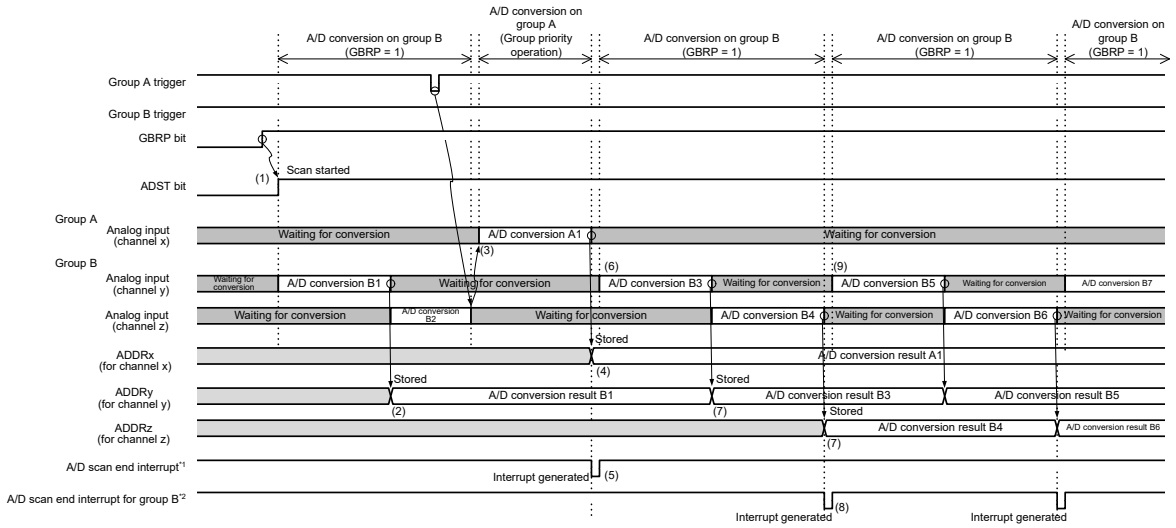
Operation example 1-5 shows the group priority operation in group-scan mode (when ADGSPCR.GBRP = 1).

Operation example 1-5: Continuously activating single-scan operation for group B

1. When ADGSPCR.GBRP = 1 is set, the ADCSR.ADST bit is set to 1 (starting A/D conversion) and A/D conversion for the analog input channels selected in the ADANSB0 and ADANSB1 registers starts according to the conversion order from the channel with the smallest number n.
2. On completion of A/D conversion for each channel in group B, the result is stored in the corresponding A/D Data Register y (ADDRy).
3. When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1, and then A/D conversion for group A analog input channels selected in the ADANSA0 and ADANSA1 registers starts according to the conversion order from the channel with the smallest number n. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).
4. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
5. On completion of A/D conversion for group A, an ADC120_ADI interrupt request is generated.
6. If ADGSPCR.GBRP = 1 is set (performing single scan continuously), A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order from the channel with the smallest number n while the ADCSR.ADST remains 1 (starting A/D conversion).
7. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
8. If the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan), a group B scan end interrupt request is generated.

9. If ADGSPCR.GBRP = 1 is set (performing single scan continuously), A/D conversion for the group B analog input channels selected in the ADANSB0 and ADANSB1 registers restarts according to the conversion order from the channel with the smallest number n while the ADCSR.ADST remains 1 (starting A/D conversion).

Steps 6 to 9 are repeated as long as the ADGSPCR.GBRP bit remains 1. Do not clear the ADCSR.ADST bit as long as the ADGSPCR.GBRP bit is 1. To forcibly stop A/D conversion while ADGSPCR.GBRP = 1, follow the procedure shown in Figure 29.36.



Note: x, y, and z (y < z) are the selected channel numbers by ADANSA0/1 or ADANSB0/1.
Note 1. ADC120_ADI
Note 2. ADC120_GBADI

Figure 29.25 Group priority operation example 1-5: Continuously activating single scan for group B (when ADGSPCR.GBRP = 1)

Note: To continuously activate single-scan operation for group B, disable group B trigger input.

29.3.5 Compare Function for Windows A and B

29.3.5.1 Compare Function Windows A and B

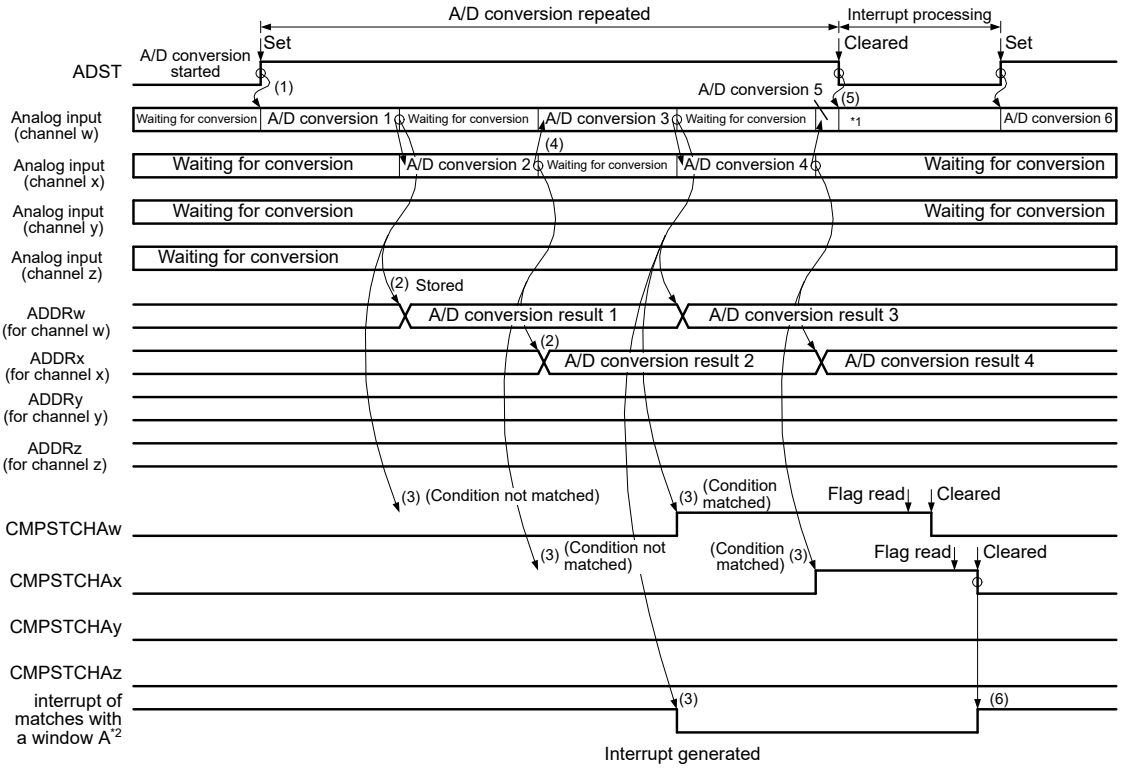
The compare function compares a reference value with the A/D conversion result. The reference value can be set for Window A and Window B independently. When the compare function is in use, the self-diagnosis function and double trigger mode cannot be used. The main differences between Window A and Window B are their different interrupt output signals and the constraint on Window B of only one selectable channel.

This section provides an example operation that combines continuous scan mode and the compare function.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, a synchronous trigger (ELC, GPT) or an asynchronous trigger, A/D conversion starts of the selected channel. Both temperature sensor and internal reference voltage are not selectable at the same time. Additionally, when the internal reference voltage is selected as the high-potential reference voltage, A/D conversion of the temperature sensor or internal reference voltage is prohibited.
2. On completion of A/D conversion, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy, ADTSDR, or ADOCDR). When ADCMPCR.CMPAE = 1, if bits in the ADCMPANSRy register or the ADCMPANSER register are set for Window A, the A/D conversion result is compared with the set ADCMPDR0/1 register value. When ADCMPCR.CMPBE = 1, if bits in the ADCMPBNSR register are set for Window B, the A/D conversion result is compared with the ADWINULB/ADWINLLB register setting.

3. As a result of the comparison, when Window A meets the condition set in ADCMPLR0/1 or ADCMPLER, the Compare Window A Flag (ADCMPSR0.CMPSTCHAn, ADCMPSR1.CMPSTCHAn, ADCMPSER.CMPSTTSA or ADCMPSER.CMPSTOCA) sets 1. At this time, if the ADCMPCR.CMPAIE bit is 1, an ADC120_CMPAI interrupt request is generated. In the same way, when Window B meets the condition set in ADCMPBNSR.CMPLB, the Compare Window B Flag (ADCMPBSR.CMPSTB) sets to 1. At this time, if the ADCMPCR.CMPBIE bit is 1, an ADC120_CMPBI interrupt request is generated.
4. On completion of all selected A/D conversions and comparisons, scan restarts.
5. After the ADC120_CMPAI and ADC120_CMPBI interrupts are accepted, the ADCSR.ADST bit is set to 0 (A/D conversion stop) and processing is performed for channels for which the compare flag is set to 1.
6. When all compare flags of Window A are cleared, the ADC120_CMPAI interrupt request is canceled. In the same way, when all compare flags of Window B are cleared, the ADC120_CMPBI interrupt request is reset. To perform comparison again, restart the A/D conversion.



Note: w, x, y, and z (w < x < y < z) are the selected channel numbers by ADANSA0/1.
Note 1. Data on conversion is ignored.
Note 2. ADC120_CMPAI

Figure 29.26 Example of compare function operation, when the analog inputs (channel w to z) are compared

29.3.5.2 Event Output of Compare Function

The event output of the compare function specifies the upper-side reference voltage value and the lower-side reference voltage value for window A and window B, respectively. The output compares the A/D converted value of the selected channel with the upper and lower side reference voltage value and outputs events (ADC120_WCMPM/ ADC120_WCMPUM) based on event conditions (A or B, A and B, A xor B) and comparison result of window A and window B.

If more than one channel is selected for window A, and even when one channel in window A meets the comparison condition, the comparison result of window A is met. When using this function, perform A/D conversion in single scan mode.

Any channels from analog input, internal reference voltage, and temperature sensor output are selectable for window A. However, neither the internal reference voltage nor the temperature sensor output can be selected together with any other channel. Additionally, if the internal reference voltage is selected as the high-potential reference voltage of the A/D converter, the internal reference voltage or the temperature sensor output cannot be A/D converted.

One channel from analog input, internal reference voltage, and temperature sensor output is selectable for window B. Additionally, if the internal reference voltage is selected as the high-potential reference voltage, the internal reference voltage or the temperature sensor output cannot be A/D converted.

The following sequence is an example of how to set up and use the event output of the compare function:

1. Confirm that the value in the ADCSR.ADCS bits is 00b (single scan mode).
2. Select the channel for window A in the ADCMPANSR0/1 and ADCMPANSER registers. Set the window comparison conditions in the ADCMPLR0/1 and ADCMPLER registers. Set the upper-side and lower-side reference values in the ADCMPDR0/1 registers.
3. Select the channel and comparison conditions for Window B in the ADCMPBNSR register, and set the upper and lower reference values in the ADWINULB and ADWINLLB registers.
4. Set the composite conditions for window A/B, window A/B operation enable, and interrupt output enable in the ADCMPCR register.

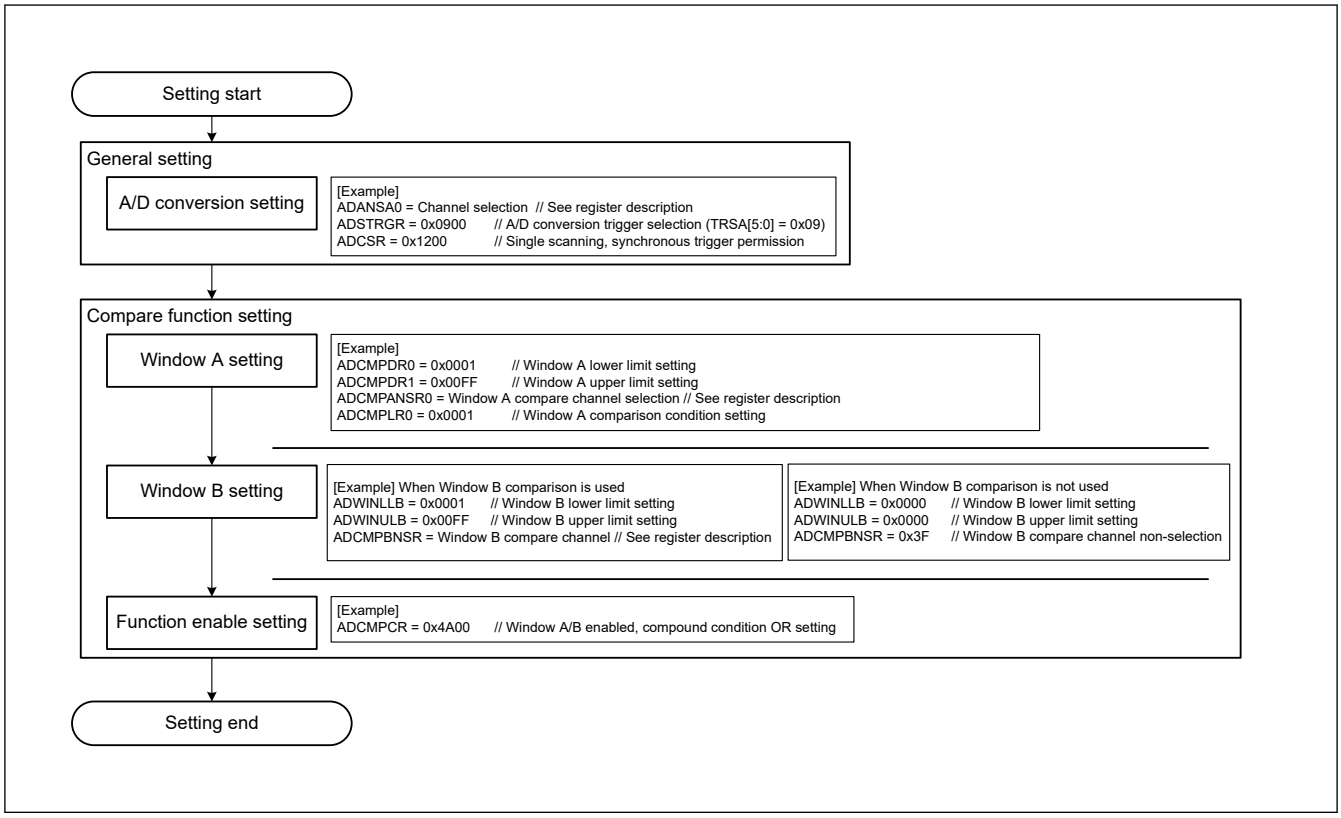


Figure 29.27 Setting example when using the event output of the compare function

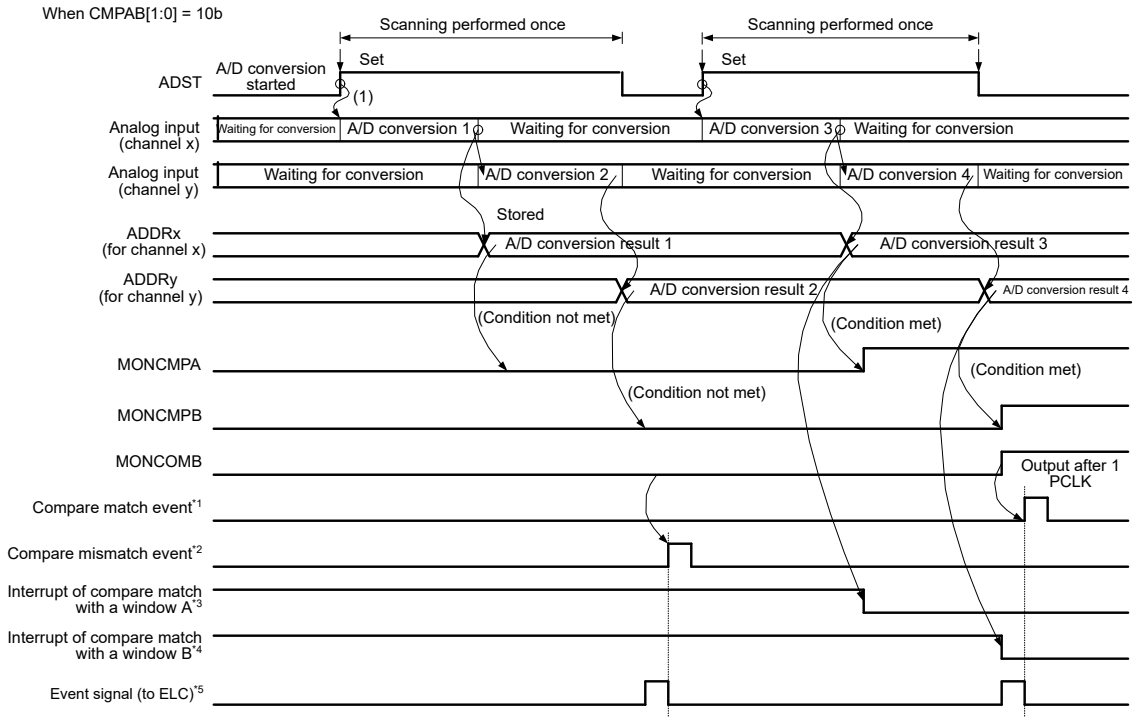
For event output usage when using only window A for the compare function, note the following:

- Enable both Window A and Window B (ADCMPCR.CMPAE = 1, ADCMPCR.CMPBE = 1)
- Set the compound condition of Window A and Window B to “OR condition” (ADCMPCR.CMPAB[1:0] = 00b)
- Set the compared channel of Window B to “No selection” (ADCMPBNSR.CMPCHB[5:0] = 0x3F)
- Set the compare condition of Window B to “0 < results < 0 always means mismatch”. (ADCMPCR.WCMPE = 1, ADWINLLB[15:0] = ADWINULB[15:0] = 0x0000, and ADCMPBNSR.CMPLB = 1)

Figure 29.28 shows the event output operation example of compare function.

A scan end event (ADC120_ADI) is output with the same timing as single scan completion. A match or mismatch event (ADC120_WCMPM/ADC120_WCMPUM) is output with 1 PCLKB cycle delay depending on the ADCMPCR.CMPAB[1:0] settings.

Note: The match and mismatch events are exclusive, so both events are never output simultaneously.



Note: x, and y (x < y) are the selected channel numbers by ADANSA0/1.
Note 1. ADC120_WCMPM
Note 2. ADC120_WCMPUM
Note 3. ADC120_CMPAI
Note 4. ADC120_CMPBI
Note 5. ADC120_ADI

Figure 29.28 Example operation of the compare function event output, when the analog inputs (channel x and y) are compared

Note: Event output of compare function outputs match/mismatch from the comparison results of Window A and Window B, based on the ADCMPCR.CMPAB[1:0] settings.

Note: The comparison result of Window A is the logical addition of the comparison results of the comparison target channels of Window A. The comparison results of Window A and Window B are updated by each A/D conversion, and are kept even when single scan ends. Set ADCMPCR.CMPAE and ADCMPCR.CMPBE to 0 to clear the comparison results to 0.

29.3.5.3 Restrictions on Compare Function

The following constraints apply for the compare function:

- The compare function cannot be used together with the self-diagnosis function or double-trigger mode. (The compare function is not available for ADRD, ADDBLDR, ADDBLDRA, and ADDBLDRB.)
- Specify single scan mode when using match/mismatch event outputs.
- When the temperature sensor output, internal reference voltage is selected for Window A, Window B operations are prohibited.

- When the temperature sensor output, internal reference voltage is selected for Window B, Window A operations are prohibited.
- Setting the same channel for Window A and Window B is prohibited.
- Set the reference voltage values so that the high-potential reference voltage value is equal to or larger than the low potential reference voltage value.

29.3.6 Analog Input Sampling and Scan Conversion Time

Scan conversion can be activated either by a software trigger, a synchronous trigger (ELC, GPT), or an asynchronous trigger (ADTRG0). After the start-of-scanning-delay time (t_D) has elapsed, processing by the channel-dedicated sample-and-hold circuits, processing for disconnection detection assistance, and processing of conversion for self-diagnosis all proceed, followed by processing for A/D conversion.

Figure 29.29 shows the scan conversion timing, in which scan conversion is activated by a software trigger or a synchronous trigger (ELC, GPT). Figure 29.30 shows the scan conversion timing, in which scan conversion is activated by an asynchronous trigger (ADTRG0). The scan conversion time (t_{SCAN}) includes the start-of-scanning-delay time (t_D), channel-dedicated sample-and-hold circuit processing time (t_{SPLSH})*1, disconnection detection assistance processing time (t_{DIS})*2, self-diagnosis A/D conversion processing time (t_{DIAG} and t_{DSD})*3, A/D conversion processing time (t_{CONV}), channel-dedicated sample-and-hold circuit end time (t_{SHED})*4, and end-of-scanning-delay time (t_{ED}).

The A/D conversion processing time (t_{CONV}) consists of input sampling time (t_{SPL}) and time for conversion by successive approximation (t_{SAM}). The sampling time (t_{SPL}) is used to charge sample-and-hold circuits in the A/D converter. If there is not sufficient sampling time due to the high impedance of an analog input signal source, or if the A/ D conversion clock (ADCLK) is slow, sampling time can be adjusted using the ADSSTRn register.

The time for conversion by successive approximation (t_{SAM}) is the following

- 31.5 ADCLK states with 12-bit accuracy and high-speed A/D conversion mode (ADCSR.ADHSC = 0) and normal conversion mode (ADACSR.ADSAC = 0) selected.
- 40.5 ADCLK states with 12-bit accuracy and Low-power A/D conversion mode (ADCSR.ADHSC = 1) and normal conversion mode (ADACSR.ADSAC = 0) selected.
- 21.5 ADCLK states with 12-bit accuracy and high-speed A/D conversion mode (ADCSR.ADHSC = 0) and fast conversion mode (ADACSR.ADSAC = 1) selected.
- 27.5 ADCLK states with 12-bit accuracy and Low-power A/D conversion mode (ADCSR.ADHSC = 1) and fast conversion mode (ADACSR.ADSAC = 1) selected.

Table 29.23 shows the time for conversion by successive approximation (t_{SAM}).

The scan conversion time (t_{SCAN}) in single scan mode for which the number of selected channels is n can be determined as follows:

$$t_{SCAN} = t_D + t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + (t_{CONV} \times n) + t_{ED}$$ *5

The scan conversion time for the first cycle in continuous scan mode is t_{SCAN} for single scan minus t_{ED} plus t_{SHED}. The scan conversion time for the second and subsequent cycles in continuous scan mode is fixed in the following:

$$t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + t_{DSD} + (t_{CONV} \times n) + t_{SHED}$$ *5

Note 1. When no channel-dedicated sample-and-hold circuits are used, t_{SPLSH} = 0.

Note 2. When disconnection detection assistance is not selected, t_{DIS} = 0.

Only when the temperature sensor or internal reference voltage is A/D-converted, the auto-discharge period of 15 ADCLK states is inserted.

Note 3. When the self-diagnosis function is not used, t_{DIAG} = 0, t_{DSD} = 0.

Note 4. When no channel-dedicated sample-and-hold circuits are used, t_{SHED} = 0, assuming continuous scan mode is active. In single scan mode and group scan mode, t_{SHED} is included in the end-of-scanning-delay (t_{ED}).

Note 5. (t_{CONV} × n) : When input sampling times (t_{SPL}) of all selected channels are the same, this element equals t_{CONV} × n. If each channel has a different sampling time, this element equals that of t_{SPL} and t_{SAM} set to each selected channel.

Table 29.23 shows the times for conversion during scanning.

Table 29.23 Conversion times during scanning (in numbers of cycles of ADCLK and PCLKB)

Item			Symbol		Type/Conditions: Synchronous trigger*5	Type/Conditions: Asynchronous trigger	Type/Conditions: Software trigger	Unit
Scan start delay time*1 *2	A/D conversion on group A under group A priority control.	Group B is to be stopped (Group A is activated after group B is stopped by of an A/D conversion source from group A).	t_D		3 PCLKB + 6 ADCLK 5 PCLKB + 3 ADCLK*6	—	—	Cycles
		Group B is not to be stopped (Activation by an A/D conversion source from group A).			2 PCLKB + 4 ADCLK	—	—	
	A/D conversion when self-diagnosis is enabled.	A/D conversion for self-diagnosis is to be started.			2 PCLKB + 4 ADCLK	4 PCLKB + 6 ADCLK	6 ADCLK	
	All other				2 PCLKB + 4 ADCLK	2 PCLKB + 4 ADCLK	4 ADCLK	
Channel-dedicated sample-and-hold processing time*1	Sampling time		t_{SPLSH}	t_{SH}	Without continuous sampling: setting in ADSHCR.SSTSH[7:0] (initial value = 26 × ADCLK) With continuous sampling:0			
	Wait time between sampling and A/D Conversion			t_W	12 × ADCLK			
Disconnection detection assistance processing time			t_{DIS}		Setting in ADNDIS[3:0] (initial value = 0x00) × ADCLK*3			
Self-diagnosis conversion processing time*1	Sampling time*4		t_{DIAG}	t_{SPL}	Setting in ADSSTRn (n = 0 to 2, 5 to 10, L, T, O) (initial value = 0x0D) × ADCLK + 0.5 ADCLK			
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	31.5 ADCLK at high-speed A/D conversion mode (ADCSR.ADHSC = 0) and normal conversion mode (ADACSR.ADSAC = 0) 40.5 ADCLK at Low-power A/D conversion mode (ADCSR.ADHSC = 1) and normal conversion mode (ADACSR.ADSAC = 0) 21.5 ADCLK at high-speed A/D conversion mode (ADCSR.ADHSC = 0) and fast conversion mode (ADACSR.ADSAC = 1) 27.5 ADCLK at Low-power A/D conversion mode (ADCSR.ADHSC = 1) and fast conversion mode (ADACSR.ADSAC = 1)			
	Wait time between self-diagnosis conversion end and analog channel sampling start.			t_{DED}	2 ADCLK			
	Wait time between last channel conversion end and self-diagnosis sampling start in continuous scan mode.			t_{DSD}	2 ADCLK			
A/D conversion processing time*1	Sampling time*4		t_{CONV}	t_{SPL}	Setting in ADSSTRn (n = 0 to 2, 5 to 10, L, T, O) (initial value = 0x0D) × ADCLK + 0.5 ADCLK			
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	31.5 ADCLK at high-speed A/D conversion mode (ADCSR.ADHSC = 0) and normal conversion mode (ADACSR.ADSAC = 0) 40.5 ADCLK at Low-power A/D conversion mode (ADCSR.ADHSC = 1) and normal conversion mode (ADACSR.ADSAC = 0) 21.5 ADCLK at high-speed A/D conversion mode (ADCSR.ADHSC = 0) and fast conversion mode (ADACSR.ADSAC = 1) 27.5 ADCLK at Low-power A/D conversion mode (ADCSR.ADHSC = 1) and fast conversion mode (ADACSR.ADSAC = 1)			
Channel-dedicated sample-and-hold end processing time			t_{SHED}		2 ADCLK			
Scan end processing time*1			t_{ED}		1 PCLKB + 3 ADCLK 2 PCLKB + 3 ADCLK*6			

Note 1. See Figure 29.29 and Figure 29.30 for an illustration of times t_D, t_{SPLSH}, t_{DIAG}, t_{CONV}, and t_{ED}.
Note 2. This is the maximum time required from software writing or trigger input to A/D conversion start.
Note 3. The value is fixed to 0x0F (15 ADCLK) when the temperature sensor output or internal reference voltage is A/D- converted.
Note 4. The sampling time setting should satisfy the electrical characteristics. For the details, see section 35.4. ADC12 Characteristics.
Note 5. This does not include the time consumed in the path from timer output to trigger input.
Note 6. If ADCLK is faster than PCLKB (PCLKB to ADCLK frequency ratio = 1:2 or 1:4).

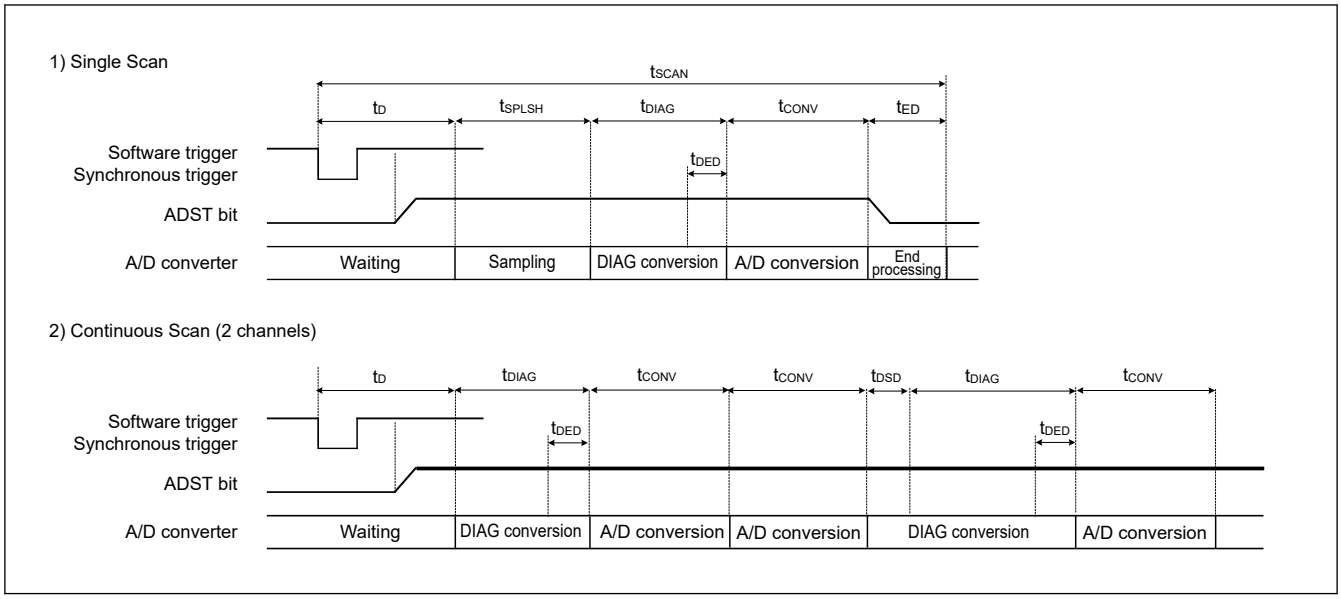


Figure 29.29 Scan conversion timing when activated by software or a synchronous trigger input (ELC)

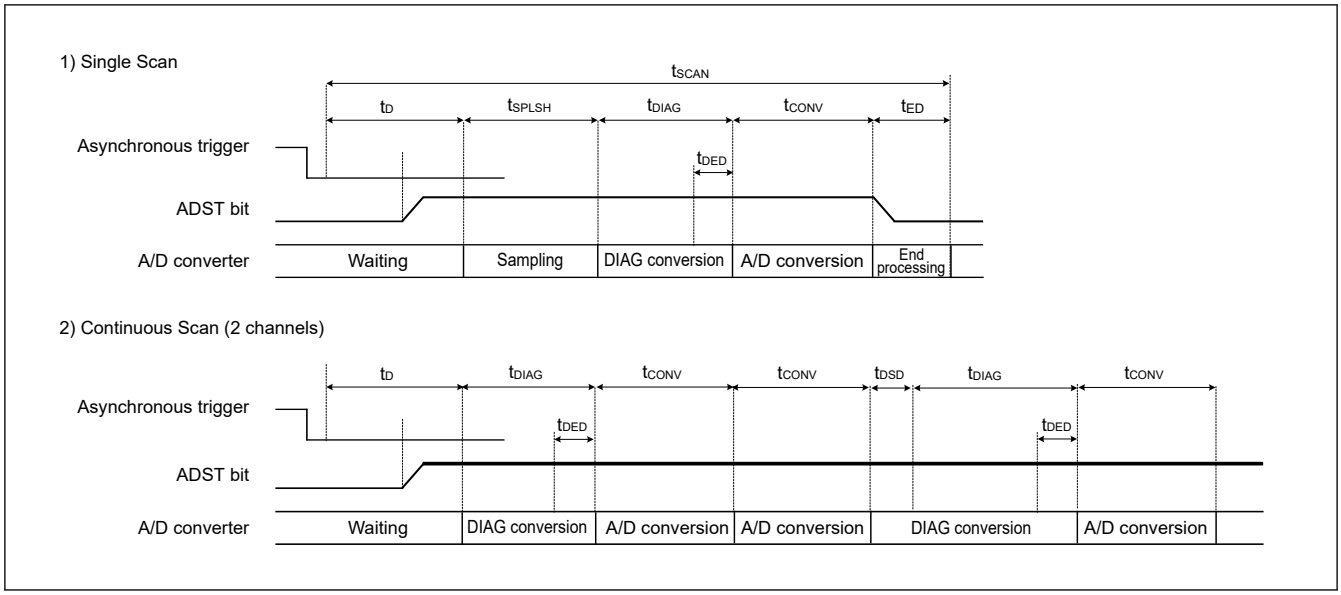


Figure 29.30 Scan conversion timing when activated by an asynchronous trigger input (ADTRG0)

29.3.7 Usage Example of A/D Data Register Automatic Clearing Function

Setting the ADCER.ACE bit to 1 automatically clears the A/D data registers (ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR) to 0x0000 when the A/D data registers are read by the CPU or DTC.

This function enables detection of update failures of the A/D data registers (ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR). This section describes examples in which the function to automatically clear the ADDRy register is enabled and disabled.

- If the ADCER.ACE bit is 0 (automatic clearing is disabled) and for some reason, if the A/D conversion result (0x0222) is not written to the ADDRy register, the ADDRy value retains the old data (0x0111). In addition, if this ADDRy value is read into a general-purpose register using an A/D scan end interrupt, the old data (0x0111) can be saved in the general-purpose register. When checking whether there is an update failure, it is necessary to frequently save the old data in SRAM or in a general-purpose register.
- If the ADCER.ACE bit is 1 (automatic clearing is enabled), when ADDRy = 0x0111 is read by the CPU or DTC, ADDRy is automatically set to 0x0000. Next, if the A/D conversion result of 0x0222 cannot be transferred to ADDRy for some reason, the cleared data (0x0000) remains as the ADDRy value. If this ADDRy value is read into a

general-purpose register using an A/D scan end interrupt, 0x0000 is saved in the general-purpose register. Occurrence of an ADDRy update failure can be determined by checking that the read data value is 0x0000.

29.3.8 A/D-Converted Value Addition/Average Mode

A/D-converted value addition/average mode can be used when A/D conversion of the analog input of the selected channels, the temperature sensor output, the internal reference voltage is selected.

In A/D-converted value addition mode, the same channel is A/D-converted 1, 2, 3, 4, or 16 consecutive times, and the sum of the converted values is stored in the data register. In A/D-converted value average mode, the same channel is A/D-converted 2 or 4 consecutive times, and the mean of the converted values is stored in the data register. The use of the average of these results can improve the accuracy of A/D conversion, depending on the types of noise components that are present. This function, however, cannot always guarantee an improvement in A/D conversion accuracy.

The A/D-converted value addition/average function can be used when A/D conversion of the analog inputs of the selected channels or A/D conversion of the temperature sensor output or A/D conversion of the internal reference voltage is selected. The A/D-converted value addition/average function can also be used for channels for which the double-trigger function is selected.

The addition function for self-diagnosis is not provided.

29.3.9 Disconnection Detection Assist Function

The ADC12 incorporates a function to fix the charge for sampling capacitance to the specified state VREFH0 or VREFL0 before the start of A/D conversion. This function enables disconnection detection in wiring of analog inputs.

Figure 29.31 shows the A/D conversion operation when the disconnection detection assist function is used. Figure 29.32 shows an example of disconnection detection when precharge is selected. Figure 29.33 shows an example of disconnection detection when discharge is selected.

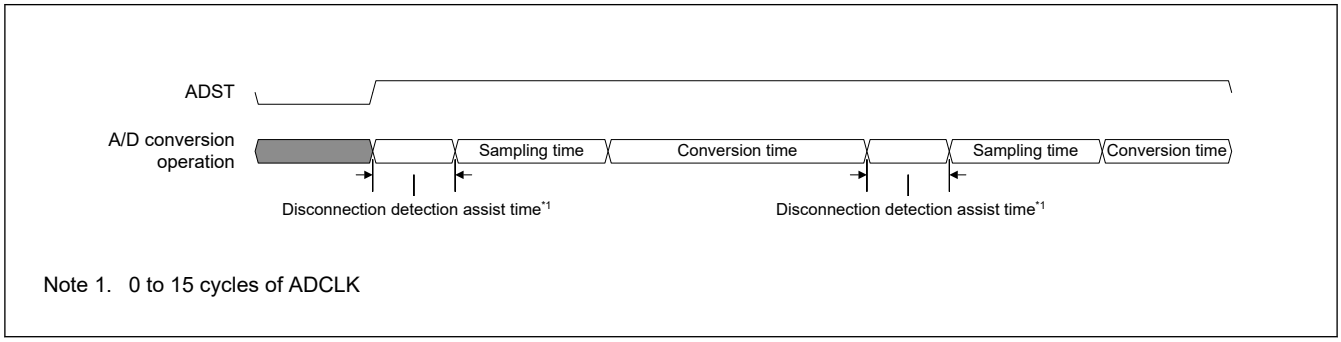
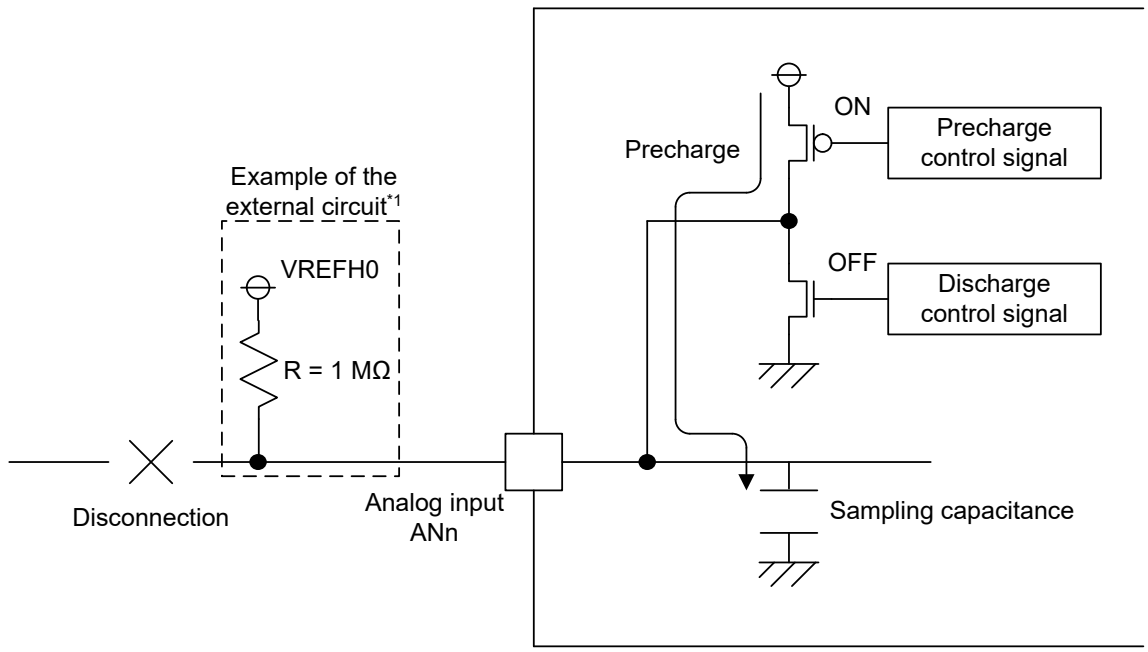
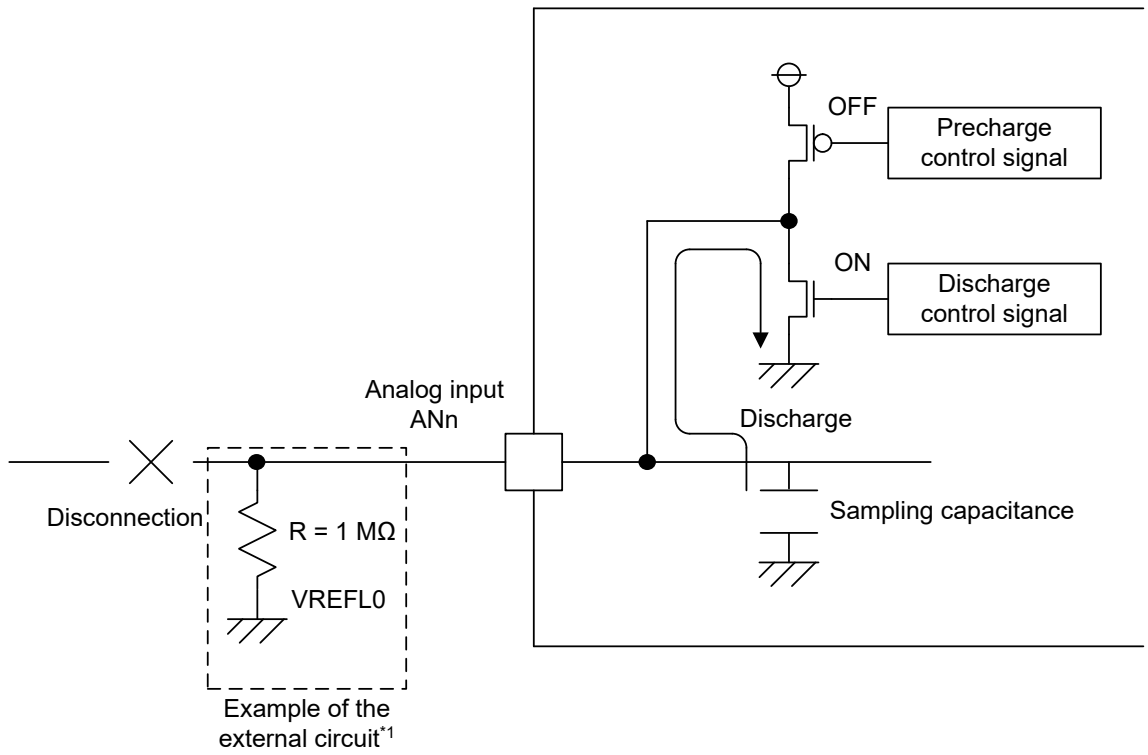


Figure 29.31 Operation of A/D conversion when disconnection detection assist function is used



Note 1. The converted result should be used after it is fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 29.32 Example of disconnection detection when precharge is selected



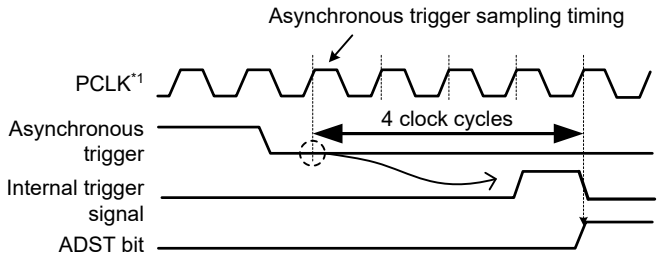
Note 1. The converted result should be used after it is fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 29.33 Example of disconnection detection when discharge is selected

29.3.10 Starting A/D Conversion with an Asynchronous Trigger

A/D conversion can be started by the input of an asynchronous trigger. To start A/D conversion by an asynchronous trigger, set the pin function in the PmnPFS register, set the A/D Conversion Start Trigger Select bits (ADSTRGR.TRSA[5:0]) to 0x00, then input a high-level signal to the asynchronous trigger (ADTRG0 pin). Finally, set both the ADCSR.TRGE and ADCSR.EXTRG bits to 1. Figure 29.34 shows timing of the asynchronous trigger input.

An asynchronous trigger cannot be selected in the A/D conversion start trigger for group B used in group scan mode. For details on setting the pin function, see section 17, I/O Ports.



Note 1. PCLKB

Figure 29.34 Asynchronous trigger input timing

29.3.11 Starting A/D Conversion with a Synchronous Trigger from a Peripheral Module

A/D conversion can be started by a synchronous trigger (ELC, GPT). To do this, set the ADCSR.TRGE bit to 1 and the ADCSR.EXTRG bit to 0, and select the relevant sources in the ADSTRGR.TRSA[5:0] bits and ADSTRGR.TRSB[5:0] bits.

29.4 Interrupt Sources and DTC Transfer Requests

29.4.1 Interrupt Requests

The ADC12 can send scan end interrupt requests ADC120_ADI and ADC120_GBADI to the CPU. The ADC12 also generates the ADC120_CMPAI/ADC120_CMPBI interrupt for the CPU in response to matches with a condition for comparison.

An ADC120_ADI interrupt is always generated. An ADC120_GBADI interrupt can be generated by setting the ADCSR.GBADIE bit to 1. Similarly, ADC120_CMPAI and ADC120_CMPBI interrupts can be generated by setting the ADCMPCR.CMPAIE and ADCMPCR.CMPBIE bit to 1.

In addition, the DTC can be started when an ADC120_ADI or an ADC120_GBADI interrupt is generated. Using an ADC120_ADI or ADC120_GBADI interrupt to activate the DTC to read the converted data enables continuous conversion without a burden on software.

Table 29.24 describes the interrupt sources and ELC events available for the ADC12.

Table 29.24 The interrupt source and ELC event of ADC12 (1 of 2)

Operation			Interrupt request or ELC event	Interrupt request	DTC activation	ELC event request	Function
Scan mode	Double trigger mode	Compare function Window A/B					
Single scan mode	Deselected	Deselected	ADC120_ADI	✓	✓	✓	ADC120_ADI generated at the end of single scan
		Selected	ADC120_ADI	✓	✓	✓	ADC120_ADI generated at the end of single scan
			ADC120_CMPAI	✓	—	—	ADC120_CMPAI generated on a match comparison condition of Window A
			ADC120_CMPBI	✓	—	—	ADC120_CMPBI generated on a match comparison condition of Window B
			ADC120_WCMPM	—	✓	✓	ADC120_WCMPM generated on a match condition of the Window A/B compare function
			ADC120_WCMPUM	—	✓	✓	ADC120_WCMPUM generated on a mismatch condition of the Window A/B compare function
	Selected	Deselected	ADC120_ADI	✓	✓	✓	ADC120_ADI generated at the end of scans in the even numbered times
Continuous scan mode	Deselected	Deselected	ADC120_ADI	✓	✓	✓	ADC120_ADI generated at the end of scan of all selected channels
		Selected	ADC120_CMPAI	✓	—	—	ADC120_CMPAI generated on a match comparison condition of Window A
			ADC120_CMPBI	✓	—	—	ADC120_CMPBI generated on a match comparison condition of Window B

Table 29.24 The interrupt source and ELC event of ADC12 (2 of 2)

Operation			Interrupt request or ELC event	Interrupt request	DTC activation	ELC event request	Function
Scan mode	Double trigger mode	Compare function Window A/B					
Group scan mode	Deselected	Deselected	ADC120_ADI	✓	✓	✓	ADC120_ADI generated at the end of group A scan
			ADC120_GBADI	✓	✓	—	ADC120_GBADI dedicated to group B generated at the end of group B scan
		Selected	ADC120_ADI	✓	✓	✓	ADC120_ADI generated at the end of group A scan
			ADC120_GBADI	✓	✓	—	ADC120_GBADI dedicated to group B generated at the end of group B scan
			ADC120_CMPAI	✓	—	—	ADC120_CMPAI generated on a match comparison condition of Window A
			ADC120_CMPBI	✓	—	—	ADC120_CMPBI generated on a match comparison condition of Window B
	Selected	Deselected	ADC120_ADI	✓	✓	✓	ADC120_ADI generated at the end of group A scans in the even-numbered times
			ADC120_GBADI	✓	✓	—	ADC120_GBADI dedicated to group B generated at the end of group B scan

Note: ✓ available
—: unavailable

For details on DTC settings, see section 15, Data Transfer Controller (DTC).

29.5 Event Link Function

29.5.1 Event Output to the ELC

The ELC uses the ADC120_ADI interrupt request signal as an event signal ADC120_ADI, enabling link operation for the preset module. The ADC120_GBADI interrupt and ADC120_CMPAI/ADC120_CMPBI interrupts cannot be used as an event signal. For details, see Table 29.24.

An event signal can be output regardless of the settings of the corresponding interrupt request enable bits. For the scan end event(ADC120_ADI), a high-level pulse for one PCLKB cycle is output at the same output timing as the interrupt output (ADC120_ADI) shown in Table 29.24. For a compare function match (ADC120_WCMPM) and mismatch event (ADC120_WCMPUM) to the ELC, a high-level pulse for one PCLKB cycle is output at the timing delayed by one cycle (PCLKB) from the interrupt output (ADC120_ADI) shown in Table 29.24.

To use compare function match (ADC120_WCMPM) or mismatch event (ADC120_WCMPUM) to the ELC, specify single-scan mode.

29.5.2 ADC12 Operation through an Event from the ELC

The ADC12 can start A/D conversion by the preset event specified in the ELSRn settings for the ELC as follows:

- Select the ELC_AD00 signal in the ELC.ELSR8 register
- Select the ELC_AD01 signal in the ELC.ELSR9 register

If an ELC event occurs during A/D conversion, the event is disabled.

29.6 Selecting Reference Voltage

The ADC12 can select VREFH0, AVCC0, or internal reference voltage (BGR) as the high-potential reference voltage and can select VREFL0 or AVSS0 as the low-potential reference voltage. Set these reference voltages before starting A/D conversion. For details on the settings, see section 29.2.38. ADHVREFCNT : A/D High-Potential/Low-Potential Reference Voltage Control Register.

29.7 A/D Conversion Procedure When Selecting Internal Reference Voltage as High-Potential Reference Voltage

The following sequence describes the A/D conversion procedure after selecting the internal reference voltage as the high-potential reference voltage. In this case, A/D conversion is possible for channels of analog input, however A/D conversion of the internal reference voltage and the temperature sensor output is prohibited.

1. Set ADHVREFCNT.HVSEL[1:0] to 11b to discharge the high-potential reference voltage path in the ADC12.
2. Wait for a 1 µs discharge period in the software.
3. Set ADHVREFCNT.HVSEL[1:0] to 10b to select internal reference voltage as the high-potential reference voltage.*1
4. Wait until the internal reference voltage is stabilized (for 5 µs) in the software, then perform A/D conversion.

Note 1. The ADC12 has a protection function that disables selection of internal reference voltage (ADHVREFCNT.HVSEL[1:0] = 10b) without discharge (ADHVREFCNT.HVSEL[1:0] = 11b) from the selection of VREFH0 (ADHVREFCNT.HVSEL[1:0] = 01b) or AVCC0 (ADHVREFCNT.HVSEL[1:0] = 00b). If the internal reference voltage is selected without discharge, discharge is forcibly set. Select the internal reference voltage again 1 µs later.

Figure 29.35 shows a waveform chart for the procedure to select internal reference voltage as the high-potential reference voltage.

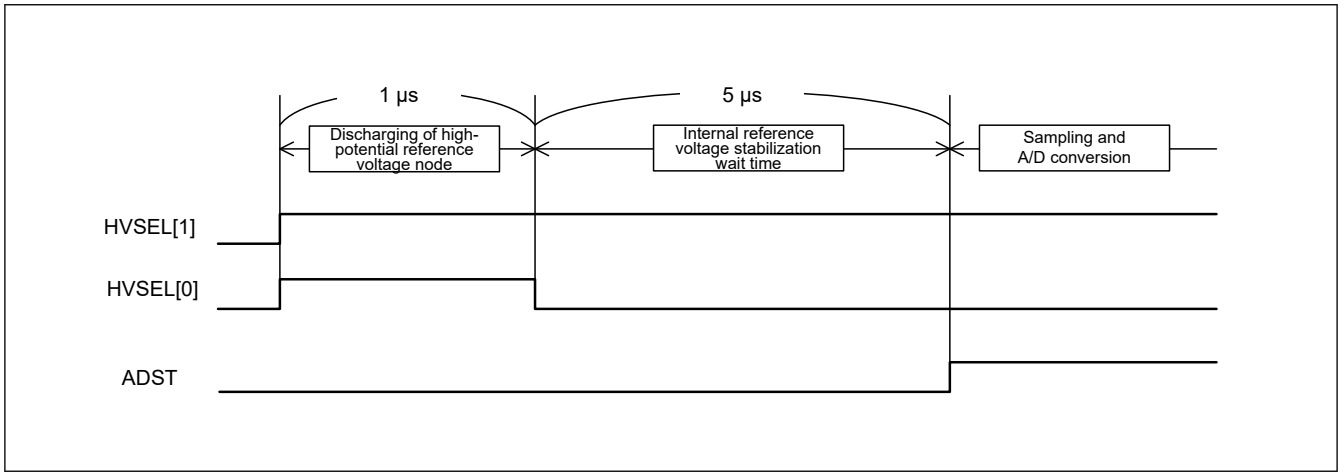


Figure 29.35 Procedure to select internal reference voltage as high-potential reference voltage

29.8 Usage Notes

29.8.1 Constraints on Setting the Registers

Set each register while the ADCSR.ADST bit is 0.

29.8.2 Constraints on Reading the Data Registers

The following registers must be read in halfword units:

- A/D Data Registers
- A/D Data Duplexing Register
- A/D Data Duplexing Register A
- A/D Data Duplexing Register B

- A/D Temperature Sensor Data Register
- A/D Internal Reference Voltage Register
- A/D Self-Diagnosis Data Register

If a register is read twice in byte units, that is, the upper byte and lower byte are read separately, the A/D-converted value read initially might disagree with the A/D-converted value read subsequently. To prevent this, never read the data registers in byte units.

29.8.3 Constraints on Stopping A/D Conversion

(1) A/D Conversion Stop Procedure

To stop A/D conversion when an asynchronous trigger or a synchronous trigger is selected as the condition for starting A/D conversion, follow the procedure shown in Figure 29.36.

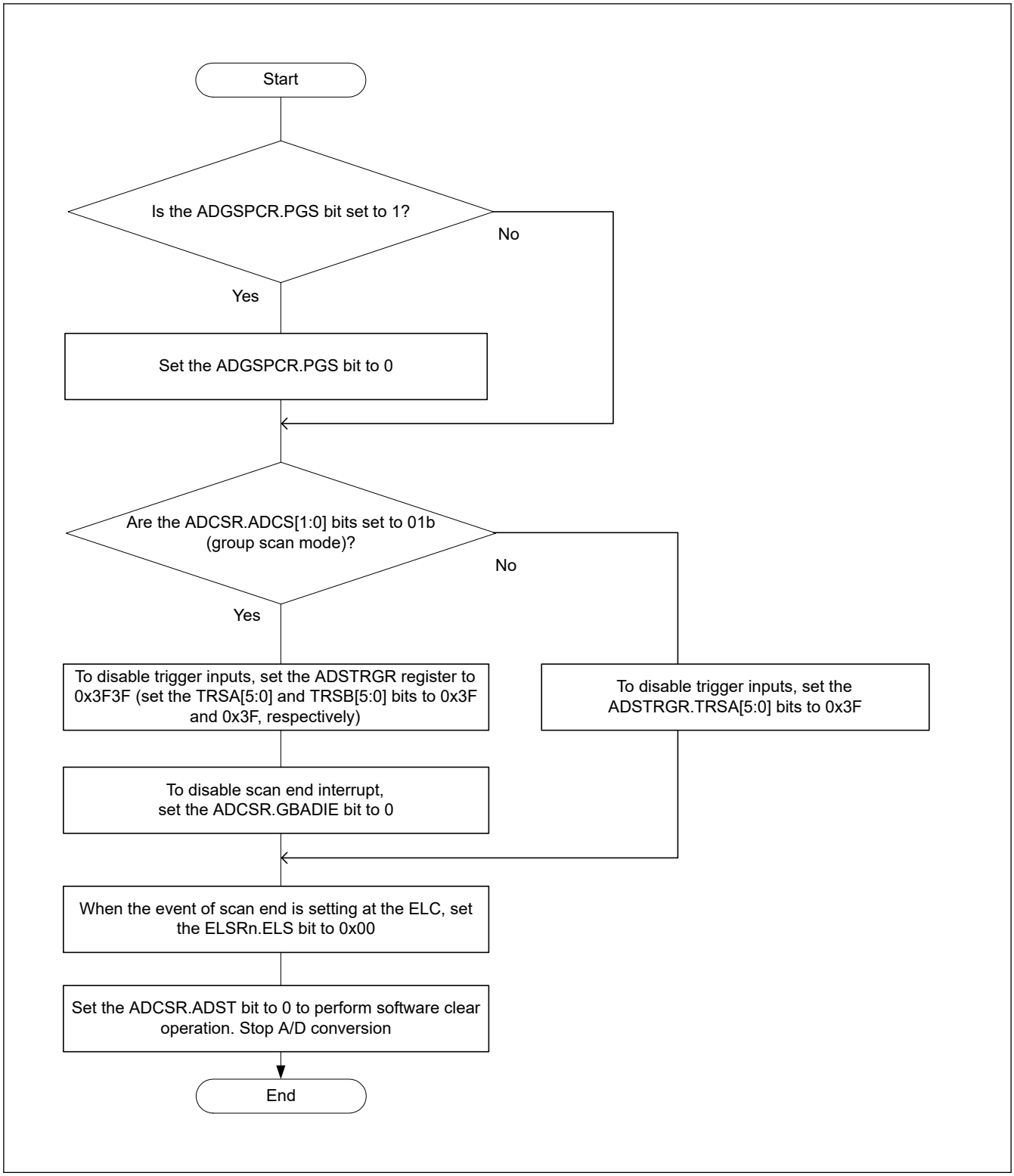


Figure 29.36 Procedures for clearing the ADCSR.ADST bit by software

To specify the following settings after performing the clear operation by software, provide a wait period for at least two ADCLK cycles.

- Enabling scan end interrupts
- Enabling scan end events for the event link controller
- Starting A/D conversion by software
- Enabling trigger input

(2) Notes on Modes and Status Bits

If necessary, individually initialize or set again the voltage status for self-diagnosis, the judgment of the even number or odd number specified for double-trigger mode, and the monitor flags of the compare function.

- To set again the voltage status for self-diagnosis, set the ADCER.DIAGLD bit to 1 and then set a desired value in the ADCER.DIAGVAL[1:0] bits.
- If the setting of the ADCSR.DBLE bit is changed from 0 to 1, the double-trigger mode operation starts from the first scanning.
- To initialize the monitor flags of the compare function (MONCMPA, MONCMPB, and MONCOMB), set the ADCMPCR.CMPAE and ADCMPCR.CMPBE bits to 0.

29.8.4 A/D Conversion Restart and Termination Timing

A maximum of 6 ADCLK cycles is required for the idle analog unit of the ADC12 to restart on setting the ADCSR.ADST bit to 1. A maximum of 3 ADCLK cycles is required for the operating analog unit of the ADC12 to terminate on setting the ADCSR.ADST bit to 0.

29.8.5 Constraints on Scan End Interrupt Handling

When scanning the same analog input twice using any trigger, the first A/D-converted data is overwritten with the second A/D-converted data. This occurs when the CPU does not complete the reading of the A/D-converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated.

29.8.6 Settings for the Module-Stop Function

The Module Stop Control Register can enable or disable ADC12 operation. The ADC12 is initially stopped after a reset. The registers become accessible on release from the module-stop state. After release from the module-stop state, wait for at least 1 µs before starting A/D conversion. For details, see section 10, Low Power Modes.

29.8.7 Notes on Entering the Low-Power States

Before entering the module-stop state or Software Standby mode, be sure to stop A/D conversion. Set the ADCSR.ADST bit in ADCSR to 0 and secure certain period until the analog unit of the ADC12 stops. Follow the procedure shown in Figure 29.36 to clear the ADCSR.ADST bit with software. Then, wait for 3 clock cycles of ADCLK before entering the module-stop state or Software Standby mode.

29.8.8 Error in Absolute Accuracy When Disconnection Detection Assistance Is in Use

Using disconnection detection assistance leads to an error in absolute accuracy of the ADC12. This error arises because an erroneous voltage is input to the analog input pins due to the resistive voltage division between the pull-up or pull- down resistor (Rp) and the resistance of the signal source (Rs). This error in absolute accuracy is calculated from the following formula:

$$\text{Maximum error in absolute accuracy (LSB)} = \left(2^{\text{Resolution}} - 1\right) \times \text{Rs} / \left(\text{Rs} + \text{Rp}\right)$$

Only use disconnection detection assistance after thorough evaluation.

29.8.9 ADHSC Bit Rewriting Procedure

Before changing the A/D Conversion Select bit (ADCSR.ADHSC) from 0 to 1 or from 1 to 0, the ADC12 must be in the standby state. Use the following procedure to modify the ADCSR.ADHSC bit. After the Sleep bit (ADHVREFCNT.ADSLP) is set to 0, wait for at least 1 µs then start the A/D conversion.

1. Set the Sleep bit (ADHVREFCNT.ADSLP) to 1.
2. Wait for at least 0.2 µs, then modify the A/D Conversion Select bit (ADCSR.ADHSC).
3. Wait for at least 4.8 µs, then set the Sleep bit (ADHVREFCNT.ADSLP) to 0.

Note: Do not set the Sleep bit (ADHVREFCNT.ADSLP) to 1 except when modifying the A/D Conversion Select bit (ADCSR.ADHSC).

Note: Do not reset the Sleep bit when the A/D Conversion Select bit (ADCSR.ADHSC) is 1. After this bit is set to 0 or the operating mode transitions to the module-stop mode, reset the Sleep bit using the ADCSR.ADHSC bit rewriting procedure.

29.8.10 Constraints on Operating Modes and Status Bits

Initialize or set again individually, if necessary, the voltage values in self-diagnosis, the value of the first scan or second scan in double trigger mode, the data buffer pointer, and status monitor in the compare function.

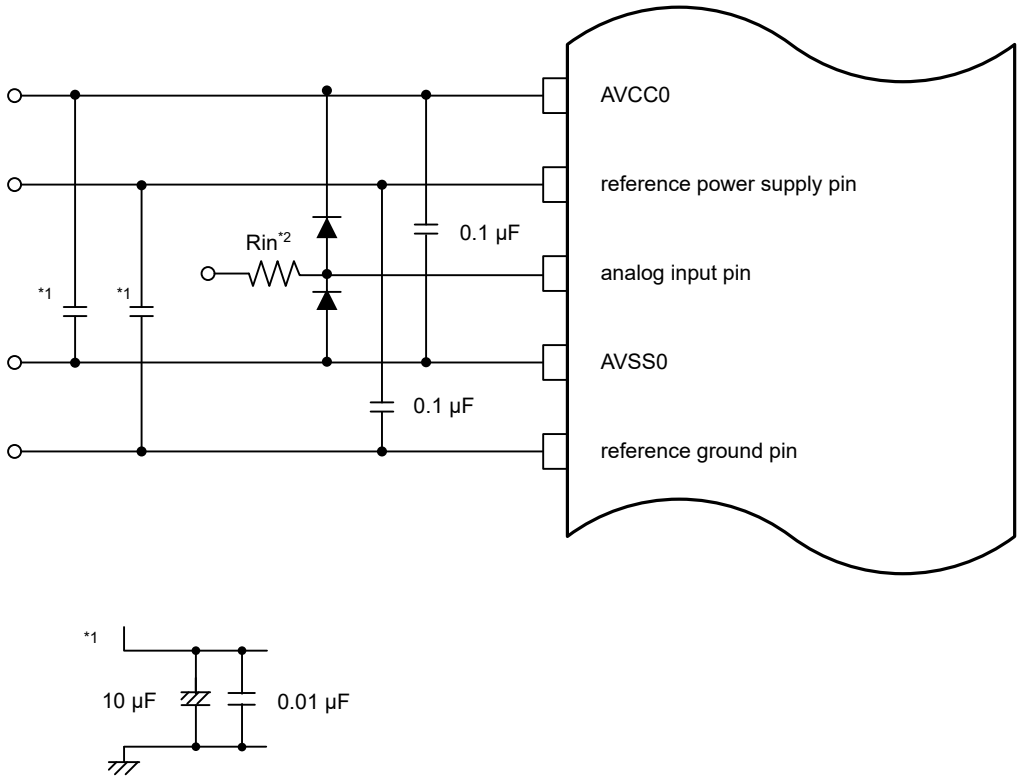
- Select the voltage values in self-diagnosis (ADCER.DIAGVAL[1:0]) after setting ADCER.DIAGLD to 1.
- Double-trigger mode operates as the first scan after setting ADCSR.DBLE from 0 to 1.
- The status monitor bits (MONCMPA, MONCMPB, MONCOMB) in the compare function are initialized after setting ADCMPCR.CMPAE and ADCMPCR.CMPBE to 0.

29.8.11 Notes on Board Design

The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or be placed near each other. If these rules are not followed, noise can occur on analog signals and A/D conversion accuracy is affected. The analog input pins, reference power supply pin (VREFH0), reference ground pin (VREFL0), and analog power supply (AVCC0) should be separated from digital circuits using the analog ground (AVSS0). The analog ground (AVSS0) should be connected to a stable digital ground (VSS) on the board (single-point ground plane connection).

29.8.12 Constraints on Noise Prevention

To prevent the analog input pins from being destroyed by abnormal voltage such as excessive surge, insert a capacitor between AVCC0 and AVSS0 and between VREFH0 and VREFL0. Additionally, connect a protection circuit to protect the analog input pins as shown in Figure 29.37.



Note 1. The values shown here are reference values.
Note 2. Rin: Signal source impedance

Figure 29.37 Example protection circuit for analog inputs

29.8.13 Port Settings When Using the ADC12 Input

When using the high-precision channels, do not use PORT0 as digital output ports. Renesas recommends that you do not use the digital output that is also used as the AD analog input if normal-precision channel is used. If the digital output that is also used as the AD analog input is used for output signals, perform A/D conversion several times, eliminate the maximum and minimum values, and obtain the average of the other results.

29.8.14 Notes on Canceling Software Standby Mode

After software standby mode is canceled, wait at least 1 µs after the stabilization time for the oscillator elapses and before starting A/D conversion. For details, see section 10, Low Power Modes

29.8.15 S/H circuit hold time limitations

The S/H circuit has a hold retention time restriction (For details, see section 35, Electrical Characteristics.).

Hold retention time is the time that the S/H circuit can hold the sampled value after it finishes sampling, and AD conversion must be completed during that time.

30. Temperature Sensor (TSN)

30.1 Overview

The on-chip Temperature Sensor (TSN) determines and monitors the die temperature for reliable operation of the device. The sensor outputs a voltage directly proportional to the die temperature, and the relationship between the die temperature and the output voltage is fairly linear. The output voltage is provided to the ADC12 for conversion and can be further used by the end application.

Table 30.1 lists the TSN specifications, and Figure 30.1 shows a block diagram.

Table 30.1 TSN specifications

Item	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the 12-bit A/D converter

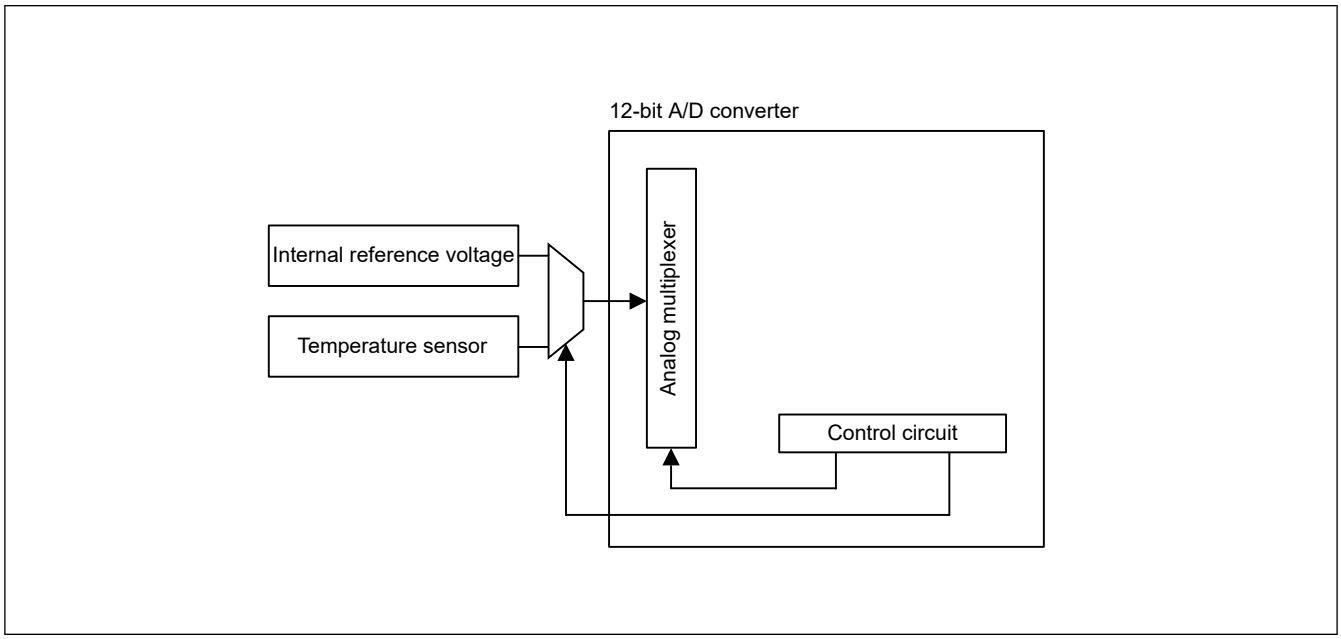


Figure 30.1 TSN block diagram

30.2 Register Descriptions

30.2.1 TSCDR : Temperature Sensor Calibration Data Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0228

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSCDR[15:0]															
Value after reset:	Chip-specific value															

Bit	Symbol	Function	R/W
15:0	TSCDR[15:0]	Temperature Sensor Calibration Data Chip-specific value	R

The TSCDR register stores temperature sensor calibration data measured for each chip at factory shipment.

Temperature sensor calibration data is the output voltage of the temperature sensor under the conditions Tj = 140°C and AVCC0 = 3.3 V converted to a digital value by the 12-bit A/D converter.

Temperature sensor calibration data is stored in the lower 12 bits of the TSCDR register.

30.3 Using the Temperature Sensor

The temperature sensor outputs a voltage that varies with the temperature. This voltage is converted to a digital value by the 12-bit A/D converter. To obtain the die temperature, convert this value into the temperature.

30.3.1 Preparation for Using the Temperature Sensor

The ambient temperature (T) is proportional to the temperature sensor voltage output (Vs), so ambient temperature is calculated with the following formula:

T = (Vs - V1) / slope + T1

- T: Ambient temperature of MCU as calculation result (°C)
- Vs: Voltage output by the temperature sensor on temperature measurement (V)
- T1: Temperature experimentally measured at one point (°C)
- V1: Voltage output by the temperature sensor on measurement of T1 (V)
- T2: Temperature experimentally measured at a second point (°C)
- V2: Voltage output by the temperature sensor on measurement of T2 (V)
- Slope: Temperature gradient of the temperature sensor (V / °C), slope = (V2 - V1) / (T2 -T1)

Characteristics vary between sensors, so Renesas recommends measuring two different sample temperatures as follows:

1. Use the 12-bit A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1.
2. Again use the 12-bit A/D converter to measure the voltage V2 output by the temperature sensor at a different temperature T2.
3. Obtain the temperature gradient (slope = (V2 - V1) / (T2 - T1)) from these results.
4. Subsequently, obtain temperatures by substituting the slope into the formula for the temperature characteristic (T = (Vs -V1) / slope + T1).

If you are using the temperature gradient given in section 35, Electrical Characteristics, use the A/D converter to measure the voltage V1 output by the temperature sensor at temperature T1, then calculate the temperature characteristic using the following formula:

T = (Vs - V1) / slope + T1

Note: This method produces less accurate temperatures than measurement at two points.

In this MCU, the TSCDR register stores the temperature value (CAL125) of the temperature sensor measured under the condition Ta = Tj = 140°C and AVCC0 = 3.3 V. If you use this value as the sample measurement result at the first point, you can omit the preparation before using the temperature sensor.

CAL125 = TSCDR register value

V1 is calculated from CAL125:

V1 = 3.3 × CAL125 / 4096 [V] (In case of 12 bit accuracy)

Using this value, the measured temperature can be calculated according to the following formula:

T = (Vs - V1) / slope + 140 [°C]

- T: Ambient temperature of MCU as calculation result (°C)
- Vs: Voltage output by the temperature sensor when the temperature is measured (V)
- V1: Voltage output by the temperature sensor when Ta = Tj = 140°C and AVCC0 = 3.3 V (V)
- Slope: Temperature gradient of the temperature sensor*1 / 1000 (V/°C)

Note 1. See section 35, Electrical Characteristics

Figure 30.2 shows the error in the measured temperature. The variation range is 3σ.

Regarding the characteristics of the 12-bit A/D converter, the typical values are used. See section 35.4. ADC12 Characteristics.

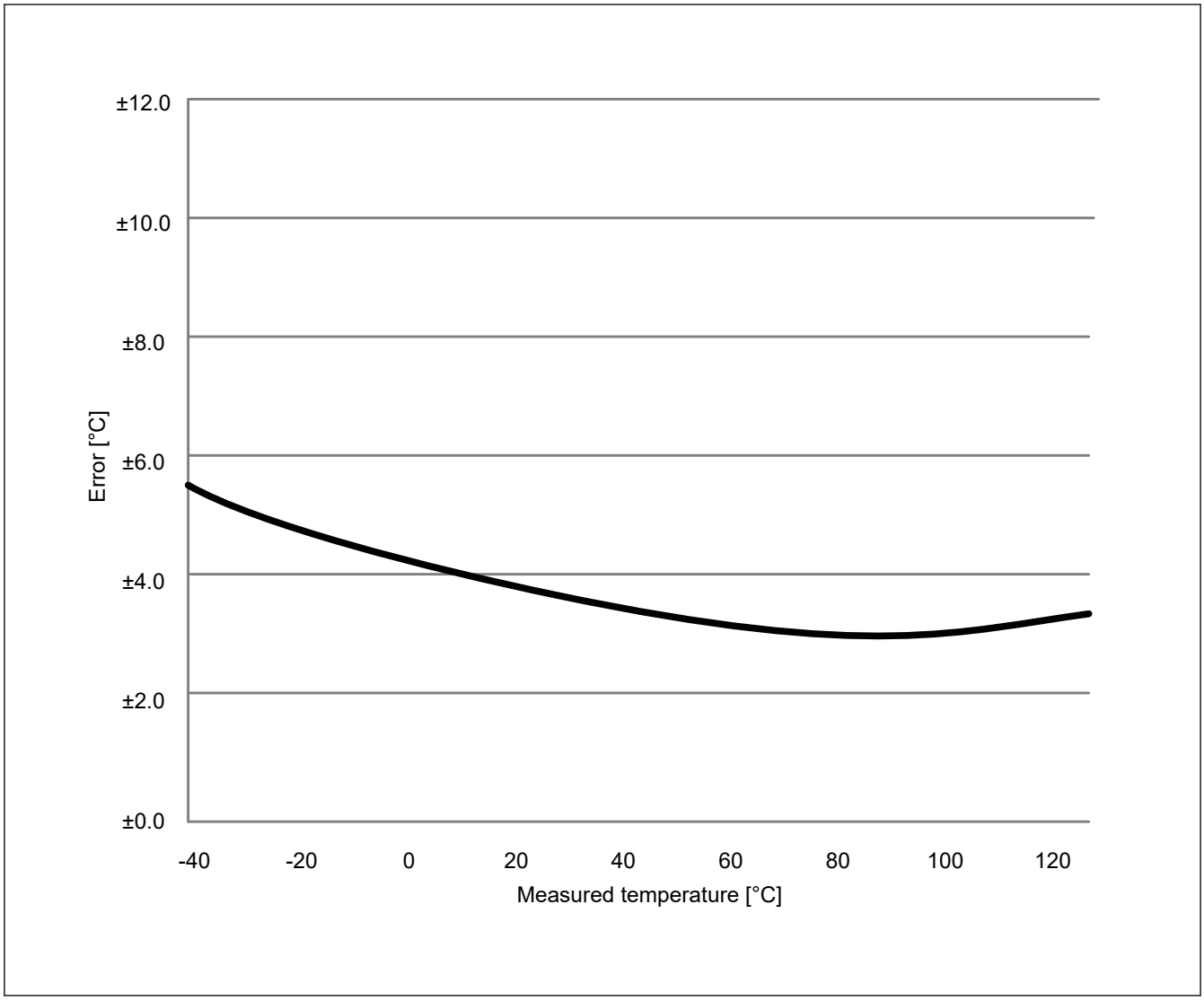


Figure 30.2 Error in the measured temperature (designed values)

30.3.2 Procedures for Using the Temperature Sensor

For details, see section 29, 12-Bit A/D Converter (ADC12).

31. Data Operation Circuit (DOC)

31.1 Overview

The Data Operation Circuit (DOC) compares, adds, and subtracts 16-bit data. When a selected condition applies, 16-bit data is compared and an interrupt can be generated. Table 31.1 lists the DOC specifications and Figure 31.1 shows a block diagram.

Table 31.1 DOC specifications

Item	Description
Data operation function	16-bit data comparison, addition, and subtraction
Module-stop function	The module-stop state can be set to reduce power consumption.
Interrupts and event link function (DOC_DOPCI)	An interrupt occurs on the following conditions: • The compared values either match or mismatch • The result of data addition is greater than 0xFFFF • The result of data subtraction is less than 0x0000

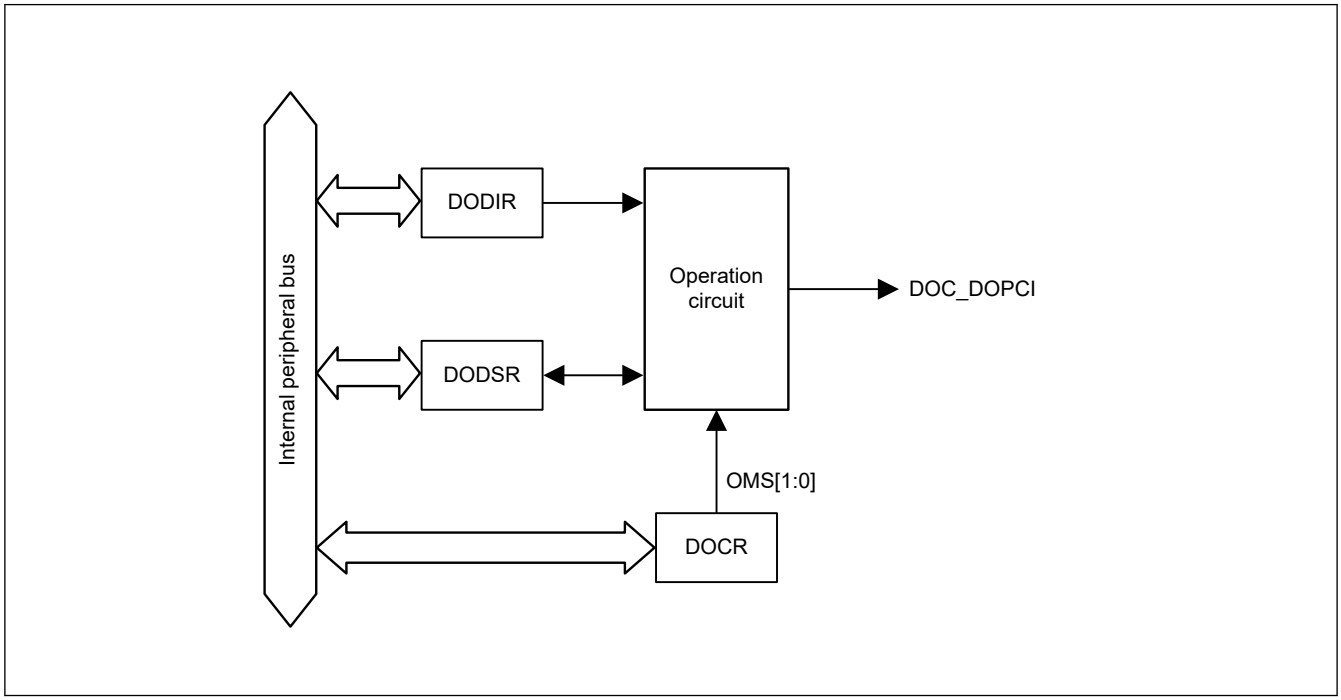


Figure 31.1 DOC block diagram

31.2 DOC Register Descriptions

31.2.1 DOCR : DOC Control Register

Base address: DOC = 0x4005_4100

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	DOPCFCL	DOPCF	—	—	DCSEL	OMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OMS[1:0]	Operating Mode Select 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
2	DCSEL*1	Detection Condition Select 0: Set DOPCF flag when data mismatch is detected 1: Set DOPCF flag when data match is detected	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	DOPCF	DOC Flag Indicates the result of an operation.	R
6	DOPCFCL	DOPCF Clear 0: Retain DOPCF flag state 1: Clear DOPCF flag	R/W
7	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only valid when data comparison mode is selected.

OMS[1:0] bits (Operating Mode Select)

The OMS[1:0] bits select the operating mode of the DOC.

DCSEL bit (Detection Condition Select)

The DCSEL bit selects the detection condition in data comparison mode. This bit is only valid when data comparison mode is selected.

DOPCF flag (DOC Flag)

The DOPCF flag indicates the result of an operation.

[Setting conditions]

- The result of data comparison matches the condition selected in the DCSEL bit
- A data addition result is greater than 0xFFFF
- A data subtraction result is less than 0x0000

[Clearing condition]

- Writing 1 to the DOPCFCL bit

DOPCFCL bit (DOPCF Clear)

Setting the DOPCFCL bit to 1 clears the DOPCF flag. This bit is read as 0.

31.2.2 DODIR : DOC Data Input Register

Base address: DOC = 0x4005_4100

Offset address: 0x02

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

Bit	Symbol	Function	R/W
15:0	n/a	It stores 16-bit data used in the operations.	R/W

31.2.3 DODSR : DOC Data Setting Register

Base address: DOC = 0x4005_4100

Offset address: 0x04

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

Bit	Symbol	Function	R/W
15:0	n/a	It stores 16-bit data used as a reference in data comparison mode. This register also stores the results of operations in data addition and subtraction modes.	R/W

31.3 Operation

31.3.1 Data Comparison Mode

Figure 31.2 shows an example operation in data comparison mode operation by the DOC. The following sequence is an example operation when DCSEL is set to 0 (data mismatch is detected as a result of data comparison):

1. Write 00b to the DOCR.OMS[1:0] bits to select data comparison mode.
2. Set 16-bit reference data in DODSR.
3. Write the 16-bit data for comparison to DODIR.
4. Continue writing the 16-bit data until all data for comparison is written to DODIR.
5. If a value written to DODIR does not match that in DODSR, the DOCR.DOPCF flag is set to 1.

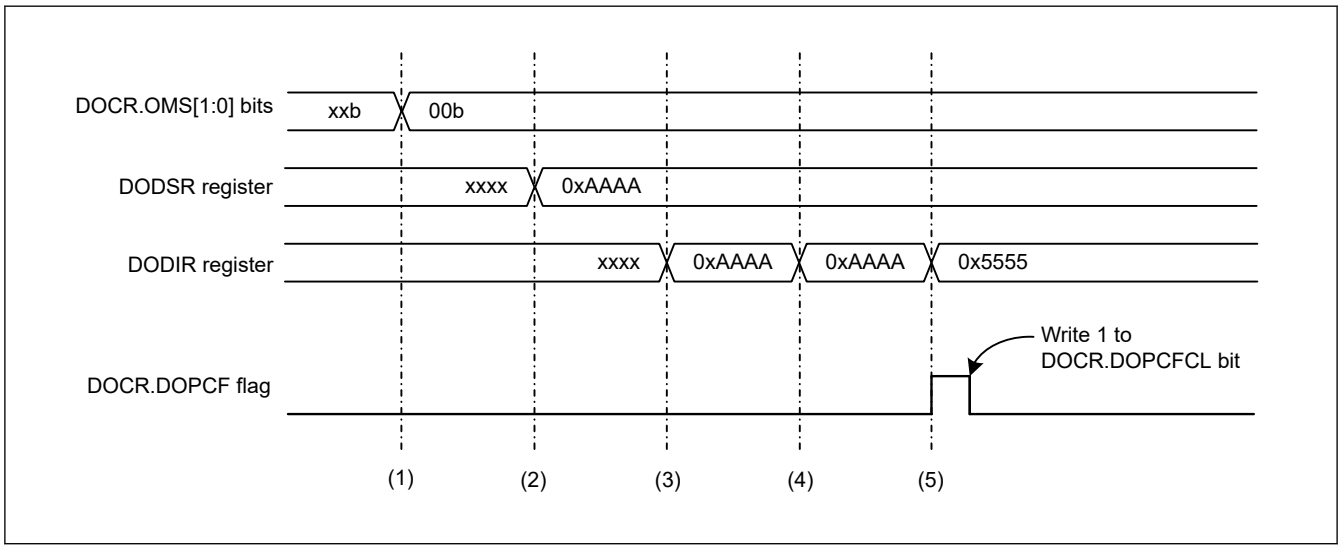


Figure 31.2 Example of operation in data comparison mode

31.3.2 Data Addition Mode

Figure 31.3 shows an example operation in data addition mode. The steps are as follows:

1. Write 01b to the DOCR.OMS[1:0] bits to select data addition mode.
2. Set 16-bit data as the initial value in the DODSR register.
3. Write the 16-bit data to be added to the DODIR register. The result of the operation is stored in the DODSR register.
4. Continue writing the 16-bit data until all data to be added is written to the DODIR.
5. If the result of an operation is greater than 0xFFFF, the DOCR.DOPCF flag is set to 1.

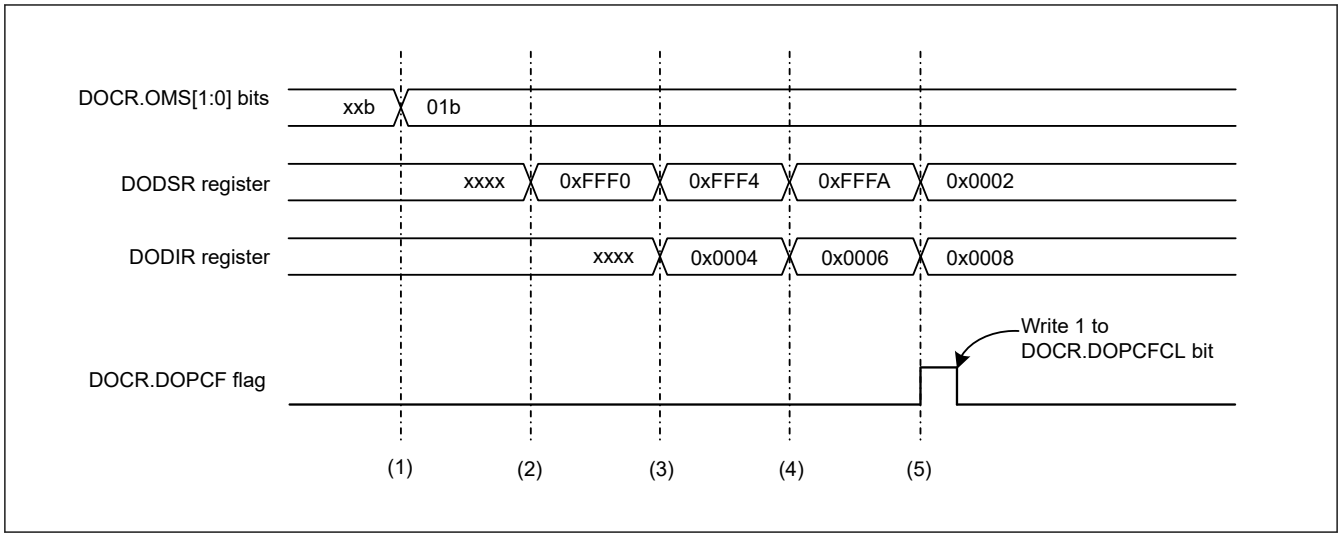


Figure 31.3 Example of operation in data addition mode

31.3.3 Data Subtraction Mode

Figure 31.4 shows an example operation in data subtraction mode. The steps are as follows:

1. Write 10b to the DOCR.OMS[1:0] bits to select data subtraction mode.
2. Set 16-bit data as the initial value in the DODSR register.
3. Write the 16-bit data to be subtracted to the DODIR register. The result of the operation is stored in DODSR.
4. Continue writing the 16-bit data to the DODIR register until all data to be subtracted is written.
5. If the result of an operation is less than 0x0000, the DOCR.DOPCF flag is set to 1.

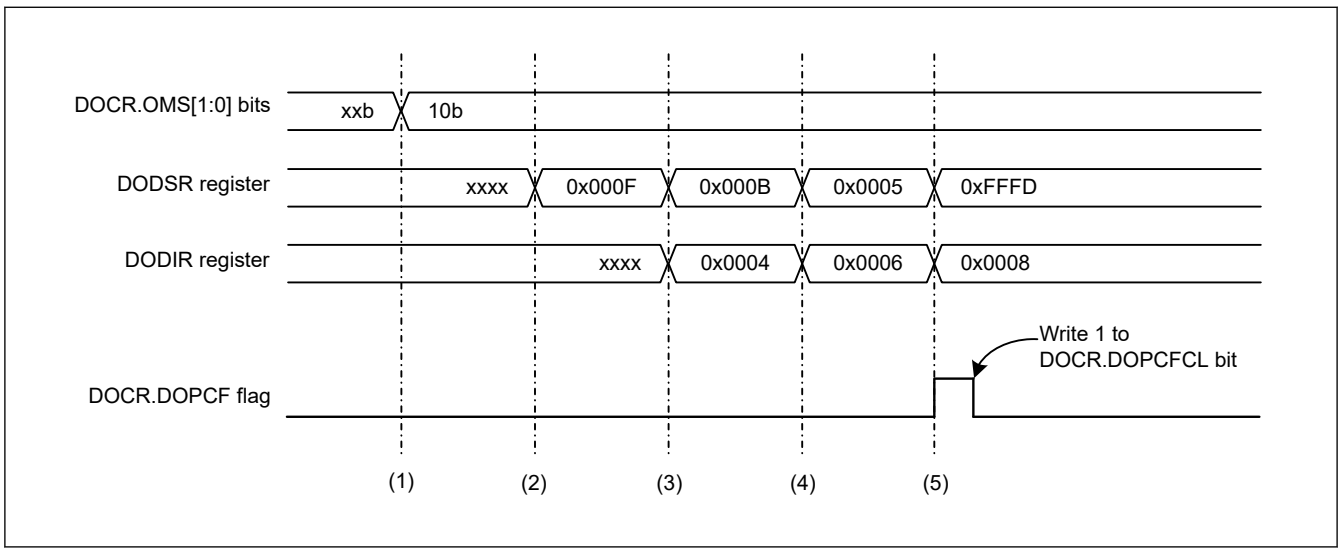


Figure 31.4 Example of operation in data subtraction mode

31.4 Interrupt Source

The DOC generates the DOC interrupt (DOC_DOPCI) as an interrupt request. Table 31.2 describes the DOC interrupt request.

Table 31.2 Interrupt request from DOC

Interrupt request	Status flag	Interrupt source
DOC interrupt	DOPCF	• The result of data comparison matches the condition selected in the DOCR.DCSEL bit. • The result of data addition is greater than 0xFFFF. • The result of data subtraction is less than 0x0000.

31.5 Output of an Event Signal to the Event Link Controller (ELC)

The DOC outputs an event signal for the ELC under the following conditions:

- The compared values either match or mismatch
- The data addition result is greater than 0xFFFF
- The data subtraction result is less than 0x0000

This signal can be used to initiate operations by other modules selected in advance and can also be used as an interrupt request. When an event signal is generated, the DOC Flag (DOCR.DOPCF) is set to 1.

31.6 Usage Notes

31.6.1 Settings for the Module-Stop State

The module Stop Control Register C (MSTPCRC) can enable or disable DOC operation. The DOC is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 10, Low Power Modes.

32. SRAM

32.1 Overview

The MCU provides an on-chip, high-density SRAM module with parity-bit checking. Parity check is performed on the all SRAM areas.

Table 32.1 lists the SRAM specifications.

Table 32.1 SRAM specifications

Parameter	Description
SRAM capacity	SRAM0: 8-KB
SRAM address	SRAM0: 0x2000_4000 to 0x2000_5FFF
Access*1	0 wait for both reading and writing
Parity	Even parity with 8-bit data and 1-bit parity
Error checking	Even parity error check

Note: SRAM0 and Trace RAM are shared. For the Trace RAM specifications, see *ARM® CoreSight™ MTB-M23 Technical Reference Manual (ARM DDI 0564C)*.

Note 1. For details, see section 32.3.3. Access Cycle.

32.2 Register Descriptions

32.2.1 PARIOAD : SRAM Parity Error Operation After Detection Register

Base address: SRAM = 0x4000_2000

Offset address: 0x00

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

Bit	Symbol	Function	R/W
0	OAD	Operation After Detection 0: Non-maskable interrupt 1: Reset	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

The PARIOAD register controls the operation on detection of a parity error. The SRAM Protection Register (SRAMPRCR) protects this register against writes. Always set the SRAMPRCR bit in SRAMPRCR to 1 before writing to this bit. Do not write to the PARIOAD register while accessing the SRAM.

OAD bit (Operation After Detection)

The OAD bit specifies the generation of either a reset or non-maskable interrupt when a parity error is detected. The OAD bit is commonly used for SRAM0.

32.2.2 SRAMPRCR : SRAM Protection Register

Base address: SRAM = 0x4000_2000

Offset address: 0x04

Bit position:	7						1	0
Bit field:	KW[6:0]							SRAM PRCR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SRAMPRCR	Register Write Control 0: Disable writes to protected registers 1: Enable writes to protected registers	R/W
7:1	KW[6:0]	Write Key Code These bits enable or disable writes to the SRAMPRCR bit	W

SRAMPRCR bit (Register Write Control)

The SRAMPRCR bit controls the write mode of the PARIOAD register. Setting the bit to 1 enables writes to the PARIOAD register. When you write to this bit, always write 0x78 to KW[6:0] bits simultaneously.

KW[6:0] bits (Write Key Code)

The KW[6:0] bits enable or disable writes to the SRAMPRCR bit. When you write to the SRAMPRCR bit, always write 0x78 to these bits simultaneously. When a value other than 0x78 is written to KW[6:0], the SRAMPRCR bit is not updated. The KW[6:0] bits are always read as 0x00.

32.2.3 Trace Control (for the MTB)

The Micro Trace Buffer (MTB) has programmable registers to control the behavior of the trace features and the POSITION, MASTER, FLOW, and BASE registers. Table 32.2 shows the registers in offset order from the base address.

Table 32.2 Address of MTB registers

Address	Register	Value on reset
MTB_BASE + 0x000	MTB_POSITION	Bits [31:0] = UNKNOWN
MTB_BASE + 0x004	MTB_MASTER	Bits [31] = 0, Bits [30:10] = UNKNOWN, Bits [9:8] = 0, Bits [7] = 1, Bits [6:5] = 0, Bits [4:0] = UNKNOWN
MTB_BASE + 0x008	MTB_FLOW	Bits [31:2] = UNKNOWN, Bits [1:0] = 0
MTB_BASE + 0x00C	MTB_BASE	Bits [31:0] = 0x2000_4000

Note: MTB_BASE = 0x4001_9000

For more information on these registers, see the *ARM® CoreSight™ MTB-M23 Technical Reference Manual (ARM DDI 0564C)*.

Note: Do not attempt to access reserved or unused address locations.

The MTB for trace is limited from 0x2000_4000 to 0x2000_5FFF.

32.2.4 CoreSight™ (for MTB)

See the *ARM® CoreSight™ Architecture Specification* for more information about the registers and access types. Table 32.3 shows the registers in offset order from the base address.

Table 32.3 Address of CoreSight

Address	Register
MTB_BASE + 0xFF0 to 0xFFC	Component ID
MTB_BASE + 0xFE0 to 0xFDC	Peripheral ID
MTB_BASE + 0xFCC	Device Type Identifier
MTB_BASE + 0xFC8	Device Configuration
MTB_BASE + 0xFBC	Device Architecture
MTB_BASE + 0xFB8	Authentication Status
MTB_BASE + 0xFB4	Lock Status
MTB_BASE + 0xFB0	Lock Access

Note: MTB_BASE = 0x4001_9000

For more information on these registers, see the *ARM® CoreSight™ MTB-M23 Technical Reference Manual (ARM DDI 0564C)*.

Note: Do not attempt to access reserved or unused address locations.

32.3 Operation

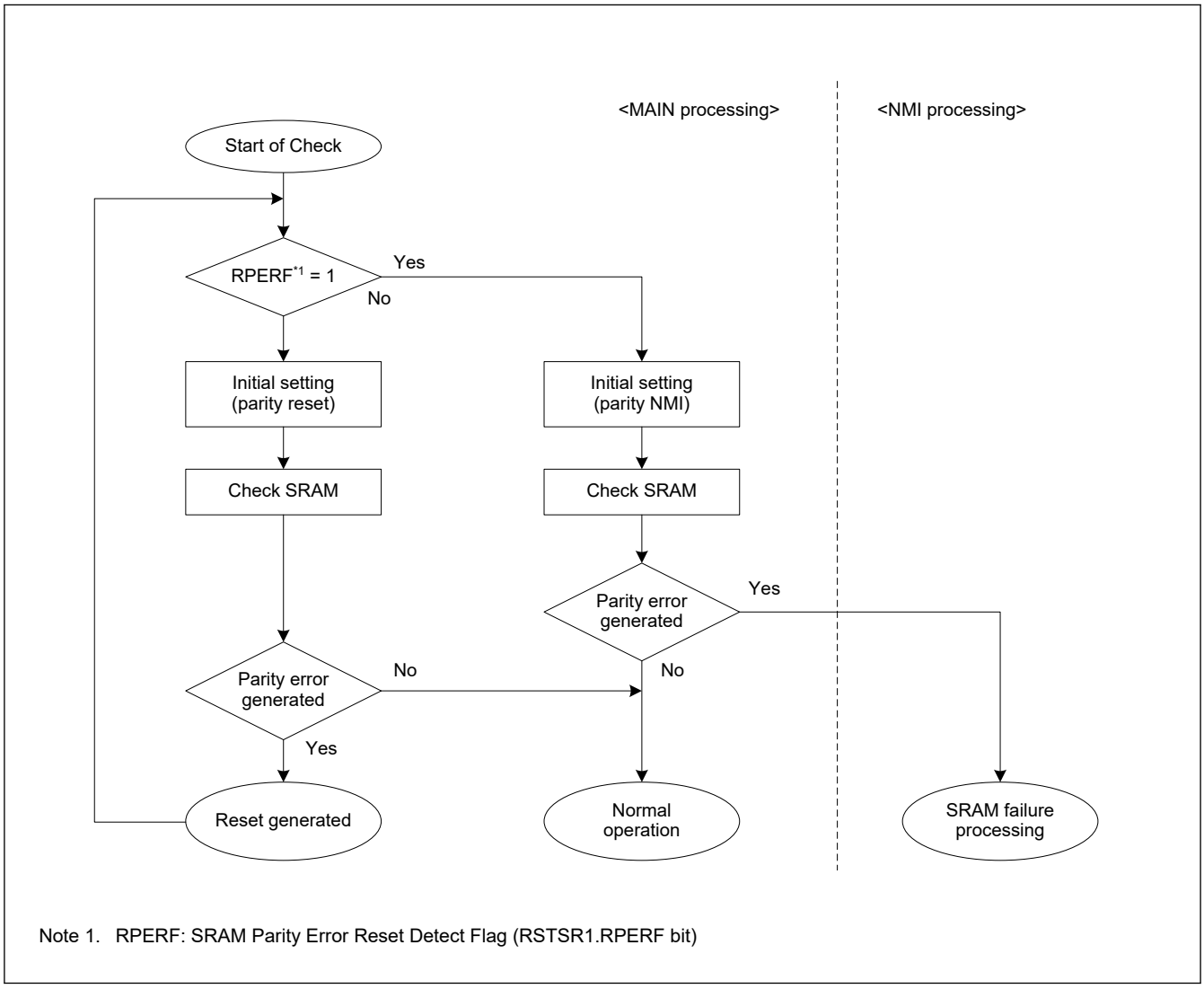
32.3.1 Parity Calculation Function

The IEC60730 standard requires the checking of SRAM data. When data is written, a parity bit is added to every 8-bit data in the SRAM which has 32-bit data width, and when data is read, the parity is checked. When a parity error occurs, a parity-error notification is generated. This function can also be used to trigger a reset.

The parity-error notification can be specified as a non-maskable interrupt or a reset in the OAD bit of the PARIOAD register. When the OAD bit is set to 1, a parity error is output to the reset function. When the OAD bit is set to 0, a parity error is output to the ICU as a non-maskable interrupt.

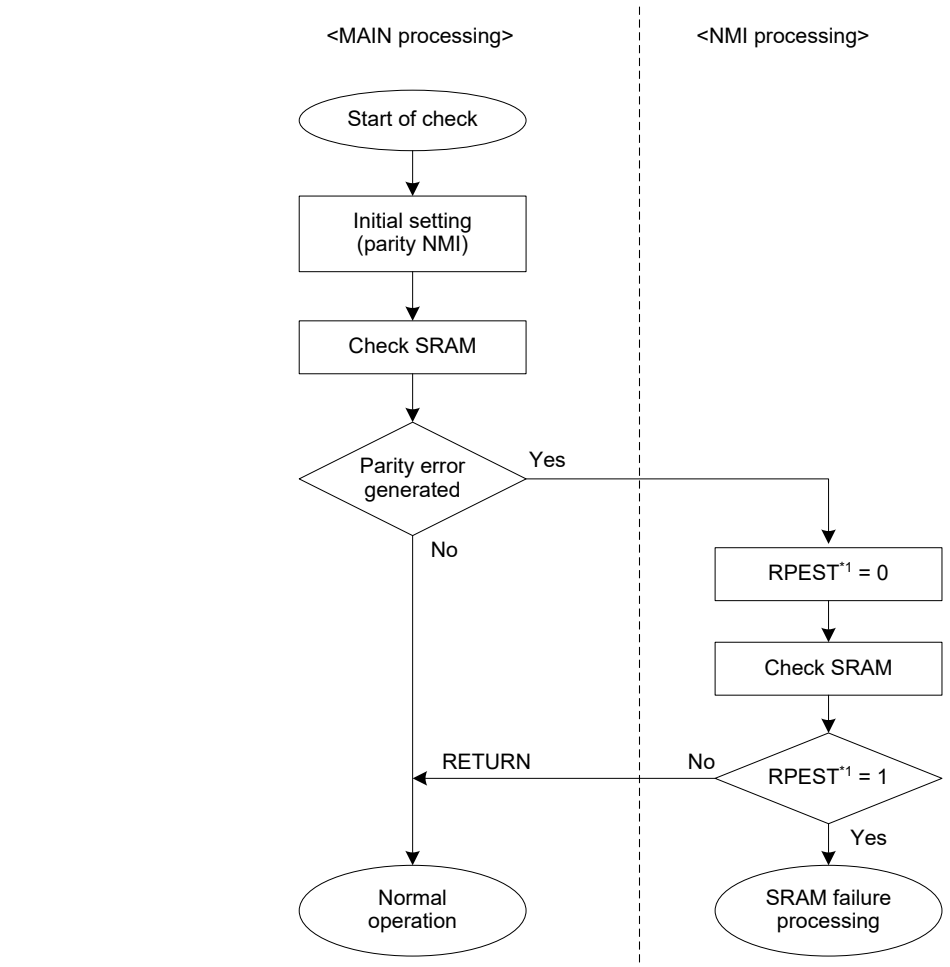
Parity errors can be occasionally caused by noise. To confirm whether the cause of the parity error is noise or corruption, follow the parity check flows shown in Figure 32.1 and Figure 32.2.

When a read access is executed in a row after a write access, read access is executed with priority. Therefore, during initialization, do not perform the read access in a row after the write access.



Note 1. RPERF: SRAM Parity Error Reset Detect Flag (RSTSR1.RPERF bit)

Figure 32.1 Flow of SRAM parity check when SRAM parity reset is enabled



Note 1. RPEST: SRAM Parity Error Interrupt Status Flag (NMISR.RPEST bit)

Figure 32.2 Flow of SRAM parity check when SRAM parity interrupt is enabled

32.3.2 SRAM Error Sources

An SRAM error is a parity error. Parity error can generate either a non-maskable interrupt or a reset, as selected with the OAD bit in the PARIOAD register. DTC activation is not supported for SRAM parity errors.

Table 32.4 SRAM error sources

SRAM error source	DTC activation
Parity error (SRAM0 area)	Not possible

32.3.3 Access Cycle

Table 32.5 SRAM0 (parity area 0x2000_4000 to 0x2000_5FFF)

Read (cycles)		Write (cycles)	
Word access	Halfword/Byte access	Word access	Halfword/Byte access
2		2	

32.3.4 Low-Power Function

Power consumption can be further reduced in Software Standby mode as the supply voltage for SRAM0 can be off, except for the 8 KB in the head area of SRAM0 (0x2000_4000 to 0x2000_5FFF) of SRAM0(Parity area). For details on Software Standby mode, see section 10, Low Power Modes.

32.4 Usage Notes

32.4.1 Instruction Fetch from the SRAM Area

When using SRAM0 to operate a program, initialize the SRAM area so that the CPU can correctly prefetch data. If the CPU prefetches data from an SRAM area that is not initialized, a parity error might occur. Initialize the additional 2-byte area from the end address of a program with a 4-byte boundary. Renesas recommends using the NOP instruction for data initialization.

32.4.2 SRAM Store Buffer

For fast access between SRAM and CPU, a store buffer is used. When a load instruction is executed from the same address after a store instruction to SRAM, the load instruction might read data from the buffer instead of data on the SRAM. To read data on the SRAM correctly, use either of the following procedures:

- After writing to the SRAM (address = A), use the NOP instruction, then read the SRAM (address = A)
- After writing to the SRAM (address = A), read data from area other than SRAM (address = A), then read the SRAM (address = A).

33. Flash Memory

33.1 Overview

The MCU provides up to 64-KB code flash memory and 2-KB data flash memory. The Flash Control Block (FCB) controls the programming commands. This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.

Table 33.1 lists the specifications of the code flash memory and data flash memory, and Figure 33.1 shows a block diagram of the related modules. Figure 33.2 shows the configuration of the code flash memory, and Figure 33.3 shows the configuration of the data flash memory.

Table 33.1 Code flash memory and data flash memory specifications

Parameter	Code flash memory	Data flash memory
Memory capacity	• 64-KB of user area • Configuration setting area (See section 6, Option-Setting Memory)	2-KB of data area
Read cycle	• ICLK frequency ≤ 64 MHz MEMWAIT = 1 with wait A read operation takes 3 cycles • ICLK frequency ≤ 32 MHz MEMWAIT = 0 without wait A read operation takes 2 cycles	• ICLK frequency ≤ 64 MHz FLDWAIT1 = 1 with 2 wait A read operation takes 4 cycles • ICLK frequency ≤ 32 MHz FLDWAIT1 = 0 with 1 wait A read operation takes 3 cycles
Value after erasure	0xFF	0xFF
Programming/erasing method	• Programming and erasure of code and data flash memory through the FCB commands specified in the registers • Programming by dedicated flash-memory programmer through a serial interface (serial programming) • Programming of flash memory by user program (self-programming)*1.	
Security function	Protection against illicit tampering with or reading of data in flash memory	
Protection	Protection against erroneous overwriting of flash memory	
Background operation (BGO)	Code flash memory can be read during data flash memory programming	
Units of programming and erasure	• 32-bit units for programming in user area • 2-KB units for erasure in user area.	• 8-bit units for programming in data area • 1-KB units for erasure in data area.
Other functions	Interrupts accepted during self-programming	
	Option-setting memory can be set in the initial MCU settings	
On-board programming	Programming in serial programming mode (SCI boot mode): • Asynchronous serial interface (SCI9) used • Transfer rate adjusted automatically. Programming in on-chip debug mode: • SWD interface used • Dedicated hardware not required. Programming by a routine for code and data flash memory programming within the user program: • Allows code and data flash memory programming without resetting the system.	

Note 1. HOCO should be stably oscillated. See section 33.12. Self-Programming.

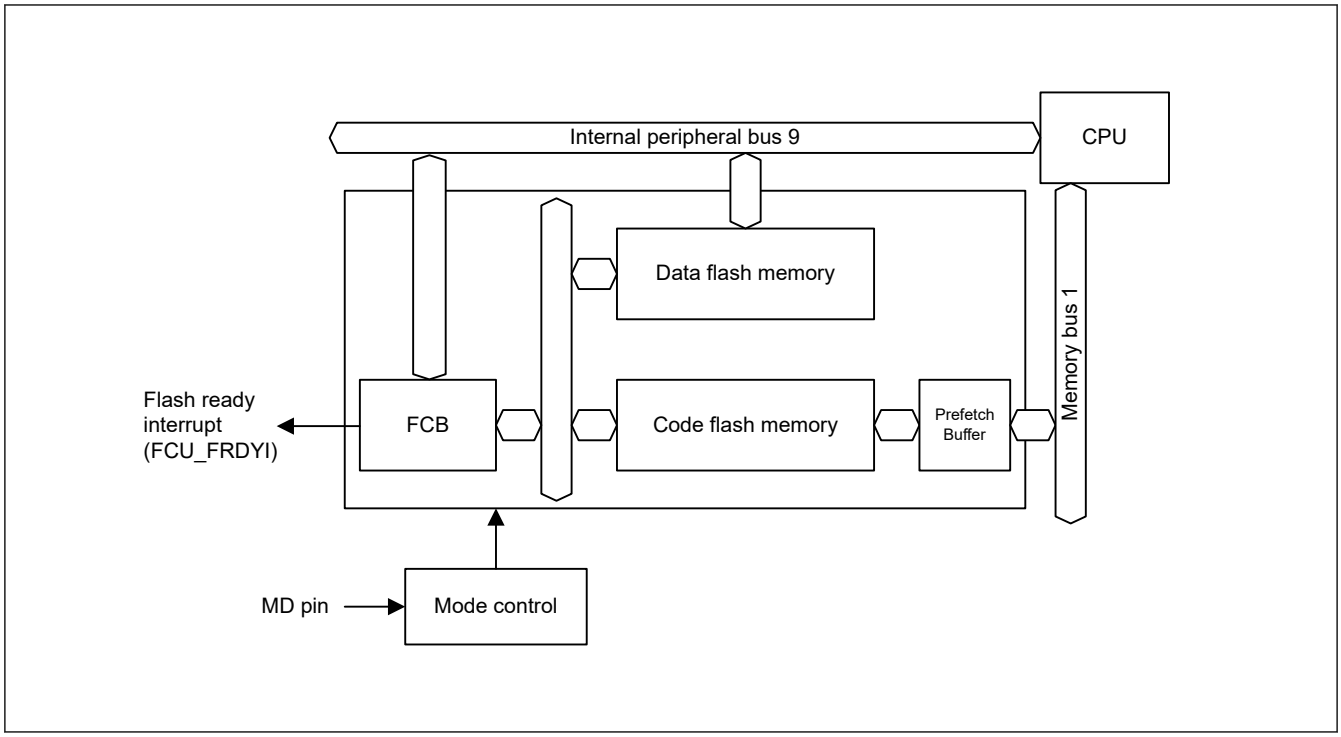


Figure 33.1 Flash memory-related modules block diagram

33.2 Memory Structure

Figure 33.2 shows the mapping of the code flash memory, and Table 33.2 shows the read and programming and erasure (P/E) addresses of the code flash memory. The user area of the code flash memory is divided into 2-KB blocks that serve as the units of erasure. The user area is available for storing the user program.

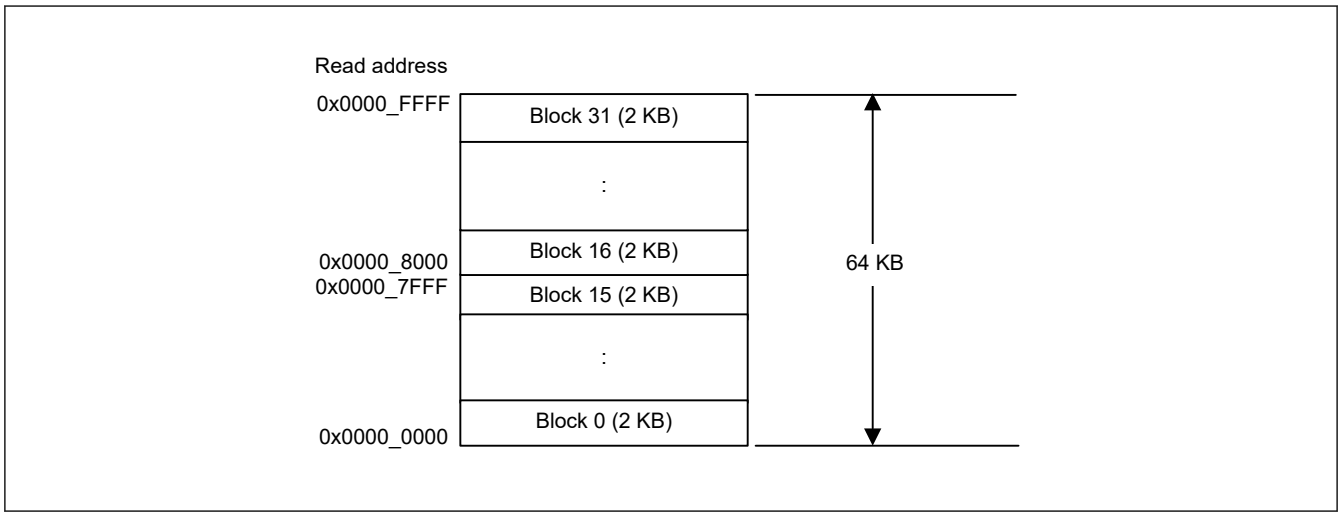


Figure 33.2 Mapping of the code flash memory

Table 33.2 Read and P/E addresses of the code flash memory

Size of code flash memory	Read address	P/E address	Number of blocks
64 KB	0x0000_0000 to 0x0000_FFFF	0x0000_0000 to 0x0000_FFFF	0 to 31

Figure 33.3 shows the mapping of the data flash memory, and Table 33.3 shows the read and programming and erasure (P/E) addresses of the data flash memory. The data area of the data flash memory is divided into 1-KB blocks, with each being a unit for erasure.

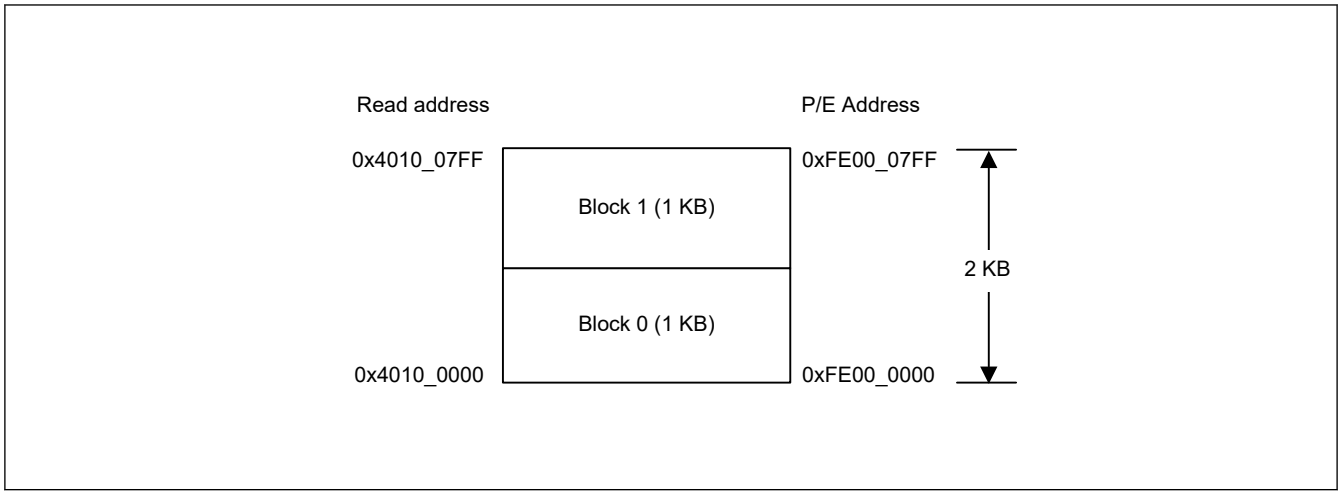


Figure 33.3 Mapping of the data flash memory

Table 33.3 Read and P/E addresses of the data flash memory

Size of data flash memory	Read address	P/E address	Number of blocks
2-KB	0x4010_0000 to 0x4010_07FF	0xFE00_0000 to 0xFE00_07FF	0, 1

33.3 Register Descriptions

33.3.1 DFLCTL : Data Flash Control Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0090

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

Bit	Symbol	Function	R/W
0	DFLEN	Data Flash Access Enable*1 0: Access to the data flash is disabled 1: Access to the data flash is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. It is necessary that DFLCTL.DFLEN bit is set to 1 before issuing the startup area information and security program, access window information program, and OCDID program command.

The DFLCTL register enables or disables accessing (reading, programming, and erasing) of the data flash. After setting the DFLCTL.DFLEN bit, Data Flash STOP recovery time (t_{DSTOP}) is necessary before reading the data flash or entering the data flash P/E mode.

33.3.2 PFBER : Prefetch Buffer Enable Register

Base address: FLCN = 0x407E_C000

Offset address: 0x3FC8

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

Bit	Symbol	Function	R/W
0	PFBE	Prefetch Buffer Enable bit 0: Prefetch buffer is disabled 1: Prefetch buffer is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

33.3.3 FENTRYR : Flash P/E Mode Entry Register

Base address: FLCN = 0x407E_C000

Offset address: 0x3FB0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FEKEY[7:0]								FENTRYD	—	—	—	—	—	—	FENTRY0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FENTRY0	Code Flash P/E Mode Entry 0 0: The code flash is the read mode 1: The code flash is the P/E mode.	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	FENTRYD	Data Flash P/E Mode Entry 0: The data flash is the read mode 1: The data flash is the P/E mode.	R/W
15:8	FEKEY[7:0]	Key Code	W

To program the code flash or the data flash, either the FENTRY0 or FENTRYD bit must be set to 1 to enter the P/E mode. Clearing the FENTRY0 bit or FENTRYD bit allows the code flash or data flash to be in read mode, but it is necessary to confirm the value of this bit before changing it. See section 33.13.1. Sequencer Modes.

FENTRY0 bit (Code Flash P/E Mode Entry 0)

[Setting condition]

- Set 0xAA01 to the FENTRYR register when it is 0x0000.

[Clearing conditions]

- Data is written by byte access
- A value other than 0xAA is set to the FEKEY[7:0] bits and written to the FENTRYR register
- Set 0xAA00 to the FENTRYR register
- Data is written to the FENTRYR register while the register has a value other than 0x0000.

FENTRYD bit (Data Flash P/E Mode Entry)

[Setting condition]

- Set 0xAA80 to the FENTRYR register when the register is 0x0000.

[Clearing conditions]

- Data is written by byte access.
- A value other than 0xAA is set to the FEKEY[7:0] bits and written to the FENTRYR register.
- Set 0xAA00 to the FENTRYR register.
- Data is written to the FENTRYR register while the register has a value other than 0x0000.

FEKEY[7:0] bits (Key Code)

The FEKEY[7:0] bits protect from unauthorized setting of FENTRY0 bit or FENTRYD bit.

Setting 0xAA to FEKEY[7:0] allows setting the FENTRY0 bit or the FENTRYD bit. The FEKEY[7:0] bits are read as 0x00.

33.3.4 FPR : Protection Unlock Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0180

Bit position:	7	6	5	4	3	2	1	0
Bit field:	FPR[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	FPR[7:0]	Protection Unlock This register is used to protect the FPMCR register from being rewritten inadvertently when the CPU runs out of control.	R/W

FPR[7:0] bits (Protection Unlock)

Writing to the FPMCR register is allowed only when the following procedure is used to access the register.

Procedure to unlock protection:

1. Write 0xA5 to the FPR register.
2. Write a set value to the FPMCR register
3. Write the inverted set value to the FPMCR register.
4. Write a set value to the FPMCR register again.

When a procedure other than the specified procedure is used to write data, the FPSR.PERR flag is set to 1.

33.3.5 FPSR : Protection Unlock Status Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0184

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

Bit	Symbol	Function	R/W
0	PERR	Protect Error Flag 0: No error 1: An error occurs	R
7:1	—	These bits are read as 0.	R

PERR bit (Protect Error Flag)

When the FPMCR register is not accessed as described in the procedure to unlock protection, data is not written to the register and this flag is set to 1.

[Setting condition]

- The FPMCR register is not accessed as described in the procedure to unlock protection described in section 33.3.4. FPR : Protection Unlock Register.

[Clearing conditions]

- The FPMCR register is accessed according to the procedure to unlock protection described in section 33.3.4. FPR : Protection Unlock Register.

33.3.6 FPMCR : Flash P/E Mode Control Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0100

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	FMS1	RPDIS	—	FMS0	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	FMS0	Flash Operating Mode Select 0 0: FMS1 = 0: Read mode FMS1 = 1: Data flash P/E mode. 1: FMS1 = 0: Code flash P/E mode FMS1 = 1: Setting prohibited.	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	RPDIS	Code Flash P/E Disable 0: Programming of the code flash is enabled 1: Programming of the code flash is disabled.	R/W
4	FMS1	Flash Operating Mode Select 1 See the description of the FMS0 bit.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The FPMCR register sets the operating mode of the flash memory and is protected from unauthorized setting.

See Figure 33.15 and Figure 33.17 for this register write control method.

See section 33.3.4. FPR : Protection Unlock Register for the procedure to unlock the protection.

FMS0 bit, FMS1 bits (Flash Operating Mode Select 0, Flash Operating Mode Select 1)

These bits set the operating mode of the flash memory.

[How to enter the code flash from the read mode to the code flash P/E mode]

Set FMS1 = 0, FMS0 = 1, and RPDIS = 0. Wait for the mode setup time t_{MS} (see section 35, Electrical Characteristics).

[How to enter the data flash from the read mode to the data flash P/E mode]

Set FMS1 = 1, FMS0 = 0, and RPDIS bit = 0.

[How to enter the code flash from the code flash P/E mode to the read mode]

Set FMS1 = 0, FMS0 = 0, and RPDIS = 1.

Wait for the read mode transition time (see section 35, Electrical Characteristics).

RPDIS bit (Code Flash P/E Disable)

RPDIS bit protects the code flash from unauthorized programming. Setting RPDIS bit to 0 allows the code flash to program.

33.3.7 FISR : Flash Initial Setting Register

Base address: FLCN = 0x407E_C000

Offset address: 0x01D8

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SAS[1:0]		PCKA[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	PCKA[5:0]	Flash-IF Clock Notification	R/W
7:6	SAS[1:0]	Startup Area Select 1 0: The startup area is switched to the default area temporarily 1 1: The startup area is switched to the alternate area temporarily. Others: The startup area is selected according to the settings of the extra area.	R/W

Note: Set or clear this register only in P/E mode. Additionally the SAS[1:0] bits are allowed to set or clear when the FSPR is 1. The FSPR bit is the protection flag of the access window and is stored in the extra area.

PCKA[5:0] bits (Flash-IF Clock Notification)

The hardware sequencer for the flash programming executes the commands according to the PCKA[5:0] bits. For this reason, it is necessary to set the PCKA[5:0] bits according to Flash-IF clock (ICLK) before execution of the programming and not during the programming.

Note: A wrong frequency setting may cause the flash macro to be damaged.

The following information describes how to set the PCKA[5:0] bits when the frequency is not an integral number, for example 31.5 MHz.

[When the frequency is higher than 4 MHz]

Set a rounded-up value for a non-integer frequency.

For example, set 32 MHz (PCKA = 011111b) when the frequency is 31.5 MHz.

[When the frequency is 4 MHz or lower]

Do not use a non-integer frequency. Use the frequency of 1, 2, 3, or 4 MHz.

Table 33.4 Frequency Settings

Flash-IF clock frequency [MHz]	PCKA[5:0]	Flash-IF clock frequency [MHz]	PCKA[5:0]	Flash-IF clock frequency [MHz]	PCKA[5:0]
64	101111b	48	100111b	32	011111b
24	010111b	20	010011b	19	010010b
18	010001b	17	010000b	16	001111b
15	001110b	14	001101b	13	001100b
12	001011b	11	001010b	10	001001b
9	001000b	8	000111b	7	000110b
6	000101b	5	000100b	4	000011b
3	000010b	2	000001b	1	000000b

SAS[1:0] bits (Startup Area Select)

The SAS[1:0] bits select the startup area. To change the startup area, the following methods can be used:

- When selecting the startup area according to the startup area settings of the extra area with the SAS[1:0] bits set to 00b or 01b, the startup area is selected accordingly. The settings are enabled after a reset is released.

- When switching the startup area to the default area temporarily with 10b written to the SAS[1:0] bits, the startup area is switched to the default area immediately after data is written to the register, regardless of the startup area settings of the extra area. When a reset is generated after this, the area is selected according to the startup area settings of the extra area.
- When switching the startup area to the alternative area temporarily with 11b written to the SAS[1:0] bits, the startup area is switched to the alternative area, regardless of the startup area settings of the extra area. When a reset is generated after this, the area is selected according to the startup area settings of the extra area.

33.3.8 FRESETR : Flash Reset Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0124

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

Bit	Symbol	Function	R/W
0	FRESET	Software reset of the registers 0: The registers related to the flash programming are not reset 1: The registers related to the flash programming are reset.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

FRESET bit (Software reset of the registers)

When this bit is set to 1, the FASR, FSARH, FSARL, FEARH, FEARL, FWBH0, FWBL0, FCR, and FEXCR registers are reset. Setting this bit to 0 allows the corresponding registers to be released from the reset state. Software commands are not allowed to execute while the FRESET bit is 1.

33.3.9 FASR : Flash Area Select Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0104

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

Bit	Symbol	Function	R/W
0	EXS	Extra Area Select 0: User area or data area 1: Extra area.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set or clear this register only in P/E mode.

EXS bit (Extra Area Select)

Set the EXS bit to 1 when programming the extra area using the FEXCR register. Set this bit to 0 when not programming the extra area.

33.3.10 FCR : Flash Control Register

Base address: FLCN = 0x407E_C000

Offset address: 0x0114

Bit position:	7	6	5	4	3	2	1	0
Bit field:	OPST	STOP	—	DRC	CMD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CMD[3:0]	Software Command Setting 0x1: Program 0x3: Blank check (code flash) 0x4: Block erase 0x5: Consecutive read 0x6: Chip erase 0xB: Blank check (data flash) Others: Setting prohibited*1.	R/W
4	DRC	Data Read Completion 0: Data is not read or next data is requested 1: Data reading is complete.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	STOP	Forced Processing Stop When this bit is set to 1, the processing being executed can be forcibly stopped.	R/W
7	OPST	Processing Start 0: Processing stops 1: Processing starts.	R/W

Note: Set or clear this register only in P/E mode. Additionally it is not allowed to be reset by the FRESETR register while the software command is being executed.

Note 1. This does not include writing 0x00 to the FCR register when the FSTATR1.FRDY bit is 1.

CMD[3:0] bits (Software Command Setting)

The following information describes the function of each software command.

[Program]

Writes data of the FWBH0 and FWBL0 registers to the flash macro to the address pointed by the FSARH and FSARL registers.

[Blank check]

Verifies whether the flash macro is the blank state (not to be programmed) from the start address pointed by the FSARH and FSARL registers to the end address pointed by the FEARH and FEARL registers. The blank check command is allowed to execute within the region of flash macro.

Note: The blank check result cannot guarantee that the flash memory is erased.

[Block erase]

Erases block of the flash memory.

Set the start address of the target erasure block in the FSARH and FSARL registers, and set the end address of the target erasure block in the FEARH and FEARL registers. If a setting other than the specified is made, erasure may not be executed correctly. The block erase command is allowed to execute within the region of flash macro.

[Consecutive read]

Reads the flash macro from the start address pointed by the FSARH and FSARL registers to the end address pointed by the FEARH and FEARL registers. The read data is stored in the FRBH and FRBL registers. The consecutive read command is allowed to execute within the region of the flash macros.

[Chip erase]

Erases all blocks of the flash macro

Set the start address of the target erasure block in the FSARH and FSARL registers, and set the end address of the target erasure block in the FEARH and FEARL registers. If a setting other than the specified is made, erasure may not be executed correctly.

DRC bit (Data Read Completion)

After executing the consecutive read command and reading the FRBH and FRBL registers, writing 1 to the DRC bit completes the processing for read data. Writing 0 to the DRC bit starts reading the next data.

STOP bit (Forced Processing Stop)

The STOP bit stops the execution of the erase command or the blank check command.

After setting 1 to the STOP bit, it is necessary to wait until the FSTATR1.FRDY bit becomes 1 (processing completed) before setting the OPST bit to 0.

OPST bit (Processing Start)

The OPST bit starts the command set for the CMD[2:0] bits. Setting the OPST bit to 0 terminates the execution of the command after the FRDY bit of the FSTATR1 register becomes 1, and is required to confirm that the FRDY bit is 0.

Note:
- Commands cannot be executed when the ID authorization for the flash programmer has failed.
- The program, the block erase, and the read commands cannot be executed when the address of each command points to an area that is protected by the access window.

33.3.11 FEXCR : Flash Extra Area Control Register

Base address: FLCN = 0x407E_C000

Offset address: 0x01DC

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

Bit	Symbol	Function	R/W
2:0	CMD[2:0]	Software Command Setting 0 1 0: Access window information program Startup area selection and security setting 0 1 1: OCDID1 program 1 0 0: OCDID2 program 1 0 1: OCDID3 program 1 1 0: OCDID4 program Others: Setting prohibited*1.	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	OPST	Processing Start 0: Processing stops 1: Processing starts.	R/W

Note: Set or clear this register only in P/E mode. Additionally it is not allowed to be reset by the FRESETR register while the software command is being executed.

Note 1. This does not include writing 0x00 to the FEXCR register when the FSTATR1.EXRDY bit is 1.

The FEXCR register programs the extra area. Before execution of each command, it is necessary to set the FWBL0 and FWBH0 registers.

When programming using the FEXCR register, the programming area is erased automatically before execution, therefore it is not necessary to erase beforehand.

CMD[2:0] bits (Software Command Setting)

The CMD[2:0] bits select the software command from the:

- Startup area selection and security setting

- Access window information program
- OCDID program.

The following information describes the function of each software command.

[Startup area selection and security setting]

Setting data to the FWBL0/FWBH0 registers, this command is allowed to select the startup area from the default area (8 KB) to the alternative area (next 8 KB) and set the security. For details, see section 33.9.1. Startup Program Protection.

Bit [15] of the FWBH0 register is 0 and the alternative area (next 8 KB) is selected as the startup area.

Bit [15] of the FWBH0 register is 1 and the default area (8 KB) is selected as the startup area.

Bit [15] of the FWBL0 register is 0.

- The access window cannot be updated because the access window information program command cannot be executed.
- The startup area cannot be changed.
- Data of the SAS bits of the FISR register cannot be changed.

Note: The startup area selection and security setting command cannot be set to 1 for the corresponding bit of the extra area after 0 is set.

The following information describes mapping for the extra bit of the startup area selection and security setting.

Table 33.5 Mapping for the extra bit of the startup area selection and security setting (address (P/E) : 0x0000_0010)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SASMF*1	—	—	—	—	FAWE[10:0]										
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FSPR*1	—	—	—	—	FAWS[10:0]										

Note 1. Once 0 is set for these bits, it cannot be changed to 1.

[Access window information program]

This command sets the access window used for area protection. The program command, block erase command, and consecutive read command of the protected area cannot be executed. The chip erase command cannot be executed when the access window is set (the start block address of the access window is not equal to the end block address). It is necessary to set the start block address of the access window to the FWBL0 register bits [10:0] and the next block address of the end block address of the access window to the FWBH0 register bits [10:0] before the execution of the access window information program command. When the start address and the end address are set to the same value, all areas of the code flash can be accessed. When the start address is larger than the end block address, all areas of the code flash cannot be accessed.

The FWBL0[10] bit for the start block address must be set to 0 when the access window is set (the end block address of the access window is larger than the start block address).

The following information describes mapping for the extra bit of the access window information program.

Table 33.6 Mapping for the extra bit of the access window information program (address (P/E) : 0x0000_0010)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SASMF*1	—	—	—	—	FAWE[10:0]										
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
FSPR*1	—	—	—	—	FAWS[10:0]										

Note 1. Once 0 is set as data in these bits, it cannot be changed to 1.

[OCDID1-4 program]

These commands set the OCDID[127:0] bits.

Table 33.7 OCDID Settings

Command	OCDID	FWBH0	FWBL0
OCDID1 program	OCDID [31:0]	OCDID [31:16]	OCDID [15:0]
OCDID2 program	OCDID [63:32]	OCDID [63:48]	OCDID [47:32]
OCDID3 program	OCDID [95:64]	OCDID [95:80]	OCDID [79:64]
OCDID4 program	OCDID [127:96]	OCDID [127:112]	OCDID [111:96]

The following information describes mapping for the extra bit of OCDID1-4 program.

Table 33.8 Mapping for the extra bit of OCDID1-4 program (address (P/E) : 0x0000_0018)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OCDID[31:16]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OCDID[15:0]															

Table 33.9 Mapping for the extra bit of OCDID1-4 program (address (P/E) : 0x0000_0020)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OCDID[63:48]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OCDID[47:32]															

Table 33.10 Mapping for the extra bit of OCDID1-4 program (address (P/E) : 0x0000_0028)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OCDID[95:80]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OCDID[79:64]															

Table 33.11 Mapping for the extra bit of OCDID1-4 program (address (P/E) : 0x0000_0030)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
OCDID[127:112]															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
OCDID[111:96]															

OPST bit (Processing Start)

The OPST bit starts the command set for the CMD[2:0] bits. Setting the OPST bit to 0 terminates the execution of the command after the EXRDY bit of the FSTATR1 register becomes 1, and is necessary to confirm that the EXRDY bit is 0.

33.3.12 FSARH : Flash Processing Start Address Register H

Base address: FLCN = 0x407E_C000

Offset address: 0x0110

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

Bit	Symbol	Function	R/W
15:0	FSARH[15:0]	Flash Processing Start Address H Flash Processing Start Address upper 16 bits See FSARL for details.	R/W

Note: Set or clear this register only in P/E mode. The write value should be 0 for b8 to b5, and those bits are read as 0.

33.3.13 FSARL : Flash Processing Start Address Register L

Base address: FLCN = 0x407E_C000

Offset address: 0x0108

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

Bit	Symbol	Function	R/W
15:0	FSARL[15:0]	Flash Processing Start Address L Flash processing start address lower 16 bits	R/W

Note: Set or clear this register only in P/E mode.

The FSARH and FSARL registers set the start address of the software command. When the FSARH and FSARL registers are read while executing a software command set by the FEXCR register, an undefined value is read. After execution of the program command, the sequencer of the software command increments data automatically. The auto increment function of the program command discards the setting of the next address to the FSARH and FSARL registers when the next address is a consecutive address. The increment unit is as follows:

Code flash: +0x4

Data flash: +0x1

See Figure 33.2 and Figure 33.3 for details on the addresses of the flash memory.

33.3.14 FEARH : Flash Processing End Address Register H

Base address: FLCN = 0x407E_C000

Offset address: 0x0120

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

Bit	Symbol	Function	R/W
15:0	FEARH[15:0]	Flash Processing End Address H Flash processing end address upper 16 bits See FEARL for details.	R/W

Note: Set or clear this register only in P/E mode. The write value should be 0 for b8 to b5, and those bits are read as 0.

33.3.15 FEARL : Flash Processing End Address Register L

Base address: FLCN = 0x407E_C000

Offset address: 0x0118

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

Bit	Symbol	Function	R/W
15:0	FEARL[15:0]	Flash Processing End Address L Flash processing end address lower 16 bits	R/W

Note: Set or clear this register only in P/E mode.

The FEARH and FEARL registers set the end address of the blank check, the block erase, the chip erase, and the consecutive read command. When the FEARH and FEARL registers are read while executing a software command set by the FEXCR register, an undefined value is read.

See Figure 33.2 and Figure 33.3 for details on the addresses of the flash memory.

33.3.16 FWBL0 : Flash Write Buffer Register L0

Base address: FLCN = 0x407E_C000

Offset address: 0x0130

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

Bit	Symbol	Function	R/W
15:0	WDATA[15:0]	Flash Write Buffer L0 Flash write buffer data lower 16 bits See FWBH0 for details.	R/W

Note: Set or clear this register only in P/E mode.

33.3.17 FWBH0 : Flash Write Buffer Register H0

Base address: FLCN = 0x407E_C000

Offset address: 0x0138

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

Bit	Symbol	Function	R/W
15:0	WDATA[15:0]	Flash Write Buffer H0 Flash write buffer data upper 16 bits	R/W

Note: Set or clear this register only in P/E mode.

The FWBH0 and FWBL0 registers set program data of the program command, the startup selection and security setting command, the access window information program command, and the OCDID program command. The following table describes how to set data according to each command.

Register	What is set to the register
FWBH0 FWBL0	• Bits [31:0] of the programming data of the program command for the code flash • Bits [7:0] of the programming data of the program command for the data flash • Bits [31:0] of the programming data of the startup selection and security setting command, the access window information program command, and the OCDID program command.

33.3.18 FRBL0 : Flash Read Buffer Register L0

Base address: FLCN = 0x407E_C000

Offset address: 0x0188

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

Bit	Symbol	Function	R/W
15:0	RDATA[15:0]	Flash Read Buffer L0 The RDATA[15:0] store bits [15:0] of the read data of the code flash or data flash read when the consecutive read command is executed. When the data flash is read, 0x00 is stored to bits [15:8].	R

33.3.19 FRBH0 : Flash Read Buffer Register H0

Base address: FLCN = 0x407E_C000

Offset address: 0x0190

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

Bit	Symbol	Function	R/W
15:0	RDATA[15:0]	Flash Read Buffer H0 RDATA[15:0] store bits [31:16] of the read data of the code flash when the consecutive read command is executed.	R

33.3.20 FSTATR1 : Flash Status Register 1

Base address: FLCN = 0x407E_C000

Offset address: 0x012C

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

Bit	Symbol	Function	R/W
0	—	This bit is read as 0.	R
1	DRRDY	Data Read Ready Flag 0: The read processing of the consecutive read command at each address is not terminated. 1: The read processing of the consecutive read command at each address is terminated and read data is stored to the FRBH and FRBL registers.	R
2	—	This bit is read as 1.	R

Bit	Symbol	Function	R/W
5:3	—	These bits are read as 0.	R
6	FRDY	Flash Ready Flag 0: The software command of the FCR register is not terminated. 1: The software command of the FCR register is terminated.	R
7	EXRDY	Extra Area Ready Flag 0: The software command of the FEXCR register is not terminated. 1: The software command of the FEXCR register is terminated.	R

FSTATR1 is a status register used to confirm the execution result of a software command. Each flag is set to 0 when the next software command is executed.

33.3.21 FSTATR2 : Flash Status Register 2

Base address: FLCN = 0x407E_C000

Offset address: 0x01F0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	EILGLERR	ILGLERR	BCERR	PRGERR01	PRGERR	ERERR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ERERR	Erase Error Flag 0: Erasure terminates normally 1: An error occurs during erasure	R
1	PRGERR	Program Error Flag 0: Programming terminates normally 1: An error occurs during programming.	R
2	PRGERR01	Program Error Flag 01 0: Programming by the FEXCR register terminates normally 1: An error occurs during programming.	R
3	BCERR	Blank Check Error Flag 0: Blank checking terminates normally 1: An error occurs during blank checking.	R
4	ILGLERR	Illegal Command Error Flag 0: No illegal software command or illegal access is detected 1: An illegal command or illegal access is detected.	R
5	EILGLERR	Extra Area Illegal Command Error Flag 0: No illegal command or illegal access to the extra area is detected 1: An illegal command or illegal access to the extra area is detected.	R
15:6	—	These bits are read as 0.	R

FSTATR2 is a status register used to confirm the execution result of a software command. Each error flag is set to 0 when the next software command is executed.

ERERR flag (Erase Error Flag)

The value of the ERERR bit is undefined when the FCR.STOP bit is set to 1 (processing is forcibly stopped) during erasure.

PRGERR flag (Program Error Flag)

The PRGERR bit is set when the program command of the FCR register or each command of the FEXCR register is abnormally terminated.

PRGERR01 flag (Program Error Flag 01)

The PRGERR01 bit is set when each command of the FEXCR register is abnormally terminated.

ILGLERR flag (Illegal Command Error Flag)

The ILGLERR flag indicates the execution of the software command of the FCR register with unexpected condition.

[Setting condition]

- Programming/erasure/read commands are executed to an area protected by the access window range
- The chip erase command is executed when the access window is set (the start block address of the access window is not equal to the end one)
- The blank check, the block erase, consecutive read, and the chip erase commands are executed when the start address set to the FSARH and FSARL registers is larger than the end address set to the FEARH and FEARL registers
- The program, the block erase, the chip erase, and the blank check commands are executed when the FASR.EXS bit is 1
- The data flash address is set to the FSARH and FSARL registers and a software command is executed in the code flash P/E mode
- The code flash address is set to the FSARH and FSARL registers and a software command is executed in the data flash P/E mode
- The code flash and the data flash are set to P/E mode simultaneously and a software command is executed.

[Clearing conditions]

- The next software command is executed.

EILGLERR flag (Extra Area Illegal Command Error Flag)

The EILGLERR flag indicates the execution of the software command of the FEXCR register with unexpected condition.

[Setting condition]

- The software commands of the FEXCR register is executed when the EXS bit of the FASR register is 0
- The access window information program command is executed when the FSPR bit is 0

[Clearing conditions]

- The next software command is executed.

33.3.22 FEAMH : Flash Error Address Monitor Register H

Base address: FLCN = 0x407E_C000

Offset address: 0x01E8

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

Bit	Symbol	Function	R/W
15:0	FEAMH[15:0]	Flash Error Address Monitor Register H Flash error address monitor upper 16 bits See FEAML for details.	R/W

33.3.23 FEAML : Flash Error Address Monitor Register L

Base address: FLCN = 0x407E_C000

Offset address: 0x01E0

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

Bit	Symbol	Function	R/W
15:0	FEAML[15:0]	Flash Error Address Monitor Register L Flash error address monitor lower 16 bits	R/W

The error address is withdrawn from the FEAMH and FEAML registers after a software command execution. See Figure 33.2 and Figure 33.3 for details on the addresses of the flash memory.

33.3.24 FSCMR : Flash Start-Up Setting Monitor Register

Base address: FLCN = 0x407E_C000

Offset address: 0x01C0

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

Note 1. The reset value depends on the state of the extra area.

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0.	R
8	SASMF	Startup Area Setting Monitor Flag 0: Setting to start up using the alternative area 1: Setting to start up using the default area	R
10:9	—	The read values are undefined.	R
11	—	This bit is read as 0.	R
13:12	—	The read values are undefined.	R
14	FSPR	Access Window Protection Flag 0: Access window setting disabled. 1: Access window setting enabled.	R
15	—	This bit is read as 0.	R

The FSCMR register monitors the extra area setting. Data of this register is updated at the reset sequence or execution of the software command of the FEXCR register.

33.3.25 FAWSMR : Flash Access Window Start Address Monitor Register

Base address: FLCN = 0x407E_C000

Offset address: 0x01C8

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FSPR	—	—	—	—	FAWS[10:0]										
Value after reset:	0	0	0	0	0	Value set by the user*1										

Note 1. The value of the blank product is 1. It is set to the same value set in bits [10:0] in the FWBH0 register after the access window information program command is executed.

Bit	Symbol	Function	R/W
10:0	FAWS[10:0]	Access Window Start Address This register is used to confirm the set value of the access window start address used for area protection	R
14:11	—	These bits are read as 0.	R
15	FSPR	Access Window Protection Flag This bit has the same value as the FSPR bit of the FSCMR register.	R

33.3.26 FAWEMR : Flash Access Window End Address Monitor Register

Base address: FLCN = 0x407E_C000

Offset address: 0x01D0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SASMF	—	—	—	—	FAWE[10:0]										
Value after reset:	0	0	0	0	0	Value set by the user*1										

Note 1. The value of the blank product is 1. It is set to the same value set in bits [10:0] in the FWBL0 register after the access window information program command is executed.

Bit	Symbol	Function	R/W
10:0	FAWE[10:0]	Access Window End Address This register is used to confirm the set value of the access window end address used for area protection	R
14:11	—	These bits are read as 0.	R
15	SASMF	Startup Area Setting Monitor Flag This bit has the same value as the SASMF bit of the FSCMR register.	R

33.3.27 UIDRn : Unique ID Registers n (n = 0 to 3)

Address: 0x0100_1C00 + n × 4

Bit position:	31	0
Bit field:	UID[31:0]	
Value after reset:	Unique value for each chip	

Bit	Symbol	Function	R/W
31:0	UID[31:0]	Unique ID	R

The UIDRn is a read-only register that stores a 16-byte ID code (unique ID) for identifying the individual MCU. The UIDRn register should be read in 32-bit units.

33.3.28 PNRn : Part Numbering Register n (n = 0 to 3)

Address: 0x0100_1C10 + n × 4

Bit position:	31	0
Bit field:	PNR[31:0]	
Value after reset:	Unique value for each chip	

Bit	Symbol	Function	R/W
31:0	PNR[31:0]	Part Number	R

The PNRn is a read-only register that stores a 16-byte part numbering. The PNRn register should be read in 32-bit units. Each byte corresponds to the ASCII code representation of the product part number as detailed in product list.

In case of the part number is ’R7FA2T1073CFL’, 16-byte part numbering is stored as follows.

Address 0x0100_1C10: ’L’, 0x4C in ASCII code

Address 0x0100_1C11: ’F’, 0x46 in ASCII code

Address 0x0100_1C12: ’C’, 0x44 in ASCII code

Address 0x0100_1C13: ’3’, 0x32 in ASCII code

Address 0x0100_1C14: ’7’, 0x37 in ASCII code

Address 0x0100_1C15: ’0’, 0x30 in ASCII code

Address 0x0100_1C16: ’1’, 0x33 in ASCII code

Address 0x0100_1C17: ’T’, 0x45 in ASCII code

Address 0x0100_1C18: ’2’, 0x32 in ASCII code

Address 0x0100_1C19: ’A’, 0x41 in ASCII code

Address 0x0100_1C1A: ’F’, 0x46 in ASCII code

Address 0x0100_1C1B: ’7’, 0x37 in ASCII code

Address 0x0100_1C1C: ’R’, 0x52 in ASCII code

Address 0x0100_1C1D: ”(space) , 0x20 in ASCII code

Address 0x0100_1C1E: ”(space) , 0x20 in ASCII code

Address 0x0100_1C1F: ”(space) , 0x20 in ASCII code

33.3.29 MCUVER : MCU Version Register

Address: 0x0100_1C20

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MCUVE[7:0]							
Value after reset:	Value depend on the chip							

Bit	Symbol	Function	R/W
7:0	MCUVE[7:0]	MCU Version	R

The MCUVER is a read-only register that stores an MCU version. The MCUVER register should be read in 8-bit units. The higher the value, the newer MCU version.

33.4 Instruction Prefetch from Flash Memory

Flash memory provides an instruction prefetch function to accelerate code execution. The prefetch function can be used by enabling the prefetch buffer. To enable the prefetch buffer, set the PFBER.PFBE bit to 1.

Note: When Flash memory is in the program or erase operation, the PFBER.PFBE bit should be set to 0 beforehand.

33.5 Operating Modes Associated with the Flash Memory

Figure 33.4 shows a diagram of the mode transitions associated with the flash memory. For information on setting up the modes, see section 3, Operating Modes.

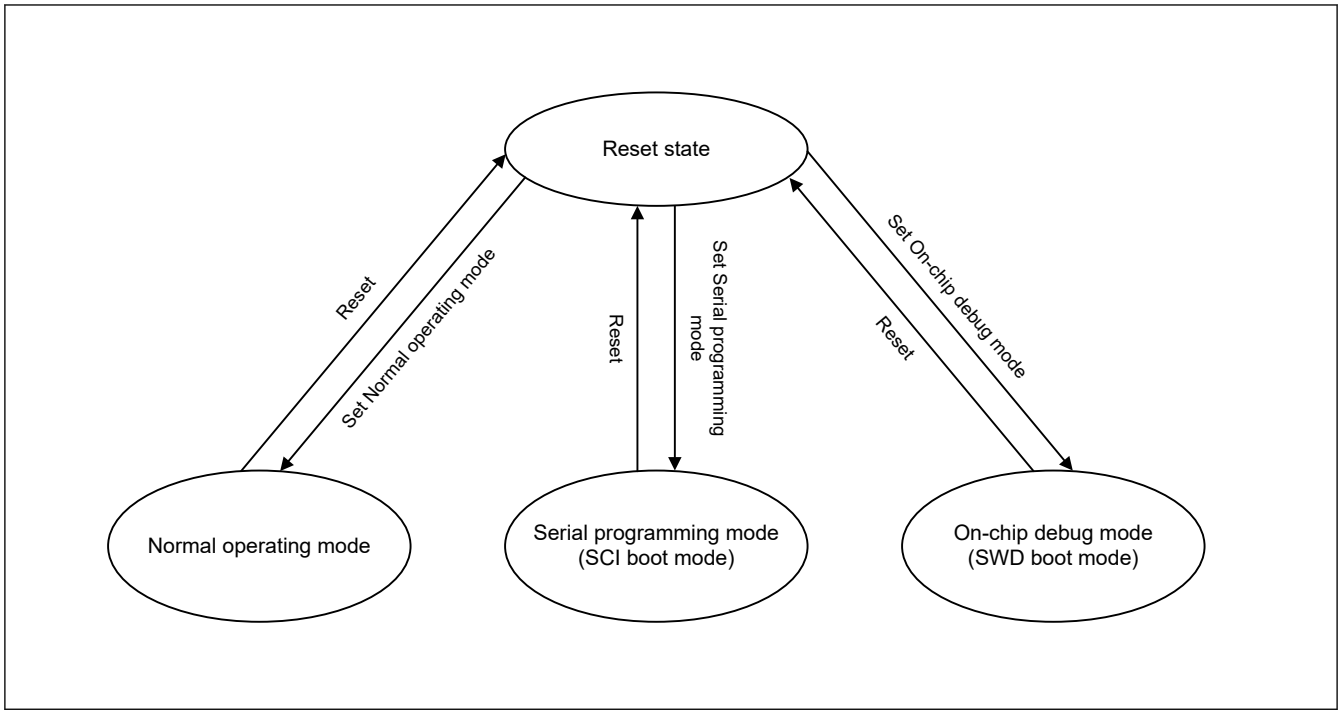


Figure 33.4 Mode transitions associated with flash memory

The flash memory areas where programming and erasure are permitted and where the boot program executes at a reset, differ with the mode. Table 33.12 shows the differences between the modes.

Table 33.12 Difference between modes

Parameter	Normal operating mode	Serial programming mode (SCI boot mode)	On-chip debug mode (SWD boot mode)
Programmable and erasable areas	• Code flash memory • Data flash memory.	• Code flash memory • Data flash memory.	• Code flash memory • Data flash memory.
Erasure in block units	Possible	Possible	Possible
Boot program at a reset	User area program	Embedded program for serial programming	Depends on debug command

33.5.1 ID Code Protection

The ID code protection function prohibits programming and on-chip debugging. When ID code protection is enabled, the device validates or invalidates the ID code sent from the host by comparing it with the ID code stored in the flash memory. Programming and on-chip debugging are enabled only when the two match.

The ID code in flash memory consists of four 32-bit words. ID code bits [127] and [126] determine whether ID code protection is enabled and the authentication method to use with the host. Table 33.13 shows how the ID code determines the authentication method.

Table 33.13 Specifications for ID code protection

Operating mode on boot up	ID code	State of protection	Operations on connection with the programmer or on-chip debugger
Serial programming mode (SCI boot mode) On-chip debug mode (SWD boot mode)	0xFF, …, 0xFF (all bytes 0xFF)	Protection disabled	The ID code is not checked, the ID code always matches, and the connection to the serial programmer or on-chip debugger*1 is permitted.
	Bit [127] = 1, bit [126] = 1, and at least one of all 16 bytes is not 0xFF	Protection enabled	Matching ID code indicates that authentication is complete and connection to the serial programmer or the on-chip debugger is permitted. Mismatching ID code indicates transition to the ID code protection wait state. When the ID code sent from the serial programmer or the on-chip debugger is ALeRASE in ASCII code (0x414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFF), the content of the user flash area is erased and all bits in the OSIS register are 1. However, when the AWS.FSPR bit is 0 or security MPU is enabled, the content of the user flash area is not erased.
	Bit [127] = 1 and bit [126] = 0	Protection enabled	Matching ID code indicates that authentication is complete and connection to the serial programmer or the on-chip debugger is permitted. Mismatching ID code indicates transition to the ID code protection wait state.
	Bit [127] = 0	Protection enabled	The ID code is not checked, the ID code is always mismatching, the connection to the serial programmer or the on-chip debugger is prohibited. When the ID code sent from the on-chip debugger is ALeRASE in ASCII code (0x414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFF), the content of the user flash area is erased and all bits in the OSIS register are 1. However, when the AWS.FSPR bit is 0 or security MPU is enabled, the content of the user flash area is not erased.

Note 1. Never send the ID code from on-chip debugger. Or send ID code 0xFF, …, 0xFF (all bytes 0xFF) from on-chip debugger.

33.6 Overview of Functions

By using a dedicated flash-memory programmer to program the on-chip flash memory through a serial interface (serial programming mode) or through SWD interface (on-chip debug mode), the device can be programmed before or after it is mounted on the target system. Additionally, security functions to prohibit overwriting of the user program prevent tampering by third parties.

Programming by the user program (self-programming) is available for applications that might require updating after system manufacturing or shipment. Protection features for safely overwriting the flash memory area are also provided. Additionally, interrupt processing during self-programming is supported so that programming can proceed while processing external communications and other functions. Table 33.14 lists the programming methods and the associated operating modes.

Table 33.14 Programming methods

Programming method	Functional overview	Operating mode
Serial programming	A dedicated flash-memory programmer connected through the SCI interface can program the on-board flash memory after the device is mounted on the target system.	Serial programming mode
	A dedicated flash-memory programmer connected through the SCI interface and a dedicated programming adapter board allow off-board programming of the flash memory, before it is mounted on the target system.	
Self-programming	A user program written to memory in advance of serial programming execution can also program the flash memory. The background operation capability makes it possible to fetch instructions or otherwise read data from code flash memory while the data flash memory is programming. As a result, a program resident in code flash memory can program data flash memory.	Normal operating mode
SWD programming	A dedicated flash-memory programmer or an on-chip debugger connected through SWD can program the on-board flash memory after the device is mounted on the target system.	On-chip debug mode
	A dedicated flash-memory programmer or an on-chip debugger connected through SWD and a dedicated programming adapter board allow off-board programming of the flash memory, before it is mounted on the target system.	

Table 33.15 lists the functions of the on-chip flash memory. Use serial programmer commands for serial programming. For self-programming, use the programming commands to read the on-chip flash memory or run the user program.

Table 33.15 Basic functions

Function	Functional overview	Availability	
		Serial programming	Self-programming
Blank check	Checks a specified block to ensure that writing to it has not already proceeded.	Not supported	Supported
Block erasure	Erases the memory contents in the specified block	Supported	Supported
Programming	Writes to the specified address	Supported	Supported
Read	Reads data programmed in the flash memory	Supported	Not supported (read by user program is possible)
ID code check	Compares the ID code sent by the host with the code stored in the code flash memory. If the two match, the FCB enters the wait state for programming and erasure commands from the host.	Supported	Not supported (ID authentication is not performed)
Security configuration	Configures the protection of security function (Access window and Start-up area selection)	Supported with conditions (see section 33.9. Protection)	Supported with conditions (see section 33.9. Protection)
Protection configuration	Configures the access window for flash area protection in the code flash memory	Supported	Supported

The on-chip flash memory supports the ID code check function. Authentication of ID code check is a security function for use with serial programming and with SWD programming. Table 33.16 lists the security functions supported by the on-chip flash memory, and Table 33.17 lists the available operations and security settings.

Table 33.16 Security functions

Function	Description
ID authentication	The result of ID authentication can be used to control the connection of a serial programmer for serial programming

Table 33.17 Available operations and security settings

Function	All security settings and erasure, programming, and read operations		Constraints on the security setting configuration
	Serial programming and on-chip debug mode	Self-programming mode	
ID authentication	When the ID codes do not match: ● Block erasure commands: not supported ● Programming commands: not supported ● Read commands: not supported ● Security configuration commands: not supported ● Protection configuration commands: not supported. When the ID codes match: ● Block erasure commands: supported ● Programming commands: supported ● Read commands: supported ● Security configuration commands: supported ● Protection configuration commands: supported.	● Blank check: supported ● Block erasure: supported ● Programming: supported ● Security configuration: supported ● Protection configuration: supported.	**Self-programming mode** ID authentication is not performed

33.6.1 Configuration Area Bit Map

The bits used for ID authentication, startup area select, access window protection, and security configuration functions are mapped in Figure 33.5. The boot program must use these bits as hexadecimal data.

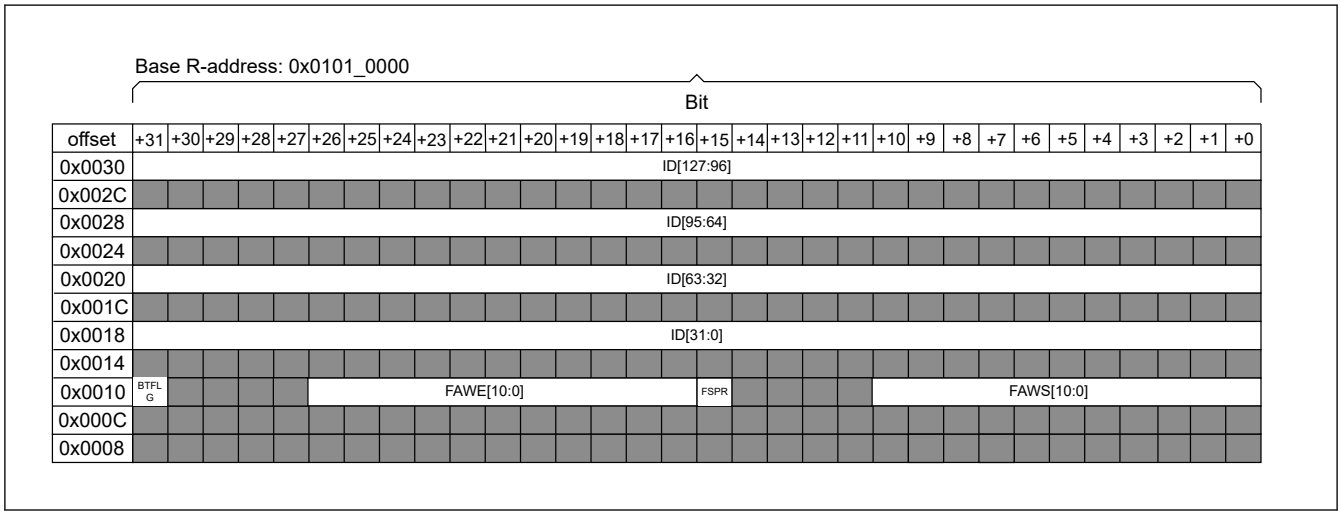
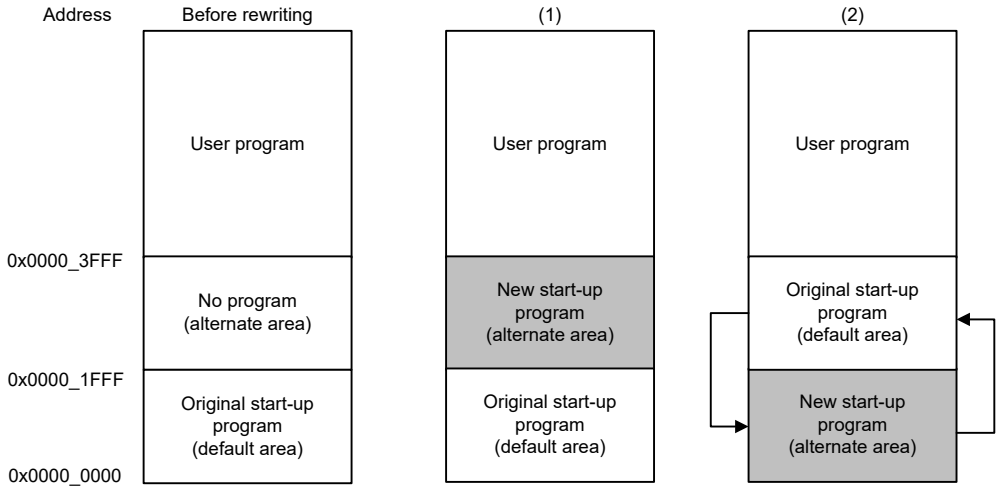


Figure 33.5 Configuration area bit map

33.6.2 Startup Area Select

The startup area select function allows the boot program to be safely updated. The startup area is 8 KB of space located in the user area. The FCB controls the address of the startup area based on the Startup Area Select Flag (BTFLG) that is located in the configuration area which names as AWS register. The startup area can be locked by the FSPR bit.

Figure 33.6 shows an overview of the startup program protection.



(1) Program a new startup program in the alternate area. If the alternate area fails to be rewritten, the new startup program can be rewritten again after starting up using the default area, because the original startup program is in the default area.
(2) After the alternate area is successfully rewritten, the default area and the alternate area are switched using the self-programming library. After that, the program in the alternate area starts after a reset.

Figure 33.6 Overview of startup program protection

33.6.3 Protection by Access Window

Issuing the program or block erase command to a flash memory area outside of the access window results in the command-locked state. The access window is only valid in the user area of the code flash memory. The access window provides protection in self-programming, serial programming, and on-chip debug modes. Figure 33.7 shows an overview of flash area protection.

The access window is specified in both the FAWS [10:0] and FAWE [10:0] bits. See section 6.2.4. AWS : Access Window Setting Register. Setting of the FAWE[10:0] and FAWS[10:0] bits in various conditions is described as follows:

- FAWE [10:0] = FAWS [10:0]: The P/E command can execute anywhere in the user area of the code flash memory
- FAWE [10:0] > FAWS [10:0]: The P/E command can only execute in the window from the block pointed to by the FAWS bits to one block lower than the block pointed to by the FAWE[10:0] bits
- FAWE [10:0] < FAWS [10:0]: The P/E command cannot execute anywhere in the user area of the code flash memory.

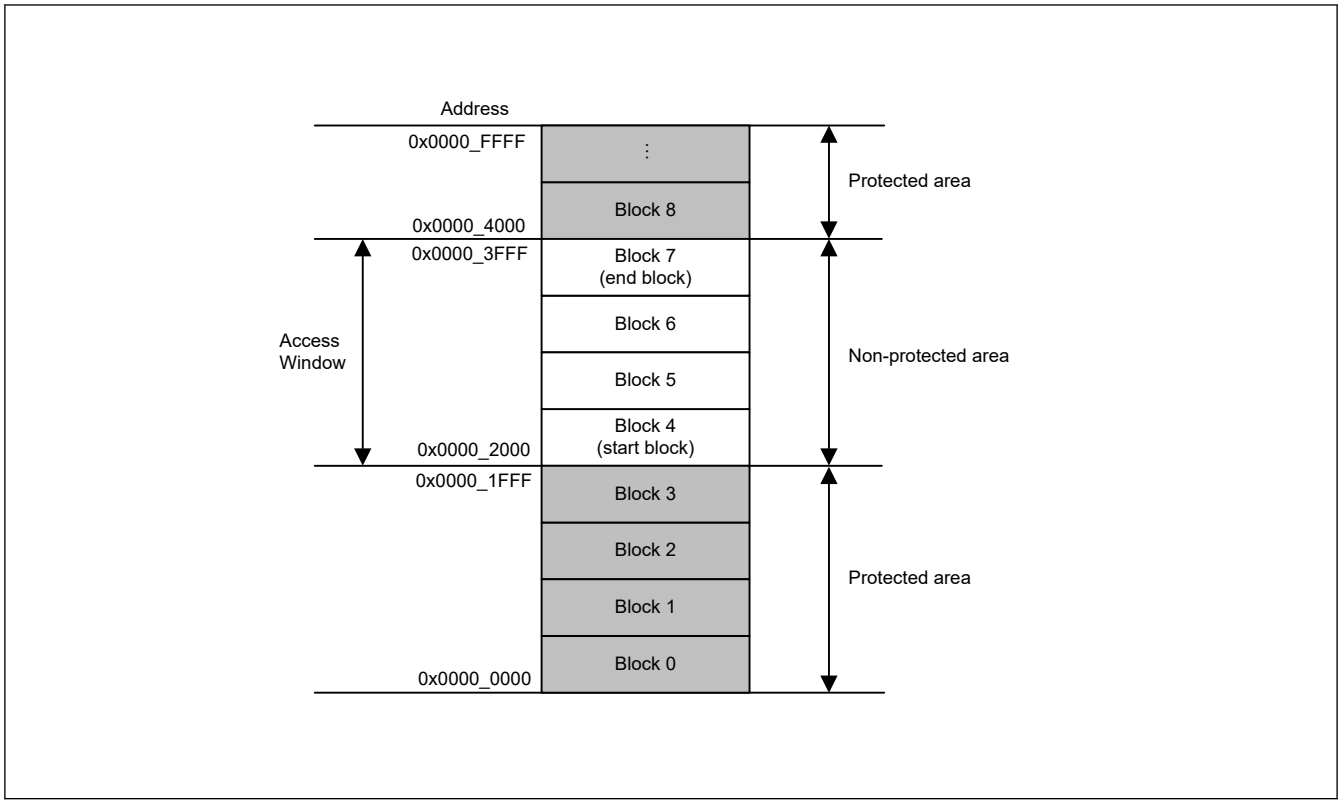


Figure 33.7 Flash area protection overview

33.7 Programming Commands

The FCB controls the programming commands.

33.8 Suspend Operation

The forced stop command forces the blank check command, the block erase command, or the chip erase command to stop. When a forced stop is executed, the stopped address values are stored in the registers. The command can restart from the stopped address after a reset to the registers for command execution by copying the saved addresses.

If a forced stop is executed during chip erase command, execute the chip erase command again and then restart.

33.9 Protection

The types of protection provided include:

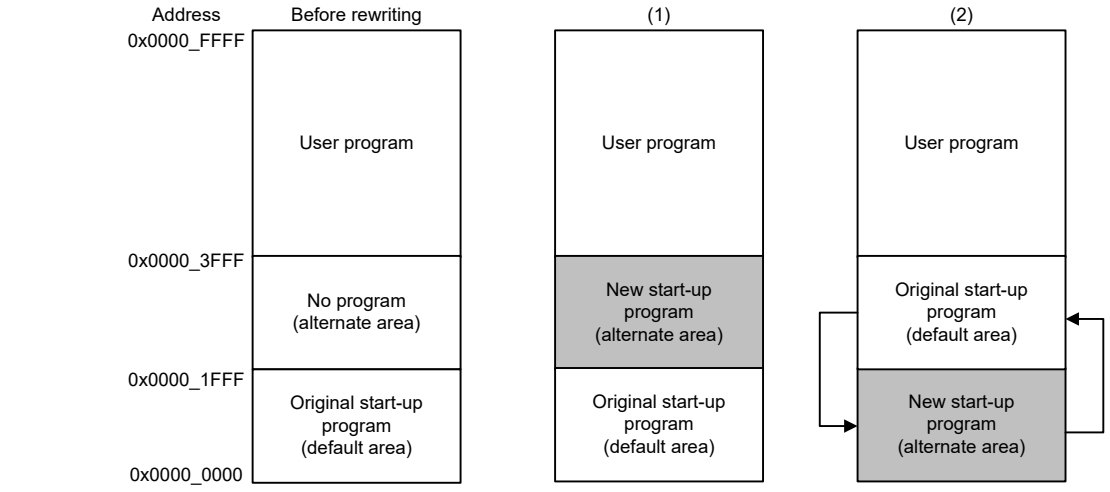
- Software protection
- Error protection
- Boot program protection.

33.9.1 Startup Program Protection

When programming of the startup area is interrupted by temporary blackout, the startup program may not be successfully programmed and the user program may not start properly.

This problem can be avoided by programming the startup program without erasing the existing startup program using the startup program protection.

Figure 33.8 shows an overview of the Startup Program Protection. In this figure, the default area indicates the 8-KB region from the start address and the alternate area indicates the next 8-KB region.



Note: (1) Program a new startup program in the alternate area. If the alternate area fails to be rewritten, the new startup program can be rewritten again after starting up using the default area because the original startup program is in the default area.
(2) After the alternate area is successfully rewritten, the default area and the alternate area are switched using the self-programming library. After that, the program in the alternate area starts after a reset.

Figure 33.8 Overview of the Startup Program Protection

33.9.2 Area Protection

Area protection enables rewriting for only selected blocks (access window) in the user area and disables programming for the other blocks. Data flash is not protected by the access window.

Select the start block and end block to set the access window. The access window is changeable and valid in programming mode (boot mode, self-programming mode, and OCD mode).

Figure 33.9 shows an overview of area protection.

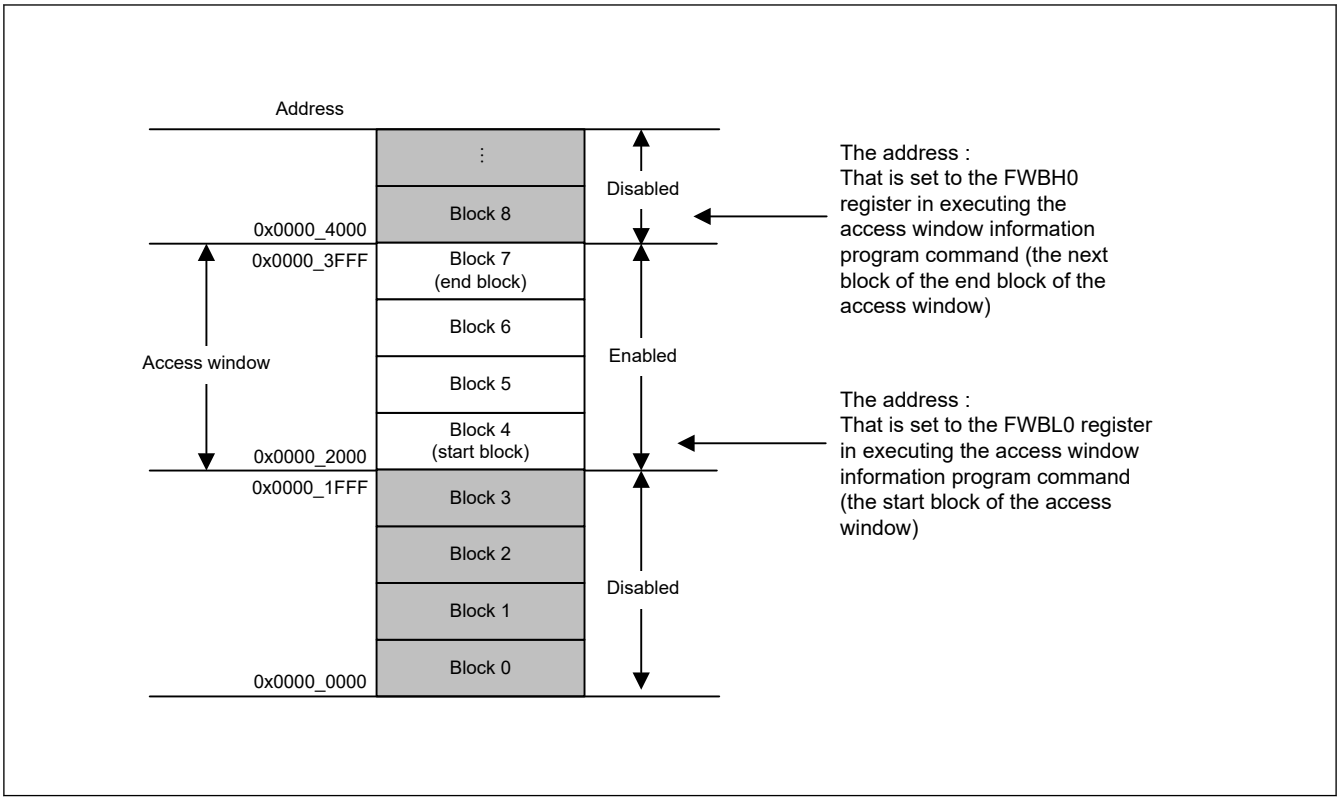


Figure 33.9 Area Protection Overview

33.10 Serial Programming Mode

The serial programming mode includes:

- Boot mode with SCI9

Table 33.18 lists the I/O pins of the flash memory-related modules.

Table 33.18 I/O pins of flash memory-related modules

Pin name	I/O	Applicable modes	Function
MD	Input	SCI boot mode (serial programming mode)	Selection of operating mode
P110/RXD9	Input	SCI boot mode	For host communication, to receive data through the SCI
P109/TXD9	Output		For host communication, to transmit data through the SCI

Note: Serial programming mode is not executed when security MPU is enabled.

33.10.1 SCI Boot Mode

In boot mode, the host sends control commands and data for programming, and the code flash memory and data flash memory areas are programmed or erased accordingly. An on-chip SCI handles transfers between the host and the MCU in asynchronous mode. Tools for transmission of control commands and the data for programming must be prepared in the host.

When the MCU is activated in boot mode, the embedded program for serial programming is executed. This program automatically adjusts the bit rate of the SCI and controls programming and erasure by receiving control commands from the host.

Figure 33.10 shows the system configuration for operations in boot mode.

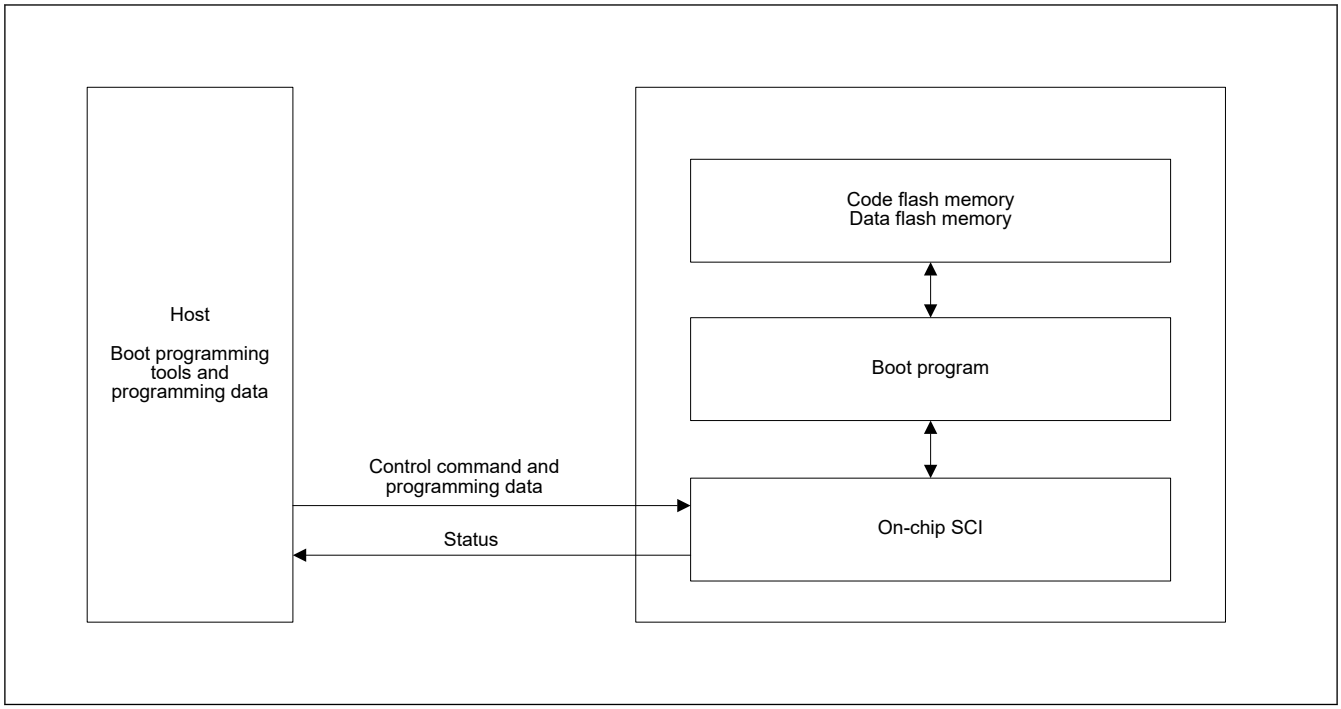


Figure 33.10 System configuration in SCI boot mode

33.11 Using a Serial Programmer

A dedicated flash memory programmer can be used to program the flash memory in serial programming mode.

33.11.1 Serial Programming

The MCU is mounted on the system board for serial programming. A connector to the board allows programming by the flash memory programmer.

Figure 33.11 shows the environment recommended by Renesas for programming the flash memory of the MCU with data.

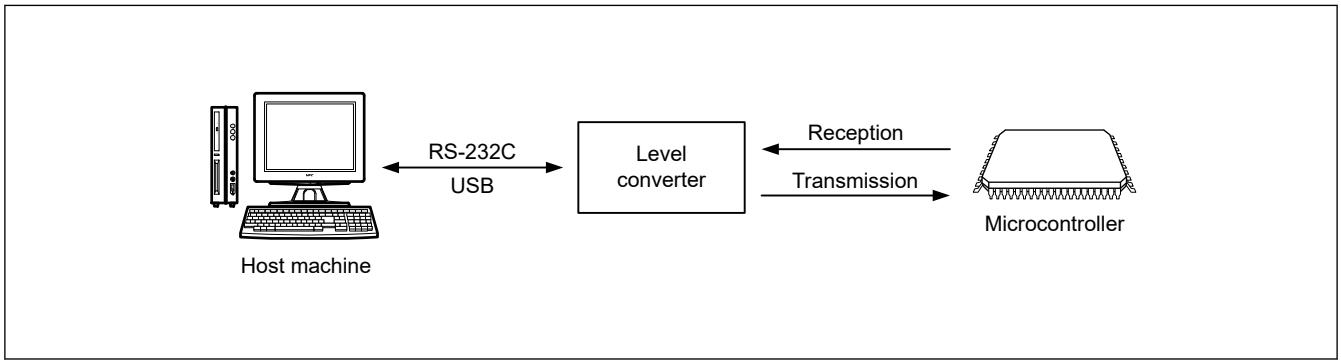


Figure 33.11 Environment for writing programs to the flash memory

33.12 Self-Programming

33.12.1 Overview

The MCU supports programming of the flash memory by the user program. The programming commands can be used with user programs for writing to the code and data flash memory. This enables updates to the user programs and overwriting of constant data fields.

In self-programming, it is necessary to supply a stable HOCO clock to the flash memory in order to generate the program voltage and erase voltage. Therefore, in case that the HOCO is stopped where another clock source is selected as the system clock, it is necessary to start the HOCO operation and ensure that the oscillation is in a stable state before executing

the self-programming. For details of HOCO clock oscillation stabilization check, see section 8.2.9. OSCSF : Oscillation Stabilization Flag Register.

The background operation facility makes it possible to execute a program from the code flash memory to program the data flash memory under the conditions shown in Figure 33.12. This program can also be copied in advance to and executed from the internal SRAM. When executing from the internal SRAM, this program can also program the code flash memory area.

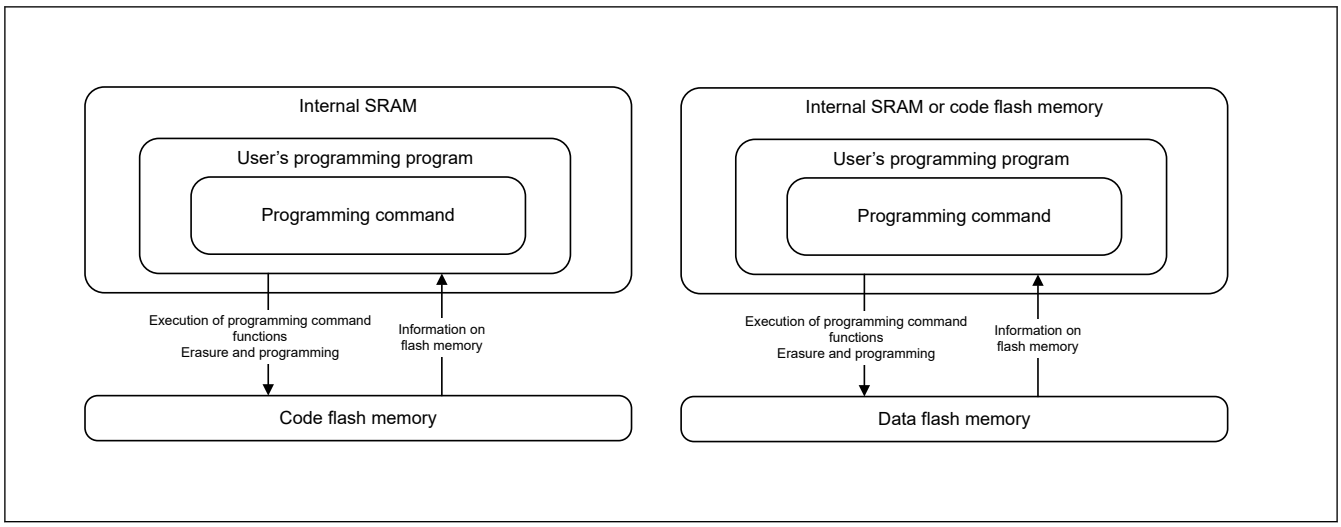


Figure 33.12 Schematic view of self-programming

33.12.2 Background Operation

Background operation can be used when a combination of the flash memory for writing and reading is as listed in Table 33.19.

Table 33.19 Conditions under which background operation is available

Product	Writable range	Readable range
All products	Data flash memory	Code flash memory

33.13 Programming and Erasure

The code flash and data flash can be programmed and erased by changing the mode of the dedicated sequencer for programming and erasure, and by issuing commands for programming and erasure.

The mode transitions and commands required to program or erase the code flash and data flash are described in the sections that follow. The descriptions apply in common to boot mode and single-chip mode.

33.13.1 Sequencer Modes

The sequencer has four modes and transitions between modes occur by writing to the FENTRYR or DFLCTL register, or by issuing commands to set the FPMCR register. Figure 33.13 shows mode transitions of the flash memory.

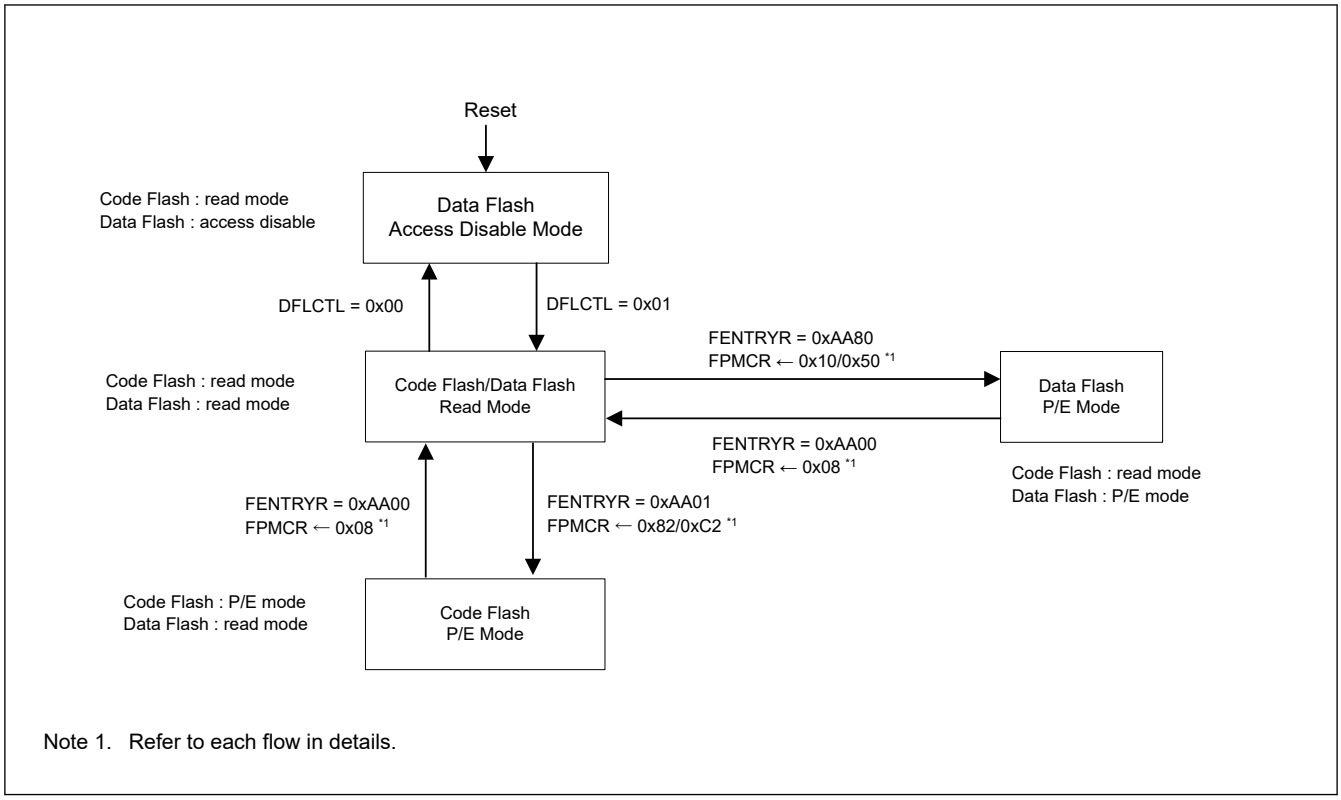


Figure 33.13 Mode transitions of the flash memory

33.13.1.1 Data Flash Access Disable Mode

Data flash access disable mode is to disable access to the data flash. Issuing a reset causes this mode. The data flash transitions to read mode by setting the DFLCTL.DFLEN bit to 1.

33.13.1.2 Read Mode

Read mode is used for high-speed reading of the code flash and data flash.

(1) Code Flash and Data Flash Read Mode

This mode is used for reading the code flash and data flash. The sequencer enters this mode when the FENTRYR.FENTRY0 bit is set to 0 while the FENTRYR.FENTRYD bit set to 0.

33.13.1.3 P/E Modes

(1) Code Flash P/E Mode

The code flash P/E mode is used for programming and erasure of the code flash. The sequencer enters this mode when the FENTRYR.FENTRYD bit is set to 0 while the FENTRYR.FENTRY0 bit set to 1. In this mode, it is not possible to access the data flash.

(2) Data Flash P/E Mode

The data flash P/E mode is used for programming and erasure of the data flash. High-speed reading from the code flash is possible. The sequencer enters this mode when the FENTRYR.FENTRY0 bit is set to 0 while the FENTRYR.FENTRYD bit is set to 1.

33.13.2 Software Commands

Software commands consist of commands for programming and erasure, and commands for programming startup program area information and access window information. Table 33.20 lists the software commands for use with the flash memory.

Table 33.20 Software Commands

Command	Function
Program	Code flash programming (4 bytes) Data flash programming (1 byte)
Block erase	Code flash/data flash erasure
Chip erase	Code flash/data flash erasure
Consecutive read	Read the specified area during code flash P/E mode or data flash P/E mode
Blank check	Check whether the specified area is blank. Confirm that data is not programmed in the area. This command does not guarantee whether the area remains erased.
Startup area information and security program	Set the FSPR or the SASMF to the extra area
Access window information program	Set the access window used for area protection to the extra area
OCDID program	Set the OCDID to the extra area

33.13.3 Software Command Usage

The following sections describe the usage of each software command.

(1) Switching from Data Flash Access Disable Mode to Read Mode

It is necessary to enter the code flash/data flash read mode from the data flash access disable mode. Figure 33.14 shows the procedure for entering the code flash/data flash read mode from the data flash access disable mode.

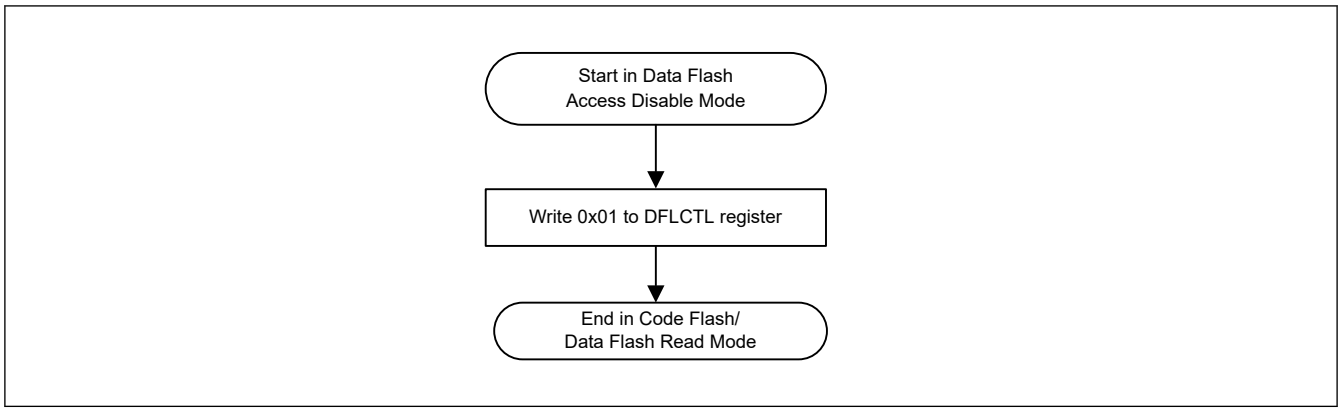
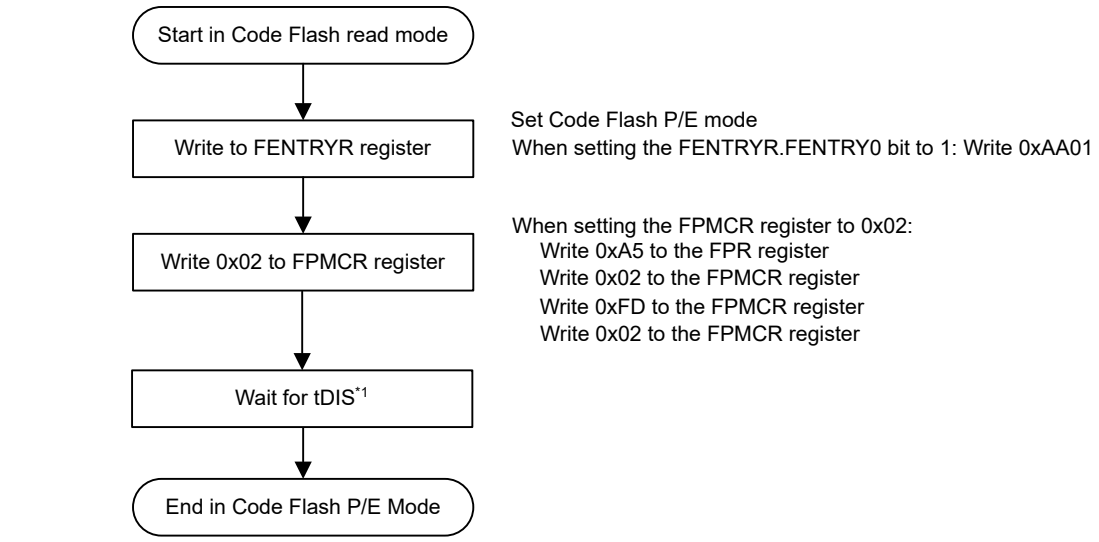


Figure 33.14 Mode transitions to read mode from data flash access disable mode

(2) Switching to Code Flash P/E Mode

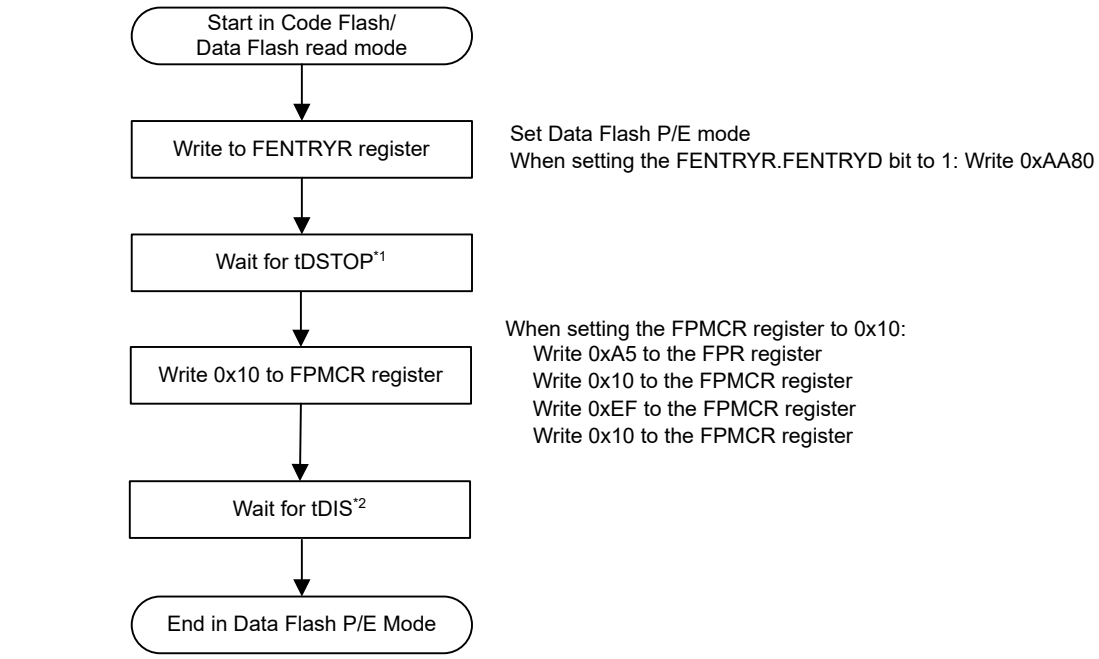
It is necessary to enter the code flash P/E mode by setting the FENTRY0 bit of the FENTRYR register before executing the software command for the code flash. Figure 33.15 shows the procedure for entering code flash P/E Mode.



Note 1. t_{DIS}: Flash memory mode transition wait time 1 (See section 35, Electrical Characteristics.)

Figure 33.15 Procedure for changing from read mode to code flash P/E mode

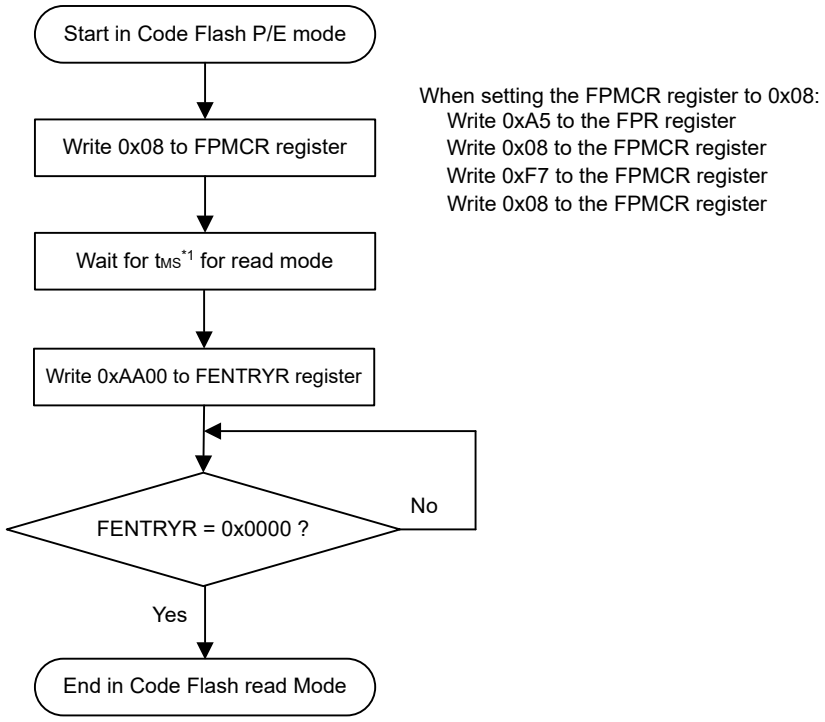
It is necessary to enter the data flash P/E mode by setting the FENTRYD bit of the FENTRYR register before executing the software command for the data flash. Figure 33.16 shows the procedure for entering to the data flash P/E Mode.



Note 1. t_{DSTOP}: STOP recovery time (See section 35, Electrical Characteristics.)
Note 2. t_{DIS}: Flash memory mode transition wait time 1 (See section 35, Electrical Characteristics.)

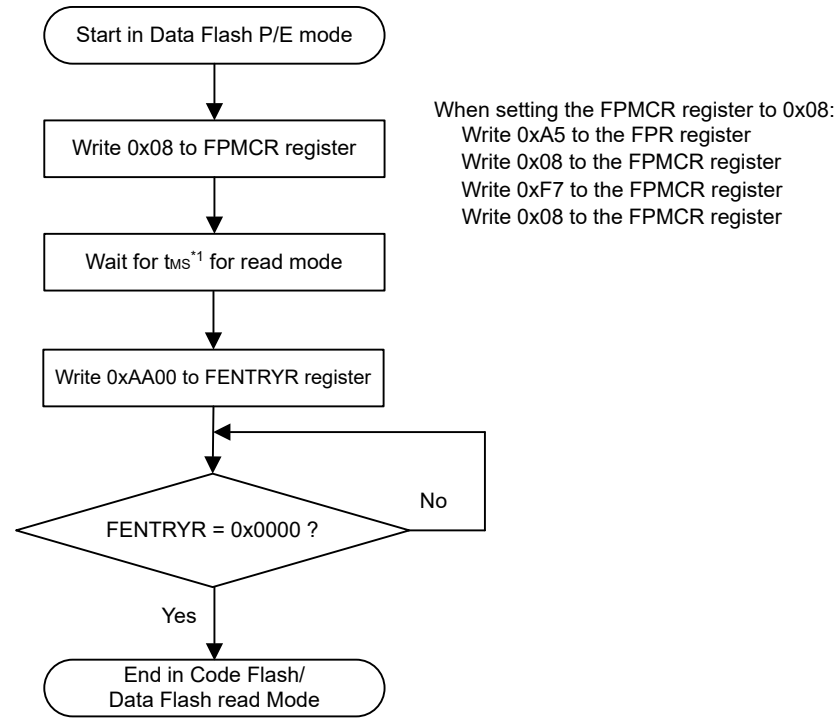
Figure 33.16 Procedure for changing from read mode to data flash P/E mode

(3) Switching the Code Flash or Data Flash P/E Mode to Read Mode



Note 1. t_{MS}: Flash memory mode transition wait time 2 (See section 35, Electrical Characteristics.)

Figure 33.17 Procedure for changing from code flash P/E mode to read mode



Note 1. t_{MS}: Flash memory mode transition wait time 2 (See section 35, Electrical Characteristics.)

Figure 33.18 Procedure for changing from data flash P/E mode to read mode

(4) Flowchart for programming the code flash or the data flash

The following figures describe the flow for programming the code flash or the data flash.

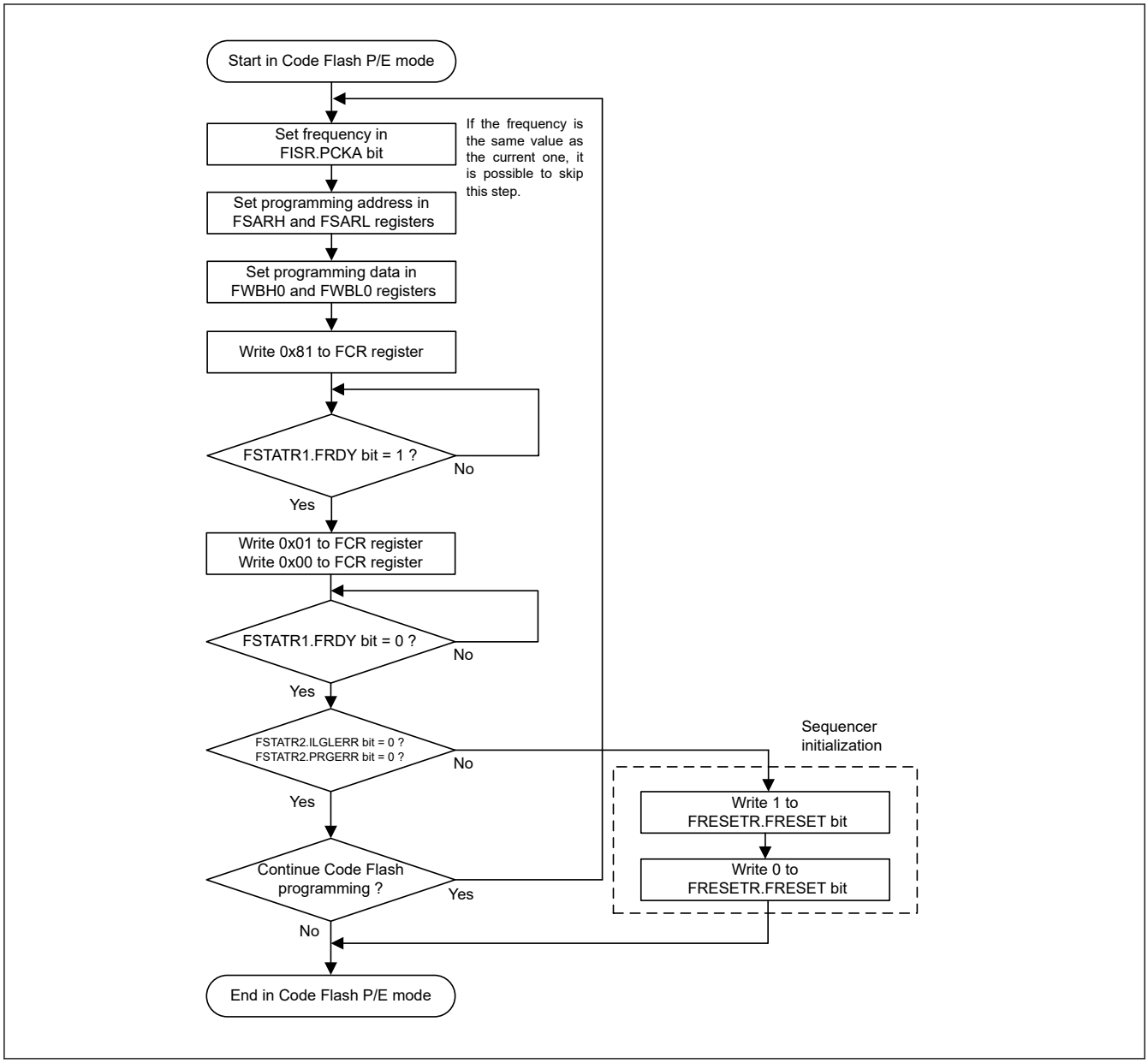


Figure 33.19 Flowchart for programming of the code flash

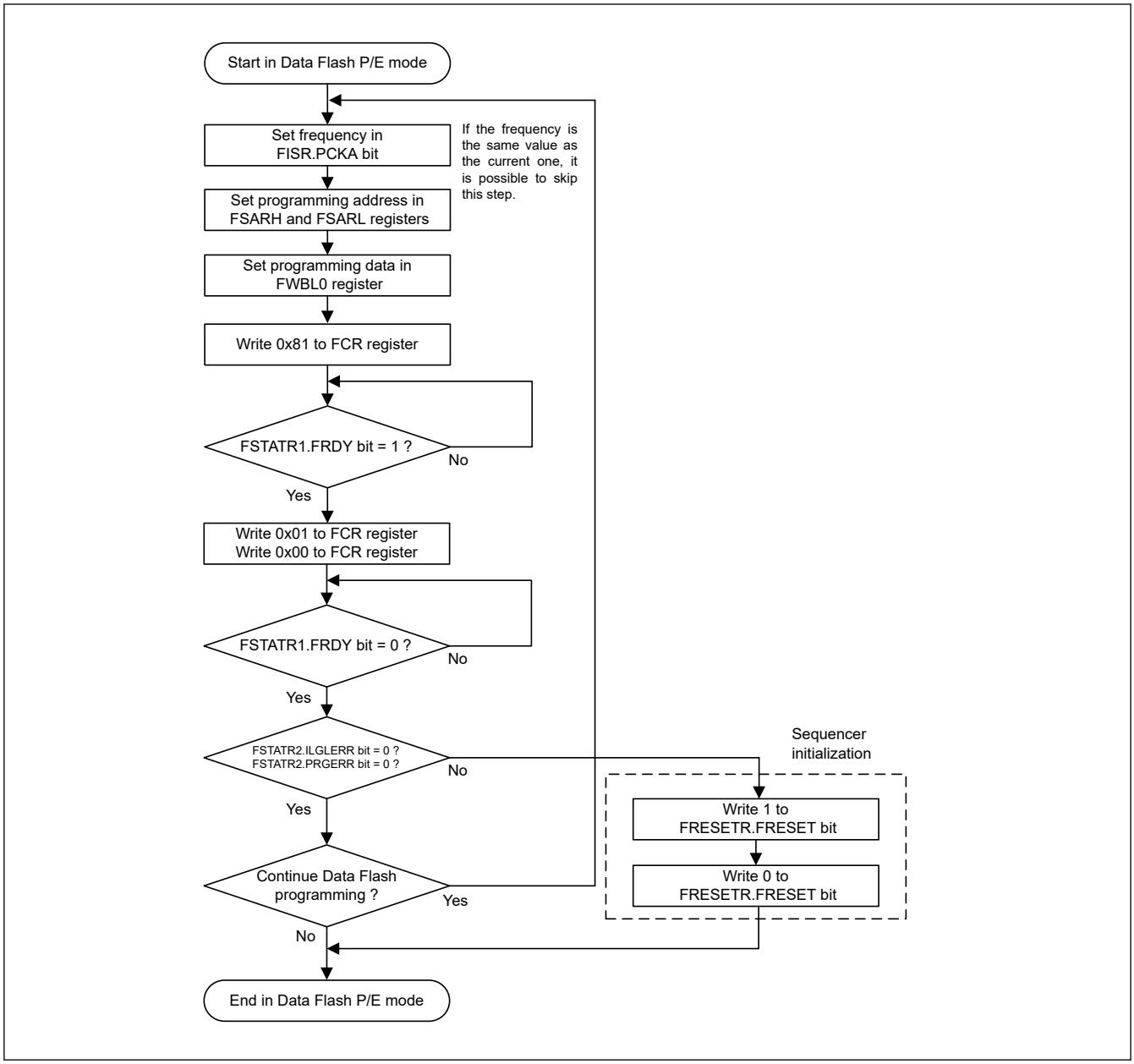


Figure 33.20 Flowchart for programming of the data flash

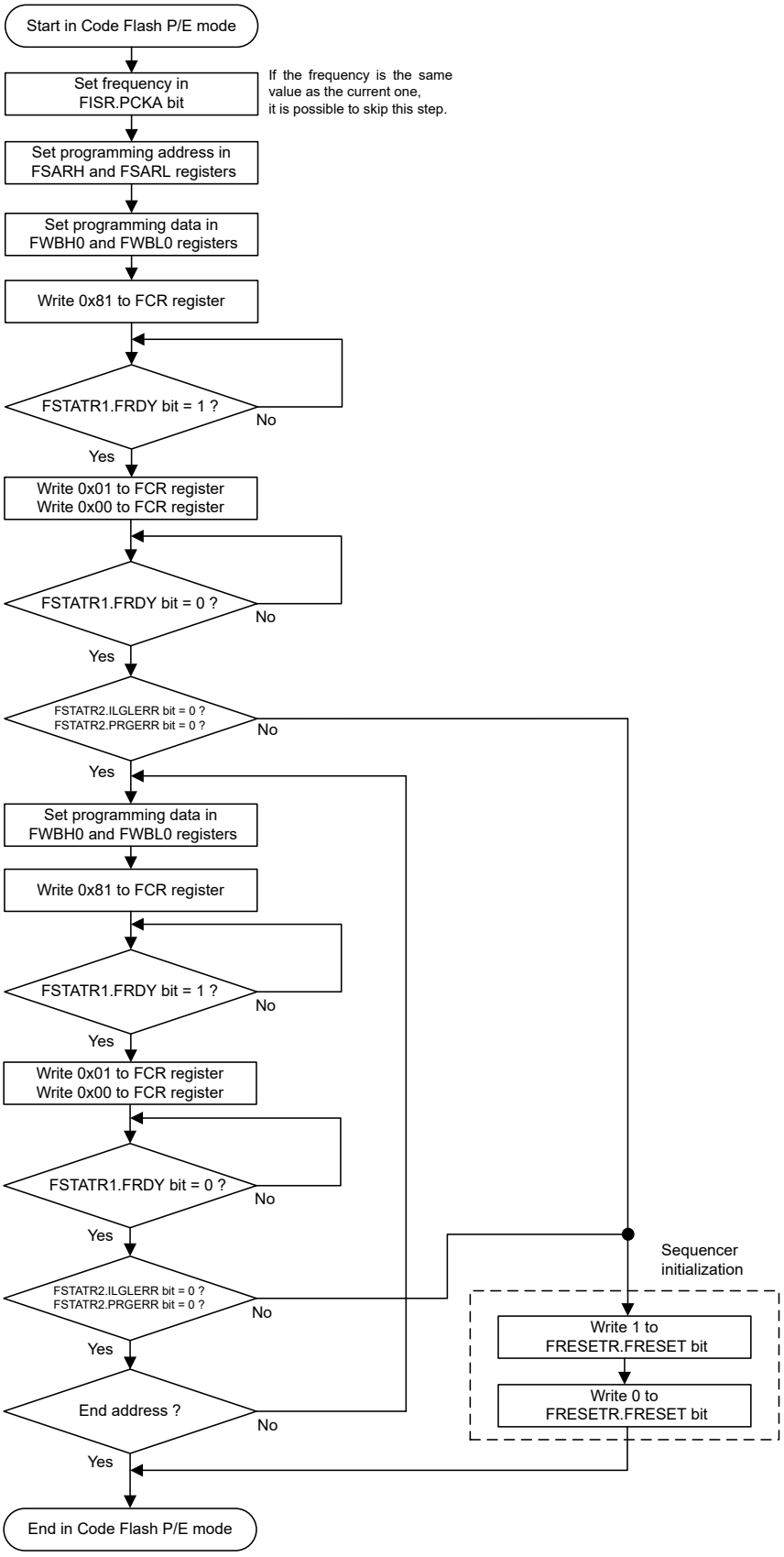


Figure 33.21 Flowchart for consecutive programming of the code flash

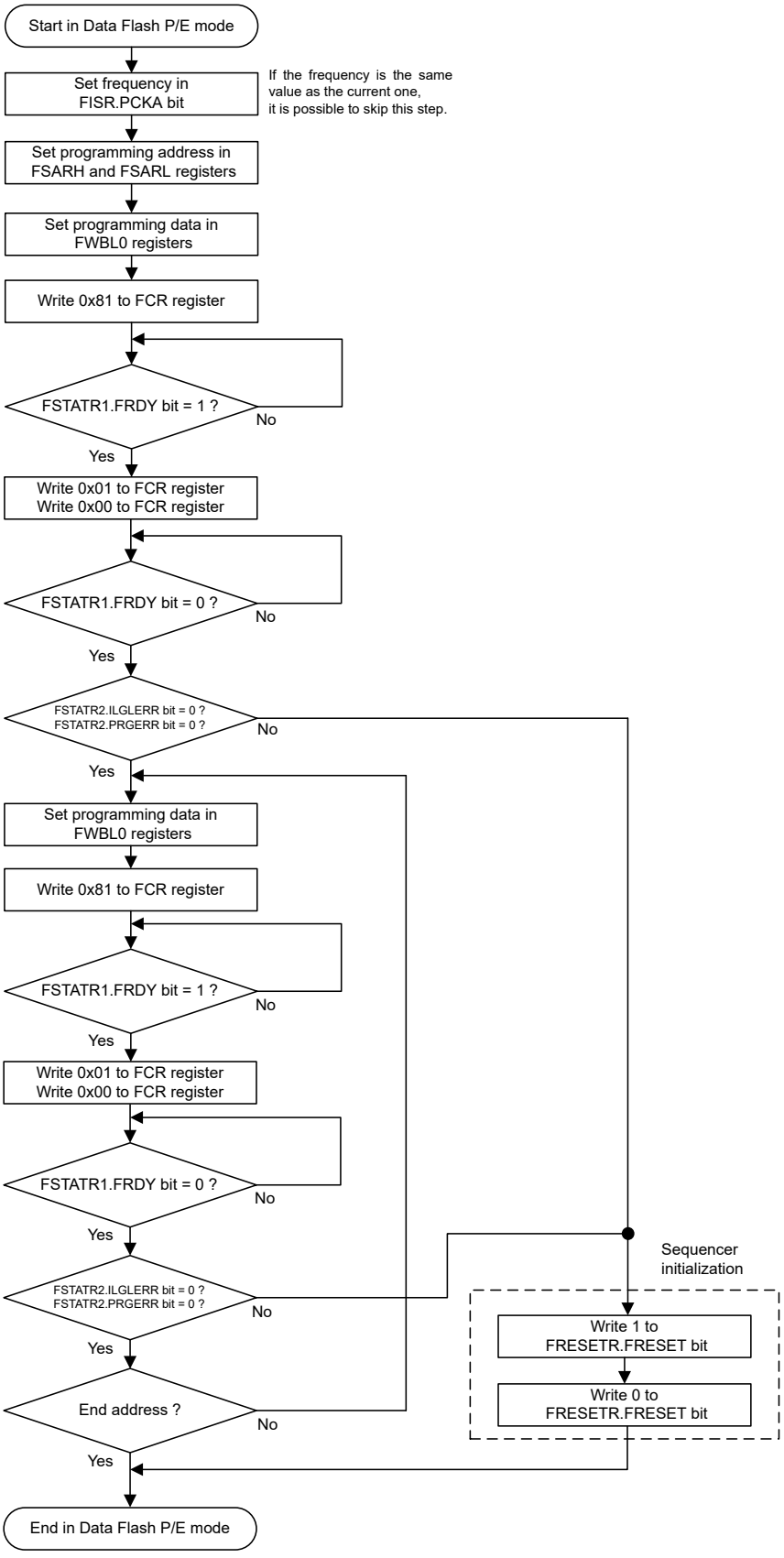


Figure 33.22 Flowchart for consecutive programming of the data flash

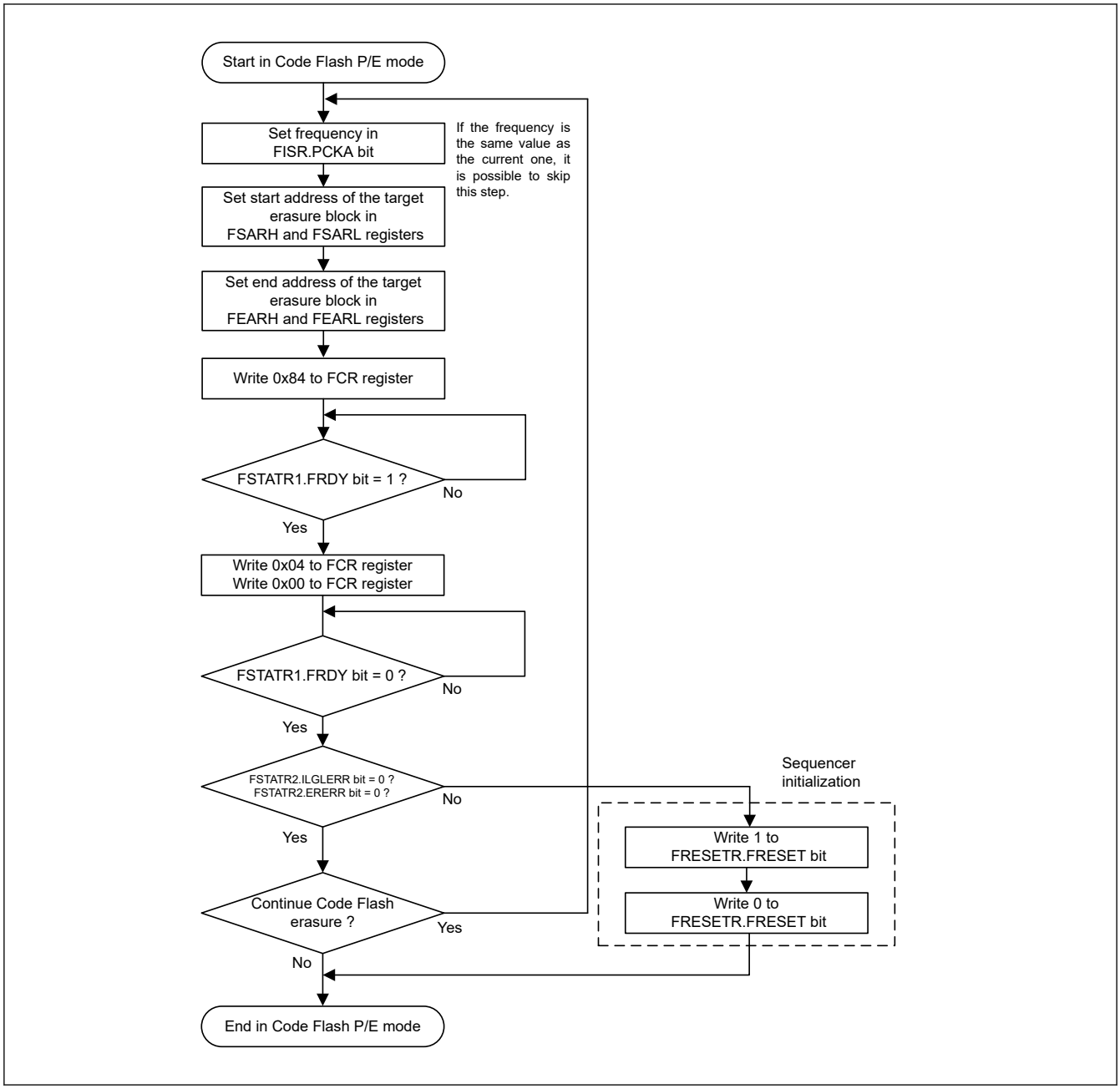


Figure 33.23 Flowchart for the code flash block erase procedure

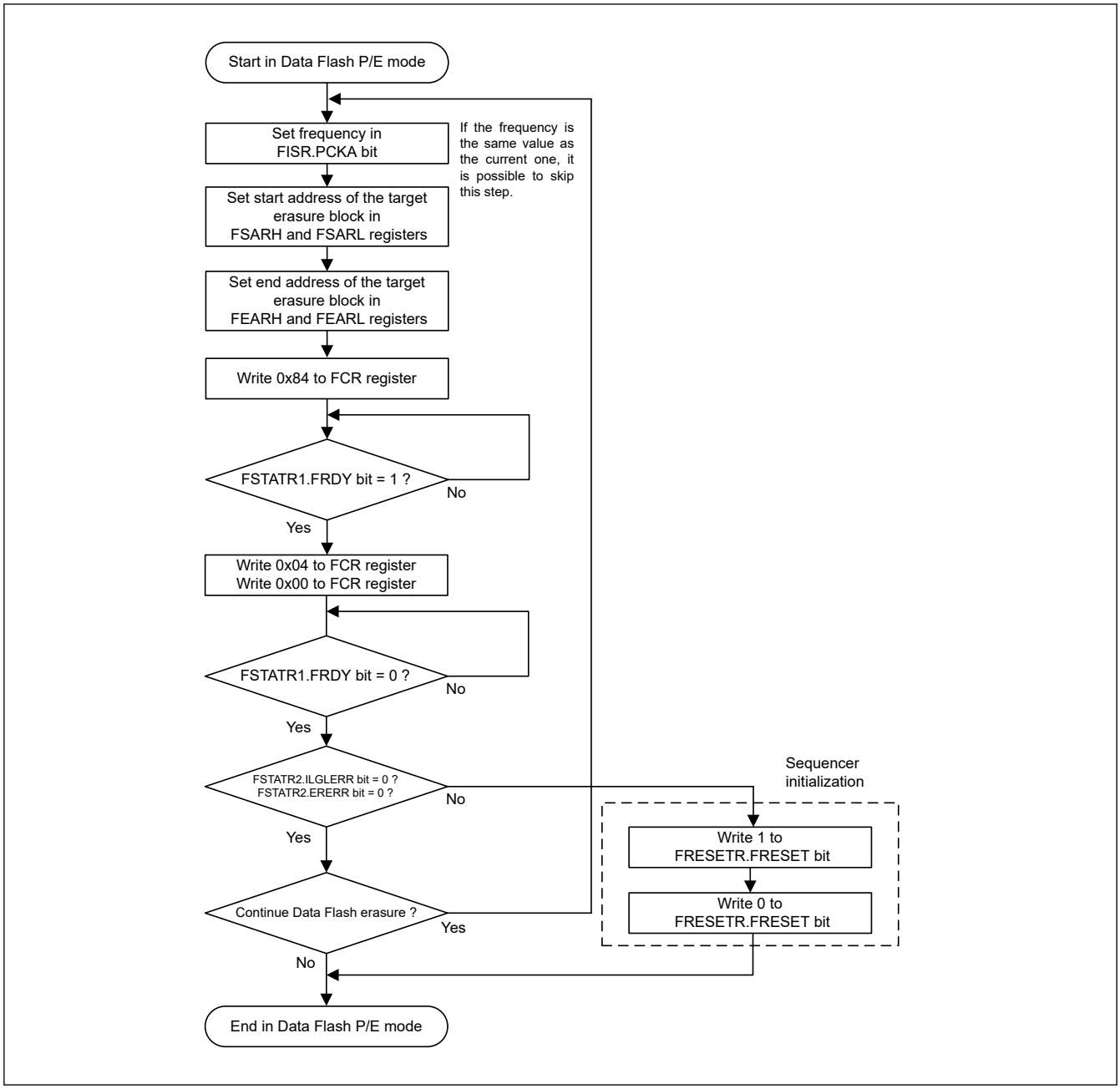


Figure 33.24 Flowchart for the data flash block erase procedure

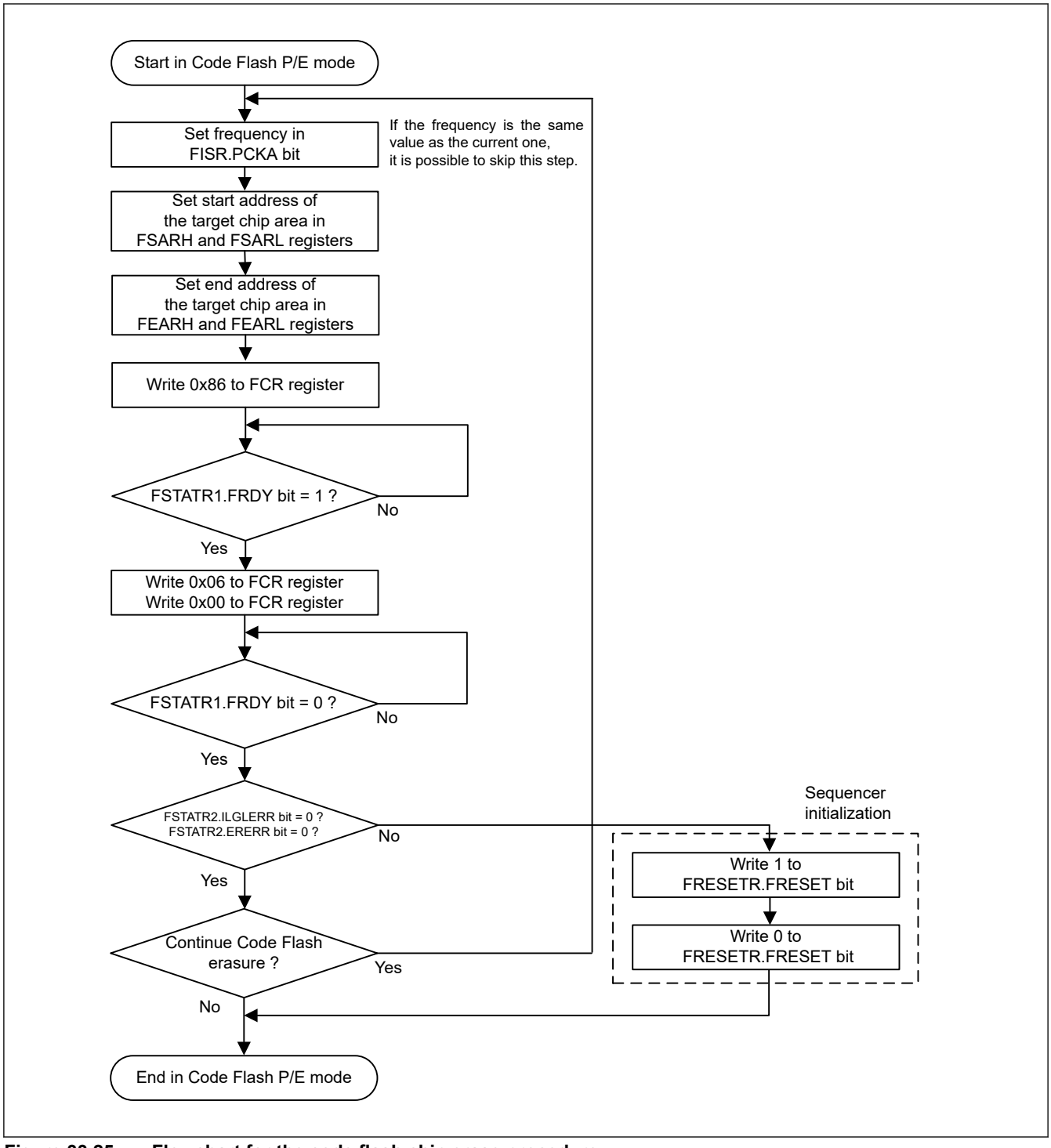


Figure 33.25 Flowchart for the code flash chip erase procedure

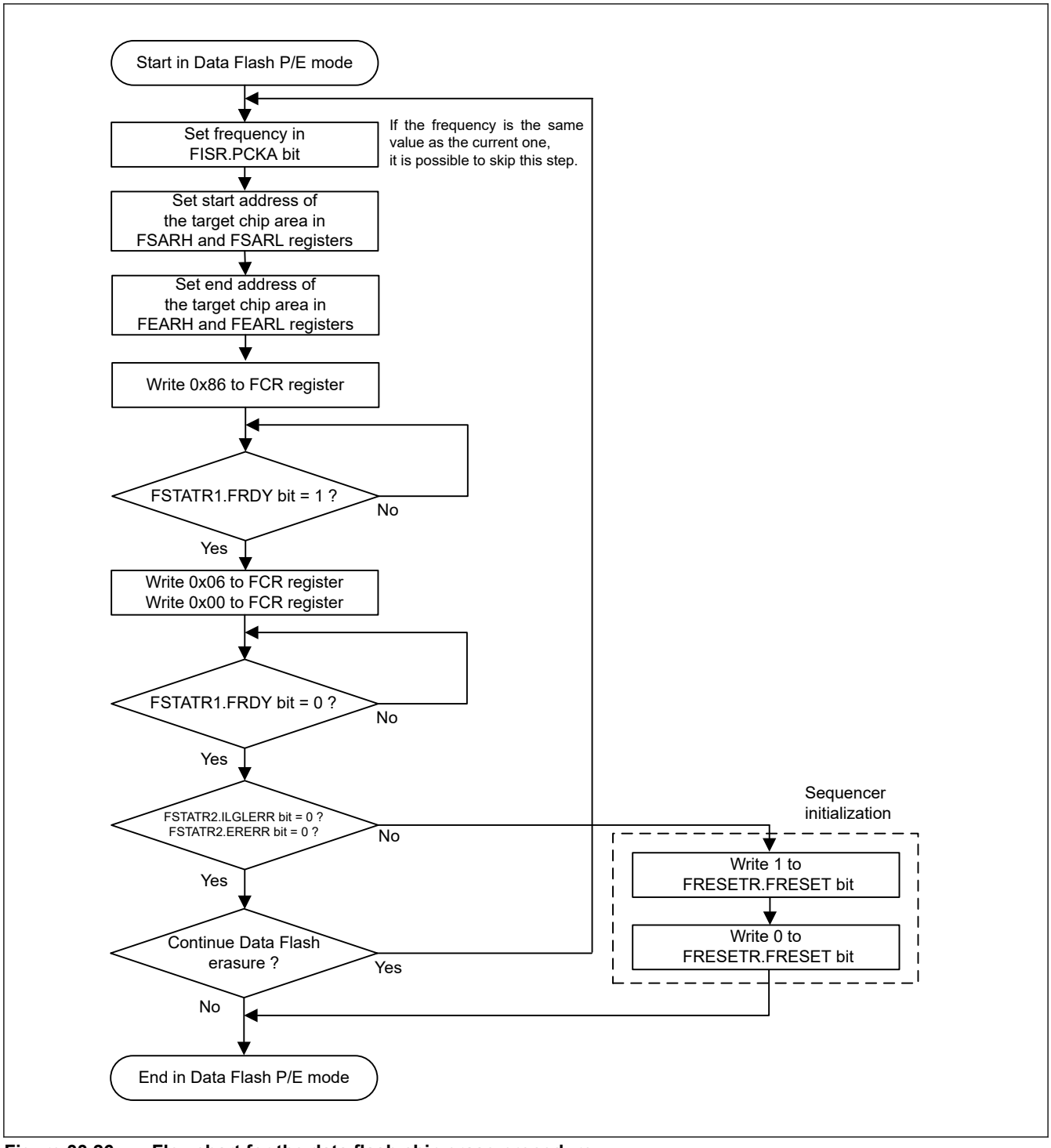


Figure 33.26 Flowchart for the data flash chip erase procedure

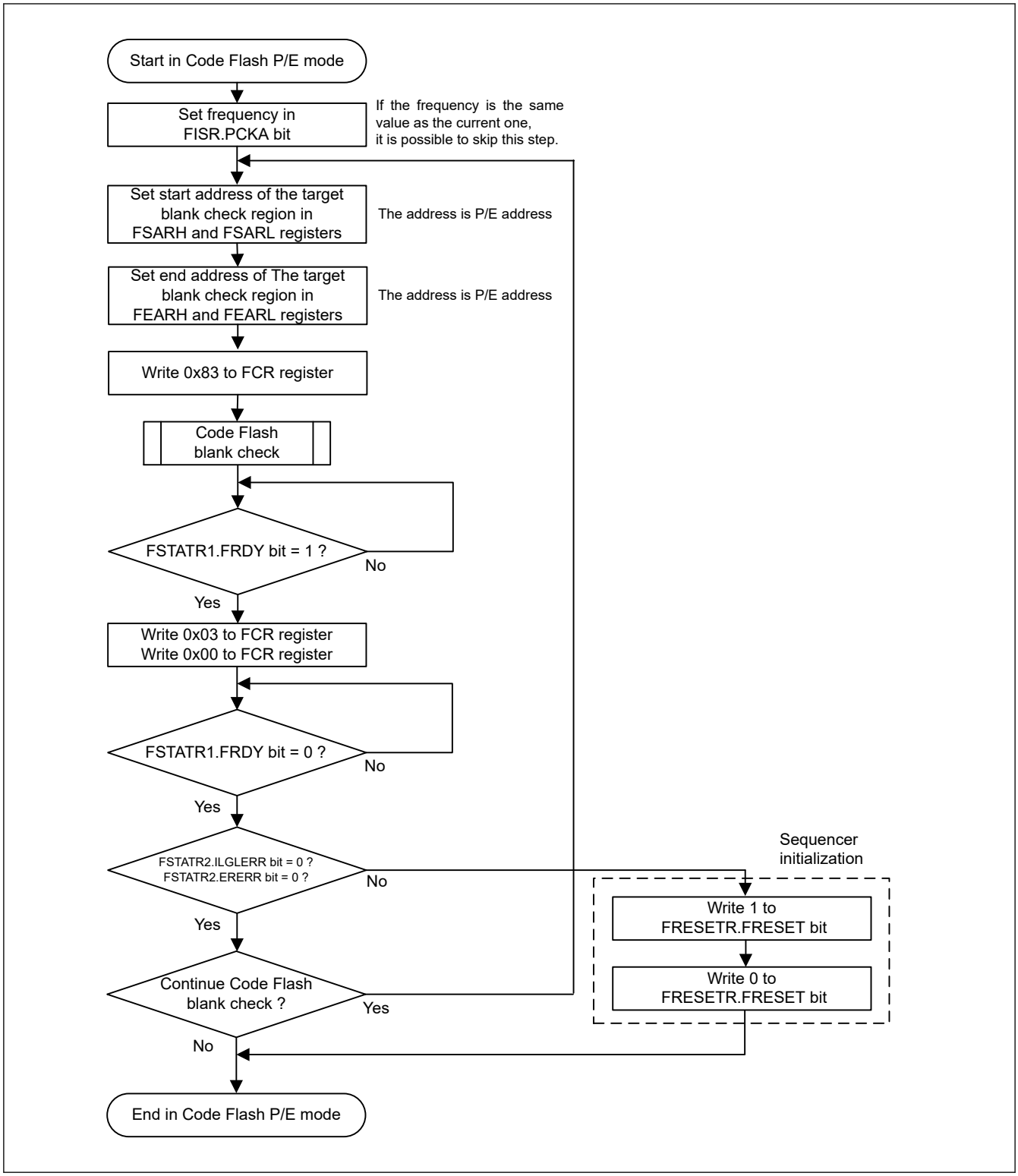


Figure 33.27 Flowchart for the code flash blank check procedure

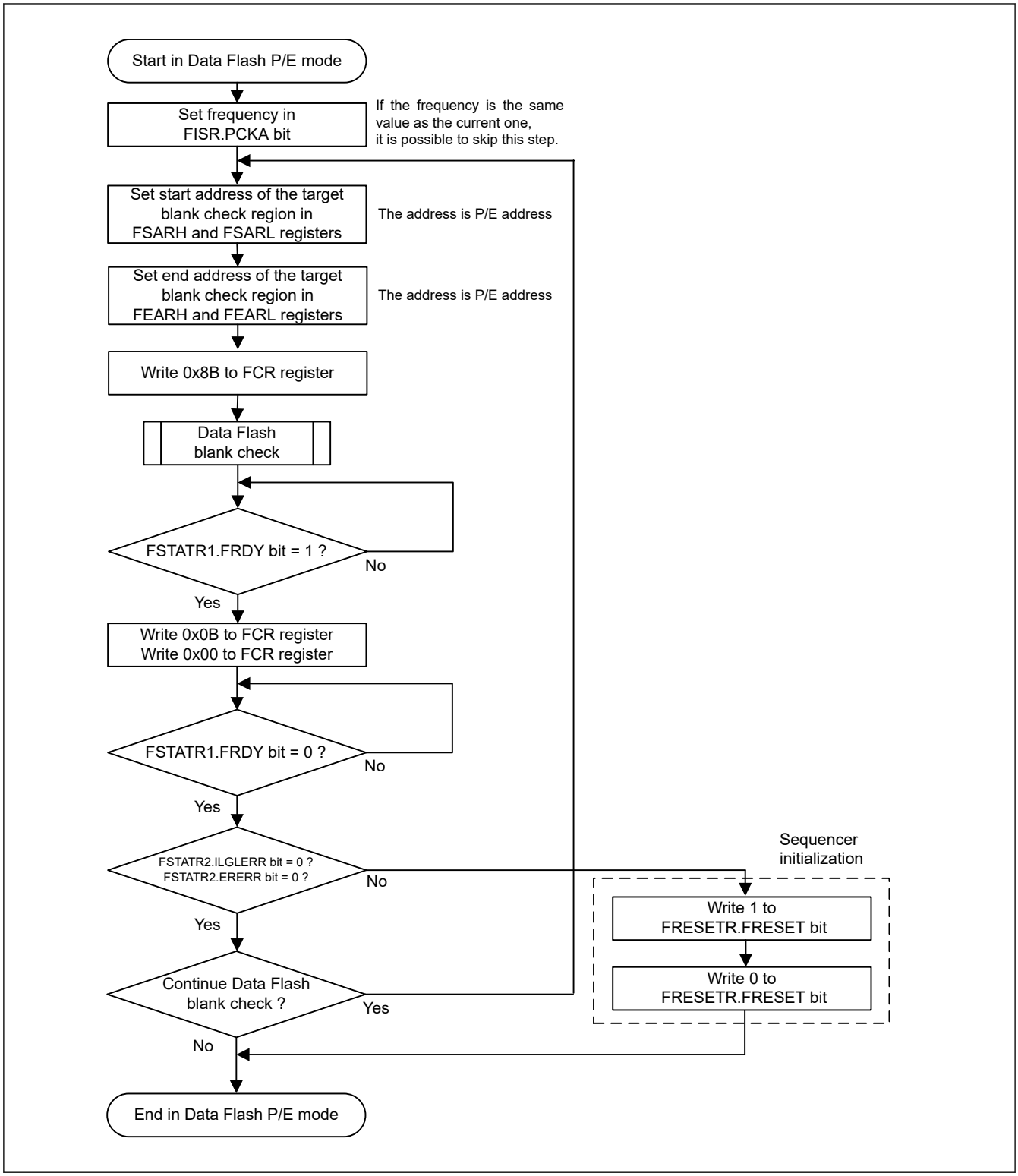
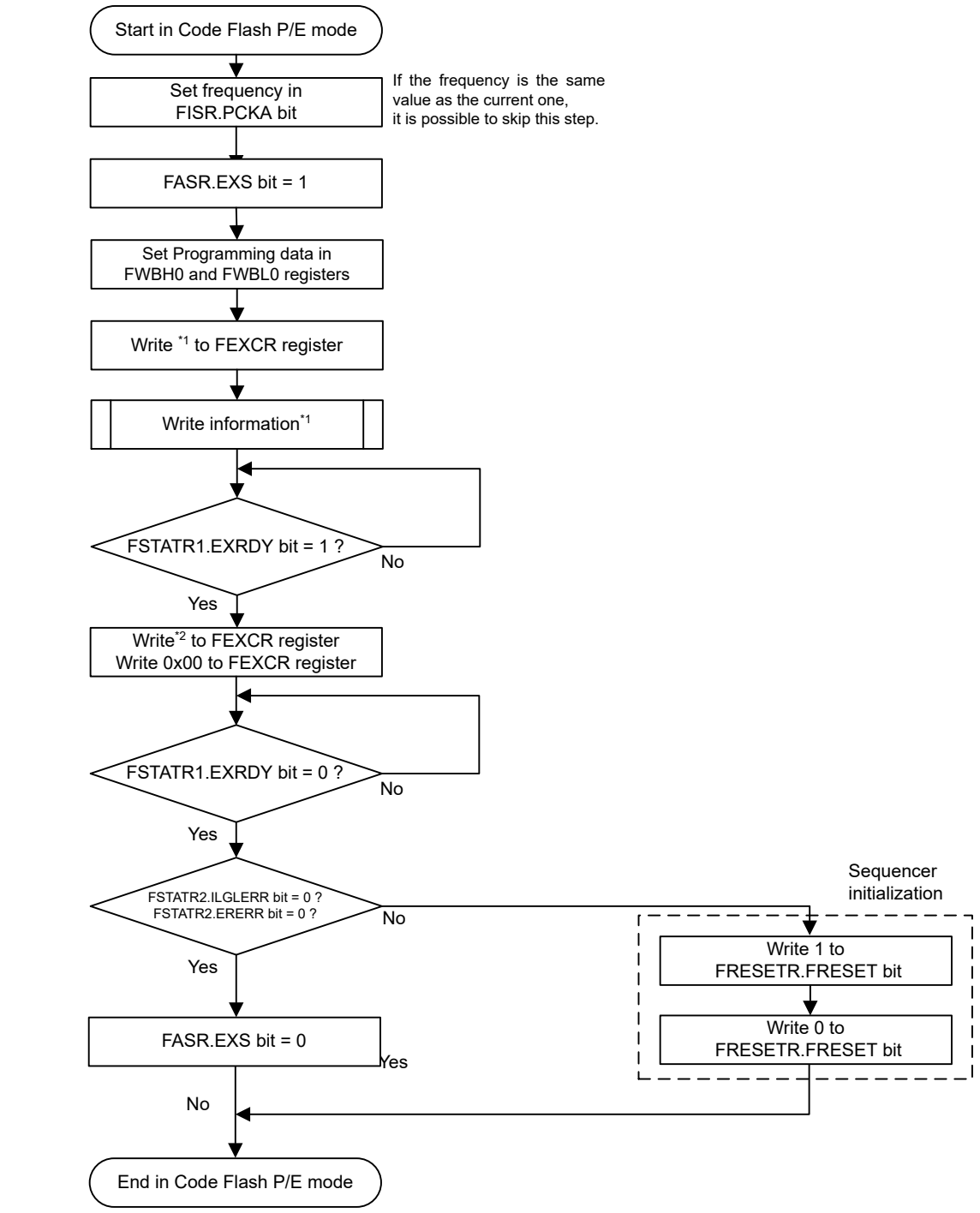


Figure 33.28 Flowchart for the data flash blank check procedure

(5) Startup Area Information and FSPR Program/Access Window Information Program/OCDID information Program

Figure 33.29 is a simple flowchart of the procedure for the startup area information and FSPR program/access window information program/OCDID information program.



Note: Order of the FSPR Bit Setting by Startup Area Information and FSPR Program
Note 1. Write data :
0x81 : Reserved
0x82 : start-up area information and FSPR bit access window information
0x83 : OCDID1 information
0x84 : OCDID2 information
0x85 : OCDID3 information
0x86 : OCDID4 information
Note 2. 0x0y (y = 1 to 6)

Figure 33.29 Simple flowchart for the procedure for Startup Area Information and FSPR Program/Access Window Information Program/OCDID information Program

Set the FSPR bit after programing of the startup area information and the access window information. If the FSPR bit is set before programing of the startup area information and the access window information, the programming cannot be performed because of the security function in the FSPR. When programming using the hex file, programming in the ascending order of the address. In this case, the FSPR bit is written before the access window information. Therefore, divide the hex file for FSPR into another file, and use it after setting the access window information.

(6) Consecutive Read

Figure 33.30 shows a simple flowchart for the consecutive read procedure.

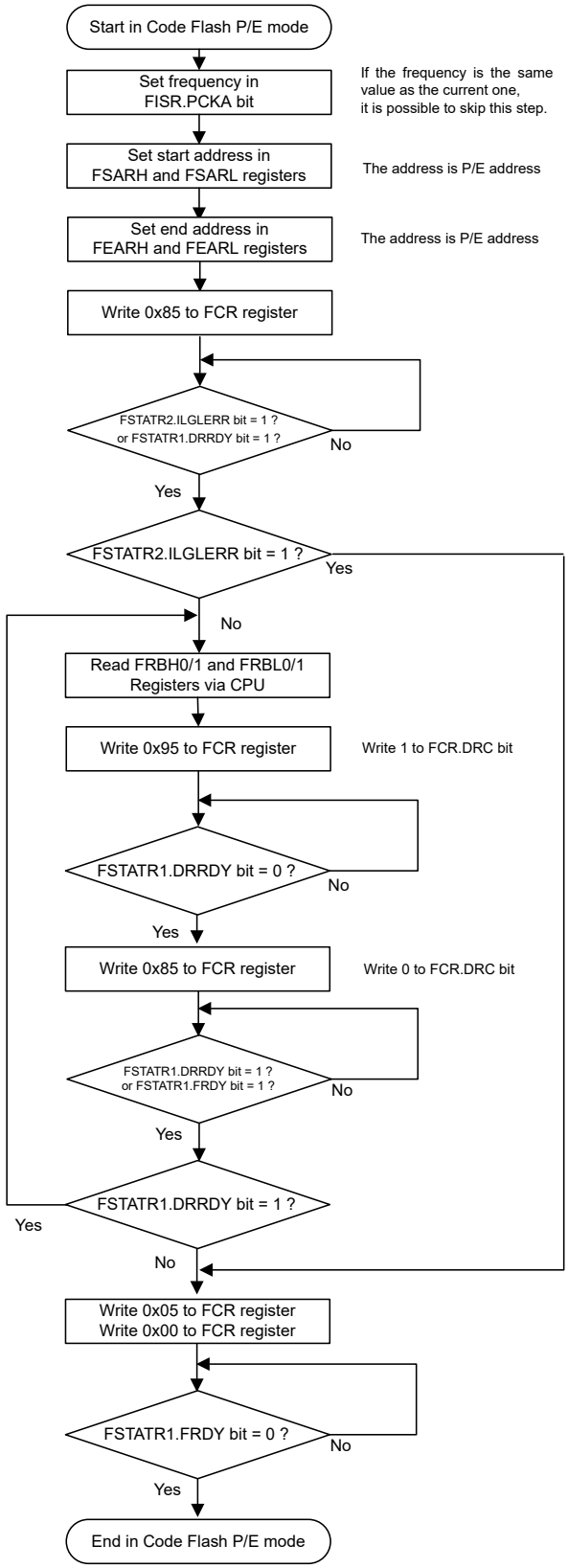


Figure 33.30 Simple flowchart for the consecutive read procedure

(7) Forced Stop by Software Command

Figure 33.31 shows a simple flowchart for the forced stop procedure to stop the blank check command, the block erase, or the chip erase command forcibly. When the forced stop command is executed, FEAMH/FEAML registers store the stopped address value. For the blank check command, the blank check can restart from the stopped address by copying the value of FEAMH/FEAML registers to FSARH/FSARL registers, respectively.

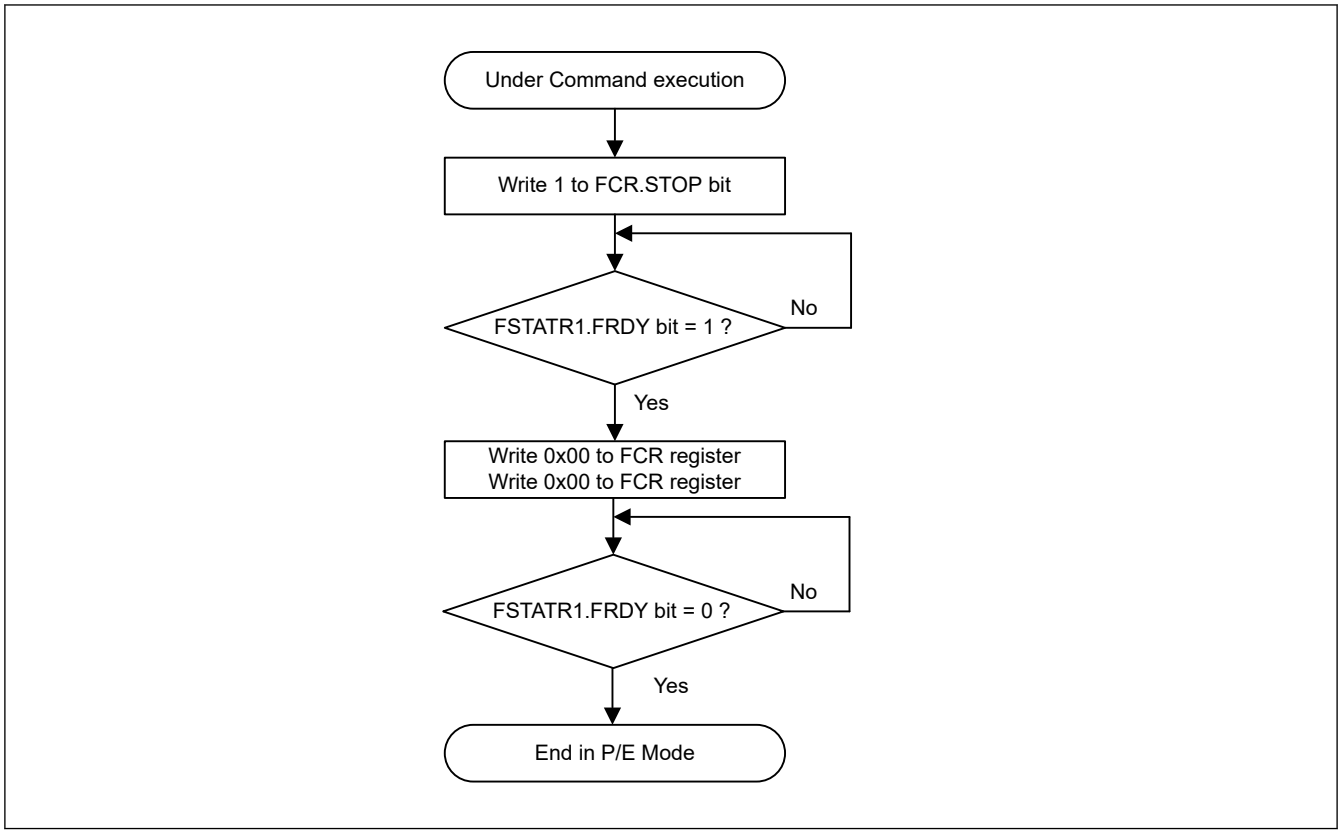


Figure 33.31 Simple flowchart for the forced stop procedure

33.14 Reading the Flash Memory

33.14.1 Reading the Code Flash Memory

No special settings are required to read the code flash memory in Normal mode. Data can be read by accessing the addresses in the code flash memory. When reading code flash memory that is erased but not yet reprogrammed, such as code flash memory in the non-programmed state, all bits are read as 1s.

33.14.2 Reading the Data Flash Memory

No special settings are required to read the data flash memory in Normal mode except when issuing a reset that causes the data flash access disable mode to disable reading. In this case, the application must transfer back to the data flash read mode. When reading data flash memory that is erased but not yet reprogrammed, such as data flash in the non-programmed state, all bits are read as 1s.

33.15 Usage Notes

33.15.1 Erase Suspended Area

Data in areas where an erase operation is suspended is undefined. To avoid malfunctions caused by reading undefined data, do not execute commands and read data in the area where erase operation is suspended.

33.15.2 Suspension by Erase Suspend Commands

When suspending an erase operation with the erase suspend command, complete the operation with a resume command.

33.15.3 Constraints on Additional Writes

Other than the configuration area, no other area can be written to twice. After a write to a flash memory area is complete, erase the area before attempting to overwrite data in that area. The configuration area can be overwritten.

33.15.4 Reset during Programming and Erasure

If inputting a reset from the RES pin, release the reset after a reset input time of at least t_{RESW}. See section 35.3.3. Reset Timing within the range of the operating voltage defined in the electrical characteristics.

The IWDT reset and software reset do not require a t_{RESW} input time.

33.15.5 Non-Maskable Interrupt Disabled during Programming and Erasure

When a non-maskable interrupt[*1] occurs during a programming or erasure operation, the vectors are fetched from the code flash memory, and undefined data is read. Therefore, do not generate a non-maskable interrupt during programming and erasure operations in the code flash memory. This constraint applies only to the code flash memory.

Note 1. A non-maskable interrupt is an NMI pin interrupt, oscillation stop detection interrupt, WDT underflow or refresh error, IWDT underflow or refresh error, voltage monitor 1 interrupt, voltage monitor 2 interrupt, SRAM parity error, MPU bus slave error, MPU bus master error, or CPU stack pointer monitor.

33.15.6 Location of Interrupt Vectors during Programming and Erasure

When an interrupt occurs during a programming and erasure operation, the vector can be fetched from the code flash memory as default setting. To avoid fetching the vector from the code flash memory, set the destination for fetching interrupt vectors to an area other than the code flash memory with the interrupt table.

33.15.7 Programming and Erasure in Subosc-Speed Operating Mode

Do not program or erase the flash memory when subosc-speed operating mode is selected in the SOPCCR register for low-power consumption functions.

33.15.8 Abnormal Termination during Programming and Erasure

When the voltage exceeds the range of the operating voltage during a programming and erasure operation, or when a programming or erasure operation did not complete successfully because of a reset or prohibited actions as described in section 33.15.9. Actions Prohibited during Programming and Erasure, erase the area again.

33.15.9 Actions Prohibited during Programming and Erasure

To prevent damage to the flash memory, comply with the following instructions during programming and erasure:

- Do not use an MCU power supply that is outside the operating voltage range
- Do not update the OPCCR.OPCM[1:0] bits value
- Do not update the SOPCCR.SOPCM bit value
- Do not change the division ratio of the system clock (ICLK)
- Do not place the MCU in Software Standby mode
- Do not access the data flash memory during a program or erase operation to the code flash memory
- Do not change the data flash access control setting during a program or erase operation to the data flash memory.

33.15.10 Additional Programming Disabled

The same address cannot be programmed more than once. When programming an area that has been already programmed, erase the area first.

33.15.11 Flash-IF clock (ICLK) during Program/Erase

For programming/erasure by self-programming, it is necessary to specify an integer frequency by setting the Flash Initial Setting Register (FISR).

34. Internal Voltage Regulator

34.1 Overview

The MCU includes one internal voltage regulator:

- Linear regulator (LDO)

This regulator supplies voltage to all internal circuits and memory except for I/O and analog domains.

34.2 Operation

Table 34.1 lists the LDO pin settings, and Figure 34.1 shows the LDO settings.

Table 34.1 LDO pin

Pins	Setting descriptions
All VCC	• Connect each pin to the system power supply. • Connect each pin to VSS through a 0.1-μF multilayer ceramic capacitor. Place the capacitor close to the pin.
VCL	Connect the pin to VSS through a 4.7-μF multilayer ceramic capacitor. Place the capacitor close to the pin.

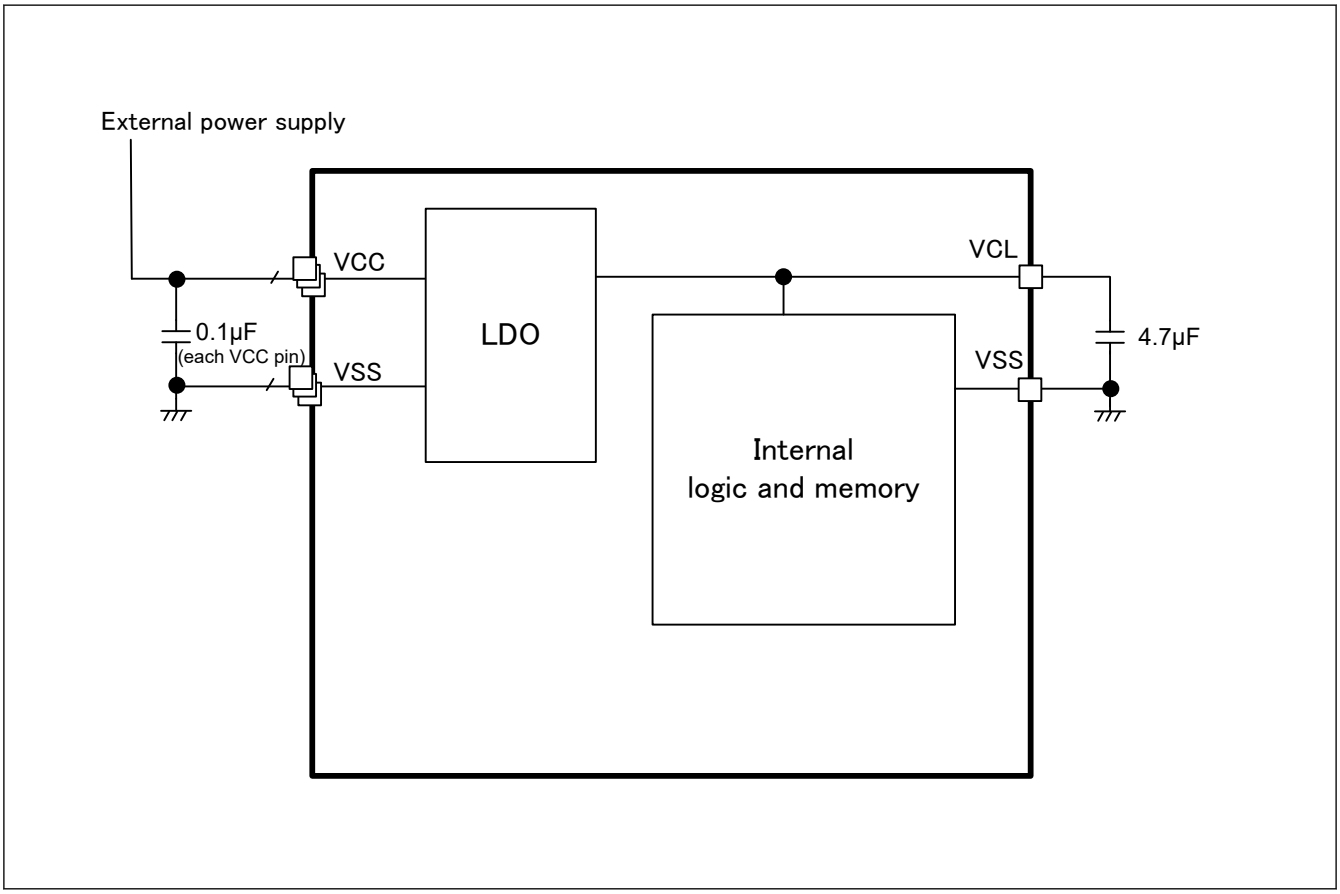


Figure 34.1 LDO settings

35. Electrical Characteristics

Unless otherwise specified, the electrical characteristics of the MCU are defined under the following conditions:

VCC[*1] = AVCC0 = 1.6 to 5.5 V, VREFH0 = 1.6 V to AVCC0

VSS = AVSS0 = VREFL0 = 0 V, Ta = T_{opr}

Note 1. The typical condition is set to VCC = 3.3 V.

Figure 35.1 shows the timing conditions.

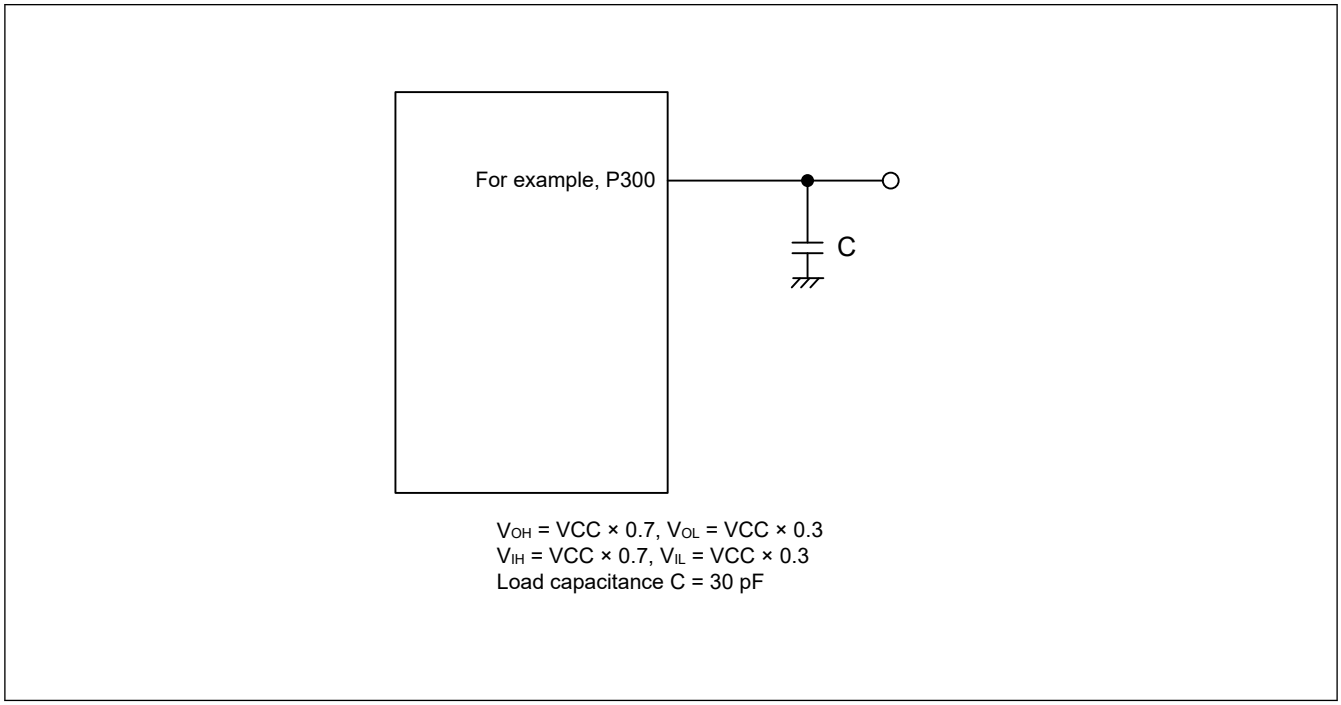


Figure 35.1 Input or output timing measurement conditions

The measurement conditions of the timing specifications for each peripheral are recommended for the best peripheral operation. However, make sure to adjust driving abilities for each pin to meet the conditions of your system.

Each function pin used for the same function must select the same drive ability. If the I/O drive ability of each function pin is mixed, the AC characteristics of each function are not guaranteed.

35.1 Absolute Maximum Ratings

Table 35.1 Absolute maximum ratings (1 of 2)

Parameter		Symbol	Value	Unit
Power supply voltage		VCC	-0.5 to +6.5	V
Input voltage	5V-tolerant ports[*1]	V_{in}	-0.3 to +6.5	V
	P000 to P002, P010 to P015	V_{in}	-0.3 to AVCC0 + 0.3	V
	Others	V_{in}	-0.3 to VCC + 0.3	V
Reference power supply voltage		VREFH0	-0.3 to +6.5	V
Analog power supply voltage		AVCC0	-0.5 to +6.5	V
Analog input voltage	When AN000 to AN002, AN005 to AN010 are used	V_{AN}	-0.3 to AVCC0 + 0.3	V
	When AN019 to AN022 are used		-0.3 to VCC + 0.3	V
Operating temperature[*2 *3 *4]		T_{opr}	-40 to +105 -40 to +125	°C

Table 35.1 Absolute maximum ratings (2 of 2)

Parameter	Symbol	Value	Unit
Storage temperature	T_{stg}	-55 to +140	°C

Note 1. Ports P400, P401 are 5V-tolerant.
Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up might cause malfunction and the abnormal current that passes in the device at this time might cause degradation of internal elements.
Note 2. See section 35.2.1. Tj/Ta Definition.
Note 3. Contact Renesas Electronics sales office for information on derating operation under Ta = +105°C to +125°C.
Derating is the systematic reduction of load for improved reliability.
Note 4. The upper limit of the operating temperature is 105°C or 125°C, depending on the product. For details, see section 1.3. Part Numbering.

Caution: Permanent damage to the MCU may result if absolute maximum ratings are exceeded.

To preclude any malfunctions due to noise interference, insert capacitors with high frequency characteristics between the VCC and VSS pins, between the AVCC0 and AVSS0 pins, and between the VREFH0 and VREFL0 pins when VREFH0 is selected as the high potential reference voltage for the ADC12. Place capacitors of the following value as close as possible to every power supply pin and use the shortest and heaviest possible traces:

- **VCC and VSS: about 0.1 µF**
- **AVCC0 and AVSS0: about 0.1 µF**
- **VREFH0 and VREFL0: about 0.1 µF**

Also, connect capacitors as stabilization capacitance.

Connect the VCL pin to a VSS pin by a 4.7 µF capacitor. Each capacitor must be placed close to the pin.

Table 35.2 Recommended operating conditions

Parameter	Symbol		Min	Typ	Max	Unit
Power supply voltages	VCC*1 *2		1.6	—	5.5	V
	VSS		—	0	—	V
Analog power supply voltages	AVCC0*1 *2		1.6	—	5.5	V
	AVSS0		—	0	—	V
	VREFH0	When used as ADC12 Reference	1.6	—	AVCC0	V
	VREFL0		—	0	—	V

Note 1. Use AVCC0 and VCC under the following conditions:
AVCC0 = VCC
Note 2. When powering on the VCC and AVCC0 pins, power them on at the same time or the VCC pin first and then the AVCC0 pins.
When powering off the VCC and AVCC0 pins, power them off at the same time or the AVCC0 pin first and then the VCC pins.

35.2 DC Characteristics

35.2.1 Tj/Ta Definition

Table 35.3 DC characteristics

Conditions: Products with operating temperature (Ta) -40 to +125°C

Parameter	Symbol	Typ	Max	Unit	Test conditions
Permissible junction temperature	Tj	—	140	°C	High-speed mode Middle-speed mode Low-speed mode Subosc-speed mode
			125*1		

Note: Make sure that Tj = T_a + θja × total power consumption (W), where total power consumption = (VCC - V_{OH}) × ΣI_{OH} + V_{OL} × ΣI_{OL} + I_{CC}max × VCC.
Note 1. The upper limit of operating temperature is 105°C or 125°C, depending on the product. For details, see section 1.3. Part Numbering. If the part number shows the operation temperature at 105°C, then the maximum value of Tj is 125°C, otherwise it is 140°C.

35.2.2 I/O V_{IH}, V_{IL}

Table 35.4 I/O V_{IH}, V_{IL}

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter	Ports & Functions		Symbol	Min	Max	Unit	Test Conditions
Input voltage	Input ports pins P000 to P002, P010 to P015		V_{IH}	AVCC0 × 0.8	—	V	—
			V_{IL}	—	AVCC0 × 0.2		
	Input ports pins except for P000 to P002, P010 to P015		V_{IH}	VCC × 0.8	—		
			V_{IL}	—	VCC × 0.2		
	EXTAL		V_{IH}	VCC × 0.8	—		
			V_{IL}	—	VCC × 0.2		
	5V-tolerant ports*3		V_{IH}	VCC × 0.8	5.8		
			V_{IL}	—	VCC × 0.2		
	RES, NMI, IRQ*4		V_{IH}	VCC × 0.8	—		
			V_{IL}	—	VCC × 0.2		
			ΔV_T*6	VCC × 0.10	—		VCC = 2.7 to 5.5 V
				VCC × 0.05	—		VCC = 1.6 to 2.7 V
	Peripheral functions*5	IIC (except for SMBus)*1	V_{IH}	VCC × 0.7	5.8*7		—
			V_{IL}	—	VCC × 0.3		
			ΔV_T*6	VCC × 0.10	—		VCC = 2.7 to 5.5 V
				VCC × 0.05	—		VCC = 1.6 to 2.7 V
		IIC (SMBus)*2	V_{IH}	2.2	—		VCC = 3.6 to 5.5 V
			V_{IL}	2.0	—		VCC = 2.7 to 3.6 V
			V_{IL}	—	0.8		VCC = 3.6 to 5.5 V
			V_{IL}	—	0.5		VCC = 2.7 to 3.6 V
		Other peripheral functions	V_{IH}	VCC × 0.8	—		—
			V_{IL}	—	VCC × 0.2		
			ΔV_T*6	VCC × 0.10	—		VCC = 2.7 to 5.5 V
				VCC × 0.05	—		VCC = 1.6 to 2.7 V

Note 1. SCL0_A, SDA0_A, SDA0_B (total 3 pins)
Note 2. SCL0_A, SCL0_C, SDA0_A, SDA0_B, SCL0_D, SDA0_C (total 6 pins)
Note 3. P400, P401 (total 2 pins)
Note 4. PmnPFS.ISEL = 1.
Note 5. PmnPFS.PMR = 1.
Note 6. This is the hysteresis characteristic of the Schmitt Trigger circuit.
Note 7. SDA0_B max spec is VCC.

35.2.3 I/O I_{OH}, I_{OL}

Table 35.5 I/O I_{OH}, I_{OL} (1 of 4)

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Permissible output current (max value per pin)	Ports P000 to P002, P010 to P015, P212, P213, P400, P401, P407	I_{OH}	—	—	-4.0	mA	
		I_{OL}	—	—	8.0	mA	
	Other output pins*1	I_{OH}	—	—	-4.0	mA	
		I_{OL}	—	—	20.0	mA	

Table 35.5 I/O I_{OH}, I_{OL} (2 of 4)

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Permissible output current (max value total pins)*2	48 pin products	Total of ports P000 to P002, P010 to P015	$\Sigma I_{OH\ (max)}$	—	—	-30	mA	AVCC0 = 2.7 to 5.5 V
				—	—	-8		AVCC0 = 1.8 to 2.7 V
				—	—	-4		AVCC0 = 1.6 to 1.8 V
			$\Sigma I_{OL\ (max)}$	—	—	50		AVCC0 = 2.7 to 5.5 V
				—	—	4		AVCC0 = 1.8 to 2.7 V
				—	—	2		AVCC0 = 1.6 to 1.8 V
		Total of ports P212, P213	$\Sigma I_{OH\ (max)}$	—	—	-8	mA	VCC = 2.7 to 5.5 V
				—	—	-2		VCC = 1.8 to 2.7 V
				—	—	-1		VCC = 1.6 to 1.8 V
			$\Sigma I_{OL\ (max)}$	—	—	16.0		VCC = 2.7 to 5.5 V
				—	—	1.2		VCC = 1.8 to 2.7 V
				—	—	0.6		VCC = 1.6 to 1.8 V
		Total of ports P206 to P208, P400 to P403, P407 to P409, P913 to P915	$\Sigma I_{OH\ (max)}$	—	—	-30	mA	VCC = 2.7 to 5.5 V
				—	—	-8		VCC = 1.8 to 2.7 V
				—	—	-4		VCC = 1.6 to 1.8 V
			$\Sigma I_{OL\ (max)}$	—	—	50		VCC = 2.7 to 5.5 V
				—	—	4		VCC = 1.8 to 2.7 V
				—	—	2		VCC = 1.6 to 1.8 V
		Total of ports P100 to P104, P108 to P112, P201, P300 to P302, P500	$\Sigma I_{OH\ (max)}$	—	—	-30	mA	VCC = 2.7 to 5.5 V
				—	—	-8		VCC = 1.8 to 2.7 V
				—	—	-4		VCC = 1.6 to 1.8 V
			$\Sigma I_{OL\ (max)}$	—	—	50		VCC = 2.7 to 5.5 V
				—	—	4		VCC = 1.8 to 2.7 V
				—	—	2		VCC = 1.6 to 1.8 V
		Total of all output pin	$\Sigma I_{OH\ (max)}$	—	—	-60	mA	—
			$\Sigma I_{OL\ (max)}$	—	—	100		—

Table 35.5 I/O I_{OH}, I_{OL} (3 of 4)

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Permissible output current (max value total pins)*2	32 pin products	Total of ports P010 to P015	$\Sigma I_{OH\ (max)}$	—	—	-24	mA	AVCC0 = 2.7 to 5.5 V
				—	—	-6		AVCC0 = 1.8 to 2.7 V
				—	—	-3		AVCC0 = 1.6 to 1.8 V
			$\Sigma I_{OL\ (max)}$	—	—	48		AVCC0 = 2.7 to 5.5 V
				—	—	3.6		AVCC0 = 1.8 to 2.7 V
				—	—	1.8		AVCC0 = 1.6 to 1.8 V
		Total of ports P212, P213	$\Sigma I_{OH\ (max)}$	—	—	-8	mA	VCC = 2.7 to 5.5 V
				—	—	-2		VCC = 1.8 to 2.7 V
				—	—	-1		VCC = 1.6 to 1.8 V
			$\Sigma I_{OL\ (max)}$	—	—	16.0		VCC = 2.7 to 5.5 V
				—	—	1.2		VCC = 1.8 to 2.7 V
				—	—	0.6		VCC = 1.6 to 1.8 V
		Total of other output ports	$\Sigma I_{OH\ (max)}$	—	—	-30	mA	VCC = 4.0 to 5.5 V
				—	—	-20		VCC = 2.7 to 4.0 V
				—	—	-12		VCC = 1.8 to 2.7 V
				—	—	-6		VCC = 1.6 to 1.8 V
			$\Sigma I_{OL\ (max)}$	—	—	50		VCC = 4.0 to 5.5 V
				—	—	20		VCC = 2.7 to 4.0 V
				—	—	8		VCC = 1.8 to 2.7 V
				—	—	4		VCC = 1.6 to 1.8 V
		Total of all output pin	$\Sigma I_{OH\ (max)}$	—	—	-54	mA	—
			$\Sigma I_{OL\ (max)}$	—	—	98		—

Table 35.5 I/O I_{OH}, I_{OL} (4 of 4)

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Permissible output current (max value total pins)*2	24 pin products	Total of ports P010, P011, P013, P014, P015	$\Sigma I_{OH\ (max)}$	—	—	-20	mA	AVCC0 = 2.7 to 5.5 V
				—	—	-5		AVCC0 = 1.8 to 2.7 V
				—	—	-2.5		AVCC0 = 1.6 to 1.8 V
			$\Sigma I_{OL\ (max)}$	—	—	40		AVCC0 = 2.7 to 5.5 V
				—	—	3		AVCC0 = 1.8 to 2.7 V
				—	—	1.5		AVCC0 = 1.6 to 1.8 V
		Total of ports P212, P213	$\Sigma I_{OH\ (max)}$	—	—	-8	mA	VCC = 2.7 to 5.5 V
				—	—	-2		VCC = 1.8 to 2.7 V
				—	—	-1		VCC = 1.6 to 1.8 V
			$\Sigma I_{OL\ (max)}$	—	—	16.0		VCC = 2.7 to 5.5 V
				—	—	1.2		VCC = 1.8 to 2.7 V
				—	—	0.6		VCC = 1.6 to 1.8 V
		Total of other output ports	$\Sigma I_{OH\ (max)}$	—	—	-30	mA	VCC = 4.0 to 5.5 V
				—	—	-20		VCC = 2.7 to 4.0 V
				—	—	-12		VCC = 1.8 to 2.7 V
				—	—	-6		VCC = 1.6 to 1.8 V
			$\Sigma I_{OL\ (max)}$	—	—	50	mA	VCC = 4.0 to 5.5 V
				—	—	20		VCC = 2.7 to 4.0 V
				—	—	8		VCC = 1.8 to 2.7 V
				—	—	4		VCC = 1.6 to 1.8 V
		Total of all output pin	$\Sigma I_{OH\ (max)}$	—	—	-50	mA	—
			$\Sigma I_{OL\ (max)}$	—	—	90		—

Note 1. Ports except for P200 are input.
Note 2. Specification under conditions where the duty factor ≤ 70%.
The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).
Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)
<Example> Where n = 80% and I_{OH} = −30.0 mA
Total output current of pins = (−30.0 × 0.7)/(80 × 0.01) ≅ −26.2 mA
However, the current that is allowed to flow into one pin does not vary depending on the duty factor.

Caution: To protect the reliability of the MCU, the output current values should not exceed the values in Table 35.5.

35.2.4 I/O V_{OH}, V_{OL}, and Other Characteristics

Table 35.6 I/O V_{OH}, V_{OL} (1)

Conditions: VCC = AVCC0 = 4.0 to 5.5 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage	Ports P000 to P002, P010 to P015	V_{OH}	AVCC0 - 0.8	—	—	V	I_{OH} = -4.0 mA
	Output pins except for P000 to P002 and P010 to P015*1	V_{OH}	VCC - 0.8	—	—		I_{OH} = -4.0 mA
	Ports P000 to P002, P010 to P015	V_{OL}	—	—	0.8		I_{OL} = 8.0 mA
	Ports P212, P213, P400, P401, P407	V_{OL}	—	—	0.8		I_{OL} = 8.0 mA
	Output pins except for P000 to P002, P010 to P015, P212, P213, P400, P401, and P407*1	V_{OL}	—	—	1.2		I_{OL} = 20.0 mA

Note 1. Ports except for P200 are input.

Table 35.7 I/O V_{OH}, V_{OL} (2)

Conditions: VCC = AVCC0 = 2.7 to 4.0 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage	Ports P000 to P002, P010 to P015	V_{OH}	AVCC0 - 0.8	—	—	V	I_{OH} = -4.0 mA
	Output pins except for P000 to P002 and P010 to P015*1	V_{OH}	VCC - 0.8	—	—		I_{OH} = -4.0 mA
	Ports P000 to P002, P010 to P015	V_{OL}	—	—	0.8		I_{OL} = 8.0 mA
	Output pins except for P000 to P002 and P010 to P015*1	V_{OL}	—	—	0.8		I_{OL} = 8.0 mA

Note 1. Ports except for P200 are input.

Table 35.8 I/O V_{OH}, V_{OL} (3)

Conditions: VCC = AVCC0 = 1.6 to 2.7 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage	Ports P000 to P002, P010 to P015	V_{OH}	AVCC0 - 0.5	—	—	V	I_{OH} = -1.0 mA AVCC0 = 1.8 to 2.7 V
			AVCC0 - 0.5	—	—		I_{OH} = -0.5 mA AVCC0 = 1.6 to 1.8 V
	Output pins except for P000 to P002 and P010 to P015*1	V_{OH}	VCC - 0.5	—	—		I_{OH} = -1.0 mA VCC = 1.8 to 2.7 V
			VCC - 0.5	—	—		I_{OH} = -0.5 mA VCC = 1.6 to 1.8 V
	Ports P000 to P002, P010 to P015	V_{OL}	—	—	0.4		I_{OL} = 0.6 mA AVCC0 = 1.8 to 2.7 V
			—	—	0.4		I_{OL} = 0.3 mA AVCC0 = 1.6 to 1.8 V
	Output pins except for P000 to P002 and P010 to P015*1	V_{OL}	—	—	0.4		I_{OL} = 0.6 mA VCC = 1.8 to 2.7 V
			—	—	0.4		I_{OL} = 0.3 mA VCC = 1.6 to 1.8 V

Note 1. Ports except for P200 are input.

Table 35.9 I/O other characteristics

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Input leakage current	RES, ports P200	$\lvert I_{in} \rvert$	—	—	1.0	μA	V_{in} = 0 V V_{in} = VCC
Three-state leakage current (off state)	5V-tolerant ports*1	$\lvert I_{TSI} \rvert$	—	—	1.0	μA	V_{in} = 0 V V_{in} = 5.8 V
	Other ports (except for P200, and 5V-tolerant ports)		—	—	1.0		V_{in} = 0 V V_{in} = VCC
Input pull-up resistor	All ports (except for P200)	R_U	10	20	100	kΩ	V_{in} = 0 V
Input capacitance	P200	C_{in}	—	—	30	pF	V_{in} = 0 V f = 1 MHz T_a = 25°C
	Other input pins		—	—	15		

Note 1. P400, P401 (total 2 pins)

35.2.5 Operating and Standby Current

Table 35.10 Operating and standby current (1) (1 of 2)

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter					Symbol	Typ*10	Max	Unit	Test Conditions
Supply current*1	High-speed mode*2	Normal mode	All peripheral clocks disabled, CoreMark code executing from flash*5	ICLK = 64 MHz	I_{CC}	6.80	—	mA	*7 *11
				ICLK = 48 MHz		5.20	—		*7 *11
				ICLK = 32 MHz		3.60	—		*7
				ICLK = 16 MHz		2.15	—		
				ICLK = 8 MHz		1.40	—		
			All peripheral clocks enabled, code executing from flash*5	ICLK = 64 MHz		—	13.9		*9 *11
		Sleep mode	All peripheral clocks disabled*5	ICLK = 64 MHz		1.50	—		*7
				ICLK = 48 MHz		1.20	—		*7
				ICLK = 32 MHz		0.95	—		*7
				ICLK = 16 MHz		0.70	—		
				ICLK = 8 MHz		0.60	—		
			All peripheral clocks enabled*5	ICLK = 64 MHz		5.65	—		*9
				ICLK = 48 MHz		4.30	—		*9
				ICLK = 32 MHz		3.65	—		*8
				ICLK = 16 MHz		2.10	—		
				ICLK = 8 MHz		1.30	—		
		Increase during BGO operation*6				2.1	—		—

Table 35.10 Operating and standby current (1) (2 of 2)

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter					Symbol	Typ*10	Max	Unit	Test Conditions
Supply current*1	Middle-speed mode*2	Normal mode	All peripheral clocks disabled, CoreMark code executing from flash*5	ICLK = 24 MHz	I_{CC}	2.75	—	mA	*7
				ICLK = 4 MHz		0.90	—		
			All peripheral clocks enabled, code executing from flash*5	ICLK = 24 MHz		—	6.6		*8
		Sleep mode	All peripheral clocks disabled*5	ICLK = 24 MHz		0.75	—		*7
				ICLK = 4 MHz		0.55	—		
			All peripheral clocks enabled*5	ICLK = 24 MHz		2.80	—		*8
				ICLK = 4 MHz		0.90	—		
		Increase during BGO operation*6				1.80	—		—
	Low-speed mode*3	Normal mode	All peripheral clocks disabled, CoreMark code executing from flash*5	ICLK = 2 MHz		0.35	—	mA	*7
			All peripheral clocks enabled, code executing from flash*5	ICLK = 2 MHz		—	1.7		*8
		Sleep mode	All peripheral clocks disabled*5	ICLK = 2 MHz		0.13	—		*7
			All peripheral clocks enabled*5	ICLK = 2 MHz		0.30	—		*8
	Subosc-speed mode*4	Normal mode	All peripheral clocks enabled, code executing from flash*5	ICLK = 32.768 kHz		—	341	µA	*8
		Sleep mode	All peripheral clocks disabled*5	ICLK = 32.768 kHz		1.60	—		*8
			All peripheral clocks enabled*5	ICLK = 32.768 kHz		4.15	—		*8

Note 1. Supply current is the total current flowing into VCC. Supply current values apply when internal pull-up MOSs are in the off state and these values do not include output charge/discharge current from any of the pins.
Note 2. The clock source is HOCO.
Note 3. The clock source is MOCO.
Note 4. The clock source is LOCO.
Note 5. This does not include BGO operation.
Note 6. This is the increase for programming or erasure of the flash memory for data storage during program execution.
Note 7. PCLKB and PCLKD are set to divided by 64.
Note 8. PCLKB and PCLKD are the same frequency as that of ICLK.
Note 9. PCLKB are set to be divided by 2 and PCLKD is the same frequency as that of ICLK.
Note 10. VCC = 3.3 V.
Note 11. The prefetch buffer is operating.

Table 35.11 Operating and standby current (2)

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter					Symbol	Typ*3	Max	Unit	Test conditions
Supply current*1	Software Standby mode*2	Peripheral modules stop	All SRAMs (0x2000_4000 to 0x2000_5FFF) are on	T_a = 25°C	I_{CC}	0.40	4.4	µA	—
				T_a = 55°C		1.10	11		
				T_a = 85°C		3.85	33		
				T_a = 105°C		8.80	71		
				T_a = 125°C		19.1	159		

Note 1. Supply current is the total current flowing into VCC. Supply current values apply when internal pull-up MOSs are in the off state and these values do not include output charge/discharge current from any of the pins.
Note 2. The IWDT and LVD are not operating.
Note 3. VCC = 3.3 V.

Table 35.12 Operating and standby current (3)

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Analog power supply current	During 12-bit A/D conversion (at high-speed A/D conversion mode)	I_{AVCC0}	—	—	1.44	mA	—
	During 12-bit A/D conversion (at high-speed A/D conversion mode) with S/H (2.7 V to 5.5V)		—	—	5.00	mA	—
	During 12-bit A/D conversion (at low-power A/D conversion mode)		—	—	0.78	mA	—
	Waiting for 12-bit A/D conversion (all units)*1		—	—	1.0	µA	—
Reference power supply current	During 12-bit A/D conversion	I_{REFH0}	—	—	120	µA	—
	Waiting for 12-bit A/D conversion		—	—	60	nA	—
Temperature Sensor (TSN) operating current		I_{TNS}	—	95	—	µA	—
High-Speed Analog Comparator (ACMPHS) operating current per channel		I_{CMPHS}	—	100	—	µA	—

Note 1. When the MCU is in Software Standby mode or the MSTPCRD.MSTPD16 (ADC120 module-stop bit) is in the module-stop state.

35.2.6 VCC Rise, Fall Gradient, and Ripple Frequency

Table 35.13 Rise and fall gradient characteristics

Conditions: VCC = AVCC0 = 0 to 5.5 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Power-on VCC rising gradient	Voltage monitor 0 reset disabled at startup	SrVCC	0.02	—	2	ms/V	—
	Voltage monitor 0 reset enabled at startup*1 *2				—		
	SCI boot mode*2				2		

Note 1. When OFS1.LVDAS = 0.
Note 2. At boot mode, the reset from voltage monitor 0 is disabled regardless of the value of OFS1.LVDAS bit.

Table 35.14 Rising and falling gradient and ripple frequency characteristics

Conditions: VCC = AVCC0 = 1.6 to 5.5 V
The ripple voltage must meet the allowable ripple frequency $f_{r(VCC)}$ within the range between the VCC upper limit (5.5 V) and lower limit (1.6 V).
When the VCC change exceeds VCC ± 10%, the allowable voltage change rising and falling gradient dt/dVCC must be met.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Allowable ripple frequency	$f_{r(VCC)}$	—	—	10	kHz	Figure 35.2 $V_{r\ (VCC)} \leq VCC \times 0.2$
		—	—	1	MHz	Figure 35.2 $V_{r\ (VCC)} \leq VCC \times 0.08$
		—	—	10	MHz	Figure 35.2 $V_{r\ (VCC)} \leq VCC \times 0.06$
Allowable voltage change rising and falling gradient	dt/dVCC	1.0	—	—	ms/V	When VCC change exceeds VCC ± 10%

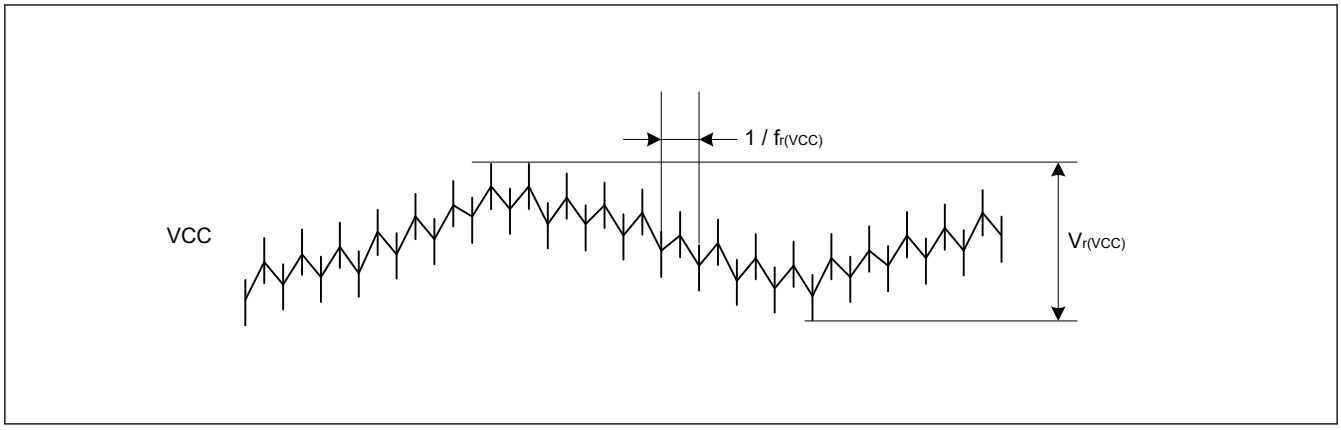


Figure 35.2 Ripple waveform

35.3 AC Characteristics

35.3.1 Frequency

Table 35.15 Operation frequency in high-speed operating mode

Conditions: VCC = AVCC0 = 1.8 to 5.5 V

Parameter			Symbol	Min	Typ	Max*5	Unit
Operation frequency	System clock (ICLK)*1*2*4	1.8 to 5.5 V	f	0.032768	—	64	MHz
	Peripheral module clock (PCLKB) *4	1.8 to 5.5 V		—	—	32	
	Peripheral module clock (PCLKD)*3 *4	1.8 to 5.5 V		—	—	64	

Note 1. The lower-limit frequency of ICLK is 1 MHz while programming or erasing the flash memory. When using ICLK for programming or erasing the flash memory at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note 2. The frequency accuracy of ICLK must be ± 1.0% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.
Note 3. The lower-limit frequency of PCLKD is 1 MHz when the ADC12 is in use.
Note 4. See section 8, Clock Generation Circuit for the relationship of frequencies between ICLK, PCLKB, and PCLKD.
Note 5. The maximum value of operation frequency does not include internal oscillator errors. For details on the range for guaranteed operation, see Table 35.19.

Table 35.16 Operation frequency in middle-speed mode

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max*5	Unit
Operation frequency	System clock (ICLK)*1*2*4	1.8 to 5.5 V	f	0.032768	—	24	MHz
		1.6 to 1.8 V		0.032768	—	4	
	Peripheral module clock (PCLKB) *4	1.8 to 5.5 V		—	—	24	
		1.6 to 1.8 V		—	—	4	
	Peripheral module clock (PCLKD)*3 *4	1.8 to 5.5 V		—	—	24	
		1.6 to 1.8 V		—	—	4	

Note 1. The lower-limit frequency of ICLK is 1 MHz while programming or erasing the flash memory. When using ICLK for programming or erasing the flash memory at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note 2. The frequency accuracy of ICLK must be ± 1.0% while programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.
Note 3. The lower-limit frequency of PCLKD is 1 MHz when the ADC12 is in use.
Note 4. See section 8, Clock Generation Circuit for the relationship of frequencies between ICLK, PCLKB, and PCLKD.
Note 5. The maximum value of operation frequency does not include internal oscillator errors. For details on the range for guaranteed operation, see Table 35.19.

Table 35.17 Operation frequency in low-speed mode

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max*5	Unit
Operation frequency	System clock (ICLK)*1*2*4	1.6 to 5.5 V	f	0.032768	—	2	MHz
	Peripheral module clock (PCLKB) *4	1.6 to 5.5 V		—	—	2	
	Peripheral module clock (PCLKD)*3 *4	1.6 to 5.5 V		—	—	2	

Note 1. The lower-limit frequency of ICLK is 1 MHz while programming or erasing the flash memory.
Note 2. The frequency accuracy of ICLK must be ± 1.0% while programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.
Note 3. The lower-limit frequency of PCLKD is 1 MHz when the ADC12 is in use.
Note 4. See section 8, Clock Generation Circuit for the relationship of frequencies between ICLK, PCLKB, and PCLKD.
Note 5. The maximum value of operation frequency does not include internal oscillator errors. For details on the range for guaranteed operation, see Table 35.19.

Table 35.18 Operation frequency in Subosc-speed mode

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max	Unit
Operation frequency	System clock (ICLK)*1*3	1.6 to 5.5 V	f	27.8528	32.768	37.6832	kHz
	Peripheral module clock (PCLKB) *3	1.6 to 5.5 V		—	—	37.6832	
	Peripheral module clock (PCLKD)*2 *3	1.6 to 5.5 V		—	—	37.6832	

Note 1. Programming and erasing the flash memory is not possible.
Note 2. The ADC12 cannot be used.
Note 3. See section 8, Clock Generation Circuit for the relationship of frequencies between ICLK, PCLKB, and PCLKD.

35.3.2 Clock Timing

Table 35.19 Clock timing (1 of 2)

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
EXTAL external clock input cycle time	t_{Xcyc}	50	—	—	ns	Figure 35.3
EXTAL external clock input high pulse width	t_{XH}	20	—	—	ns	
EXTAL external clock input low pulse width	t_{XL}	20	—	—	ns	
EXTAL external clock rising time	t_{Xr}	—	—	5	ns	
EXTAL external clock falling time	t_{Xf}	—	—	5	ns	
EXTAL external clock input wait time*1	t_{EXWT}	0.3	—	—	μs	—
EXTAL external clock input frequency	f_{EXTAL}	—	—	20	MHz	1.8 ≤ VCC ≤ 5.5
		—	—	4		1.6 ≤ VCC < 1.8
Main clock oscillator oscillation frequency	f_{MAIN}	1	—	20	MHz	1.8 ≤ VCC ≤ 5.5
		1	—	4		1.6 ≤ VCC < 1.8
LOCO clock oscillation frequency	f_{LOCO}	27.8528	32.768	37.6832	kHz	—
LOCO clock oscillation stabilization time	t_{LOCO}	—	—	100	μs	Figure 35.4
IWDT-dedicated clock oscillation frequency	f_{ILOCO}	12.75	15	17.25	kHz	—
MOCO clock oscillation frequency	f_{MOCO}	6.8	8	9.2	MHz	—
MOCO clock oscillation stabilization time	t_{MOCO}	—	—	1	μs	—

Table 35.19 Clock timing (2 of 2)

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
HOCO clock oscillation frequency*4	f_{HOCO24}	23.76	24	24.24	MHz	Ta = -40 to 105°C 1.6 ≤ VCC ≤ 5.5
	f_{HOCO32}	31.68	32	32.32		Ta = -40 to 105°C 1.6 ≤ VCC ≤ 5.5
	f_{HOCO48}	47.52	48	48.48		Ta = -40 to 105°C 1.6 ≤ VCC ≤ 5.5
	f_{HOCO64}	63.36	64	64.64		Ta = -40 to 105°C 1.6 ≤ VCC ≤ 5.5
HOCO clock oscillation stabilization time*2 *3	t_{HOCO24} t_{HOCO32} t_{HOCO48} t_{HOCO64}	—	6.7	7.7	µs	Figure 35.5

Note 1. Time until the clock can be used after the Main Clock Oscillator stop bit (MOSCCR.MOSTP) is set to 0 (operating) when the external clock is stable.
Note 2. This is a characteristic when the HOCOCR.HCSTP bit is set to 0 (oscillation) in the MOCO stop state. When the HOCOCR.HCSTP bit is set to 0 (oscillation) during MOCO oscillation, this specification is shortened by 1 µs.
Note 3. Check OSCSF.HOCOSF to confirm whether stabilization time has elapsed.
Note 4. Accuracy at production test.

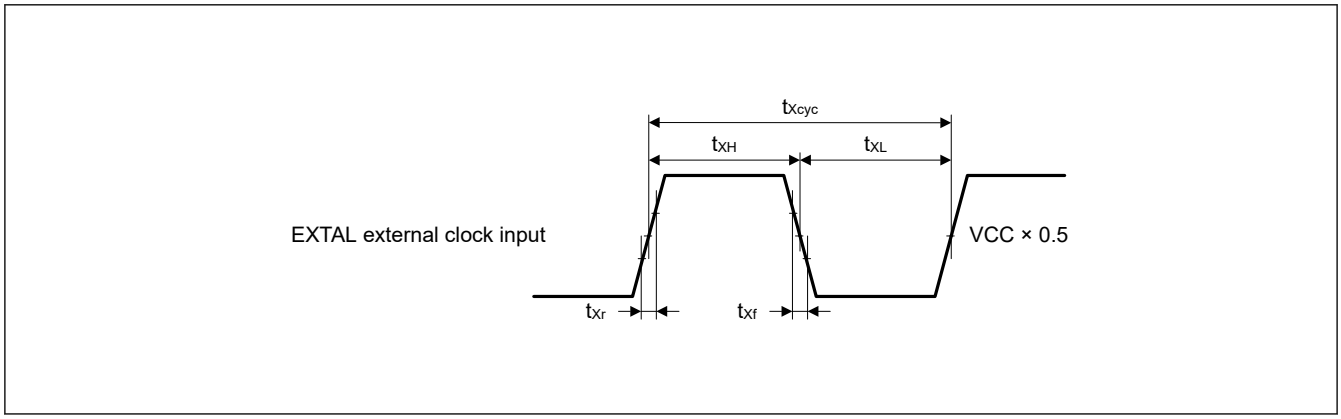


Figure 35.3 EXTAL external clock input timing

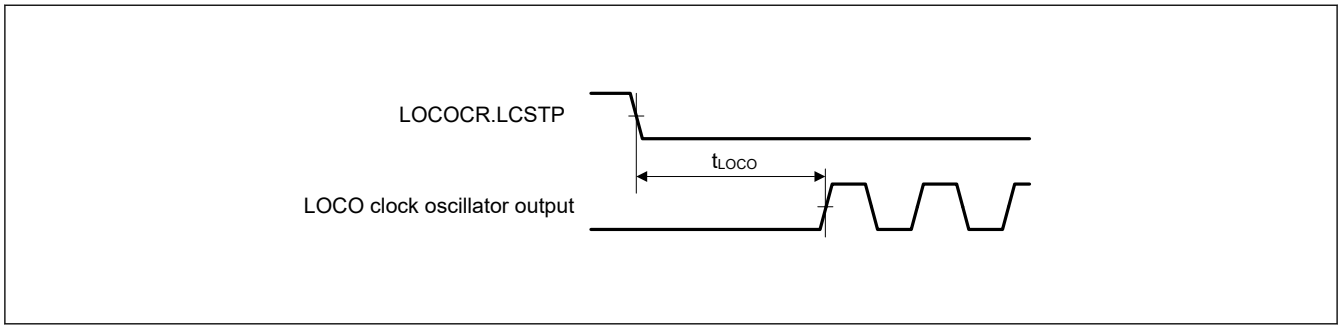
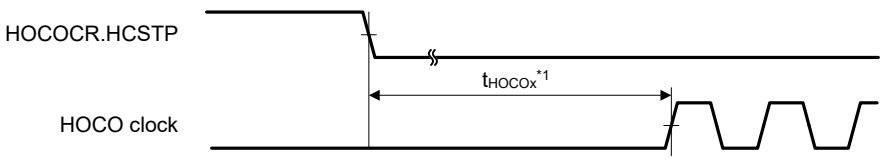


Figure 35.4 LOCO clock oscillation start timing



Note: x = 24, 32, 48, 64

Figure 35.5 HOCO clock oscillation start timing (started by setting the HOCOCR.HCSTP bit)

35.3.3 Reset Timing

Table 35.20 Reset timing

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
RES pulse width	At power-on	t_{RESWP}	10	—	—	ms	Figure 35.6
	Not at power-on	t_{RESW}	30	—	—	μs	Figure 35.7
Wait time after RES cancellation (at power-on)	LVD0 enabled*1	t_{RESWT}	—	0.9	—	ms	Figure 35.6
	LVD0 disabled*2		—	0.2	—		
Wait time after RES cancellation (during powered-on state)	LVD0 enabled*1	t_{RESWT2}	—	0.9	—	ms	Figure 35.7
	LVD0 disabled*2		—	0.2	—		
Wait time after internal reset cancellation (Watchdog timer reset, SRAM parity error reset, bus master MPU error reset, bus slave MPU error reset, stack pointer error reset, software reset)	LVD0 enabled*1	t_{RESWT3}	—	0.9	—	ms	Figure 35.8
	LVD0 disabled*2		—	0.15	—		

Note 1. When OFS1.LVDAS = 0.
Note 2. When OFS1.LVDAS = 1.

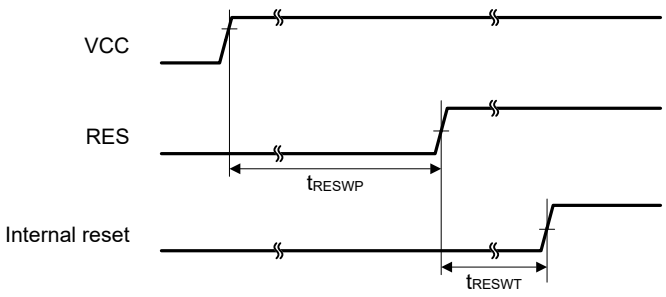


Figure 35.6 Reset input timing at power-on

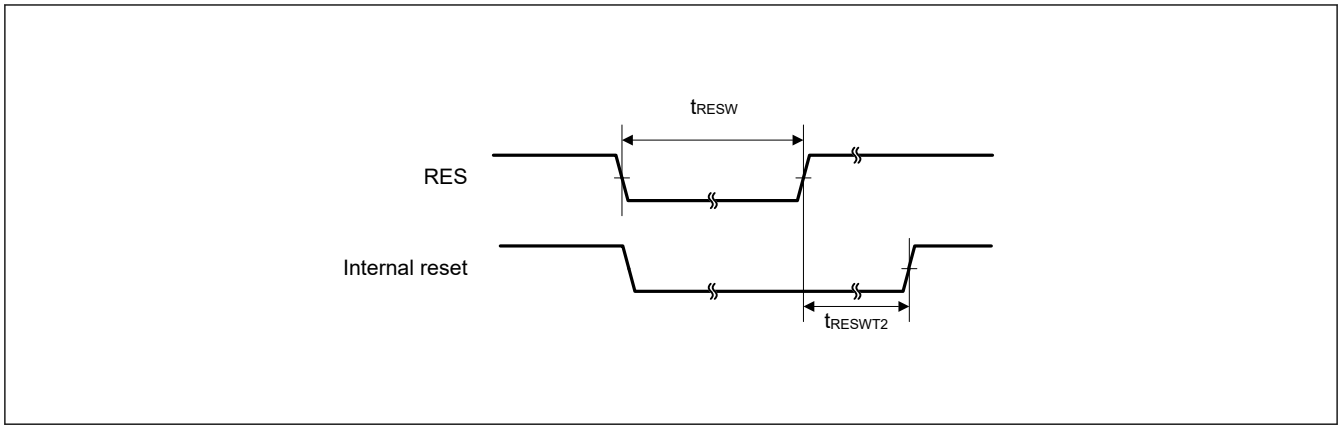


Figure 35.7 Reset input timing (1)

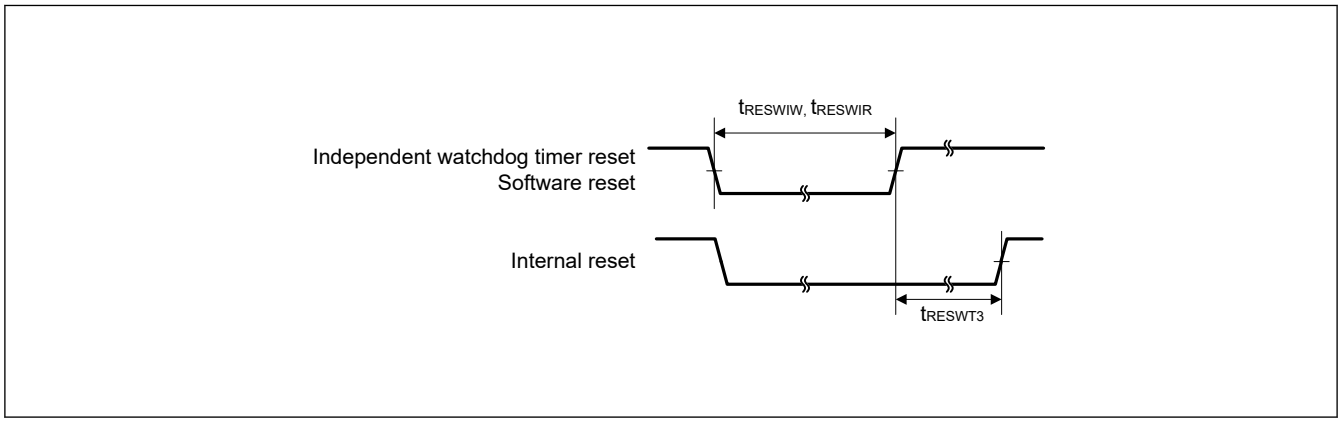


Figure 35.8 Reset input timing (2)

35.3.4 Wakeup Time

Table 35.21 Timing of recovery from low power modes (1)

Parameter				Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Software Standby mode*1	High-speed mode	Crystal resonator connected to main clock oscillator	System clock source is main clock oscillator (20 MHz)*2	t_{SBYMC}	—	2	3	ms	
		External clock input to main clock oscillator	System clock source is main clock oscillator (20 MHz)*3	t_{SBYEX}	—	2.4	3.1	µs	
		System clock source is HOCO (HOCO clock is 32 MHz)*4		t_{SBYHO}	—	7.4	9.1	µs	Figure 35.9
		System clock source is HOCO (HOCO clock is 48 MHz)*5		t_{SBYHO}	—	7.3	8.9	µs	
		System clock source is HOCO (HOCO clock is 64 MHz)*6		t_{SBYHO}	—	7.3	8.8	µs	
		System clock source is MOCO (8 MHz)		t_{SBYMO}	—	4	5	µs	

Note 1. The division ratio of ICLK and PCLKx is the minimum division ratio within the allowable frequency range. The recovery time is determined by the system clock source.
Note 2. The Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x05.
Note 3. The Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x00.
Note 4. The system clock is 32 MHz.
Note 5. The system clock is 48 MHz.
Note 6. The system clock is 64 MHz.

Table 35.22 Timing of recovery from low power modes (2)

Parameter				Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Software Standby mode*1	Middle-speed mode	Crystal resonator connected to main clock oscillator	System clock source is main clock oscillator (20 MHz)*2	t_{SBYMC}	—	2	3	ms	
		External clock input to main clock oscillator	System clock source is main clock oscillator (20 MHz)*3 VCC = 1.8 V to 5.5 V	t_{SBYEX}	—	2.4	3.1	µs	
			System clock source is main clock oscillator (4 MHz)*3 VCC = 1.6 V to 1.8 V		—	8.5	9.1		
		System clock source is HOCO*4	VCC = 1.8 V to 5.5 V*4	t_{SBYHO}	—	7.7	9.4	µs	Figure 35.9
			VCC = 1.6 V to 1.8 V		—	15.7	17.9		
		System clock source is MOCO (8 MHz)	VCC = 1.8 V to 5.5 V	t_{SBYMO}	—	4	5	µs	
			VCC = 1.6 V to 1.8 V		—	7.2	9		

Note 1. The division ratio of ICLK and PCLKx is the minimum division ratio within the allowable frequency range. The recovery time is determined by the system clock source.
Note 2. The Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x05.
Note 3. The Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x00.
Note 4. The system clock is 24 MHz.

Table 35.23 Timing of recovery from low power modes (3)

Parameter				Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Software Standby mode*1	Low-speed mode	Crystal resonator connected to main clock oscillator	System clock source is main clock oscillator (2 MHz)*2	t_{SBYMC}	—	2	3	ms	Figure 35.9
		External clock input to main clock oscillator	System clock source is main clock oscillator (2 MHz)*3	t_{SBYEX}	—	14.5	16	µs	
		System clock source is MOCO (8 MHz)		t_{SBYMO}	—	12	15	µs	

Note 1. The division ratio of ICLK and PCLKx is the minimum division ratio within the allowable frequency range. The recovery time is determined by the system clock source.
Note 2. The Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x05.
Note 3. The Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x00.

Table 35.24 Timing of recovery from low power modes (4)

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Software Standby mode*1	Subosc-speed mode	System clock source is LOCO (32.768 kHz)	t_{SBYLO}	—	0.85	1.2	ms	Figure 35.9

Note 1. LOCO itself continues oscillating in Software Standby mode during Subosc-speed mode.

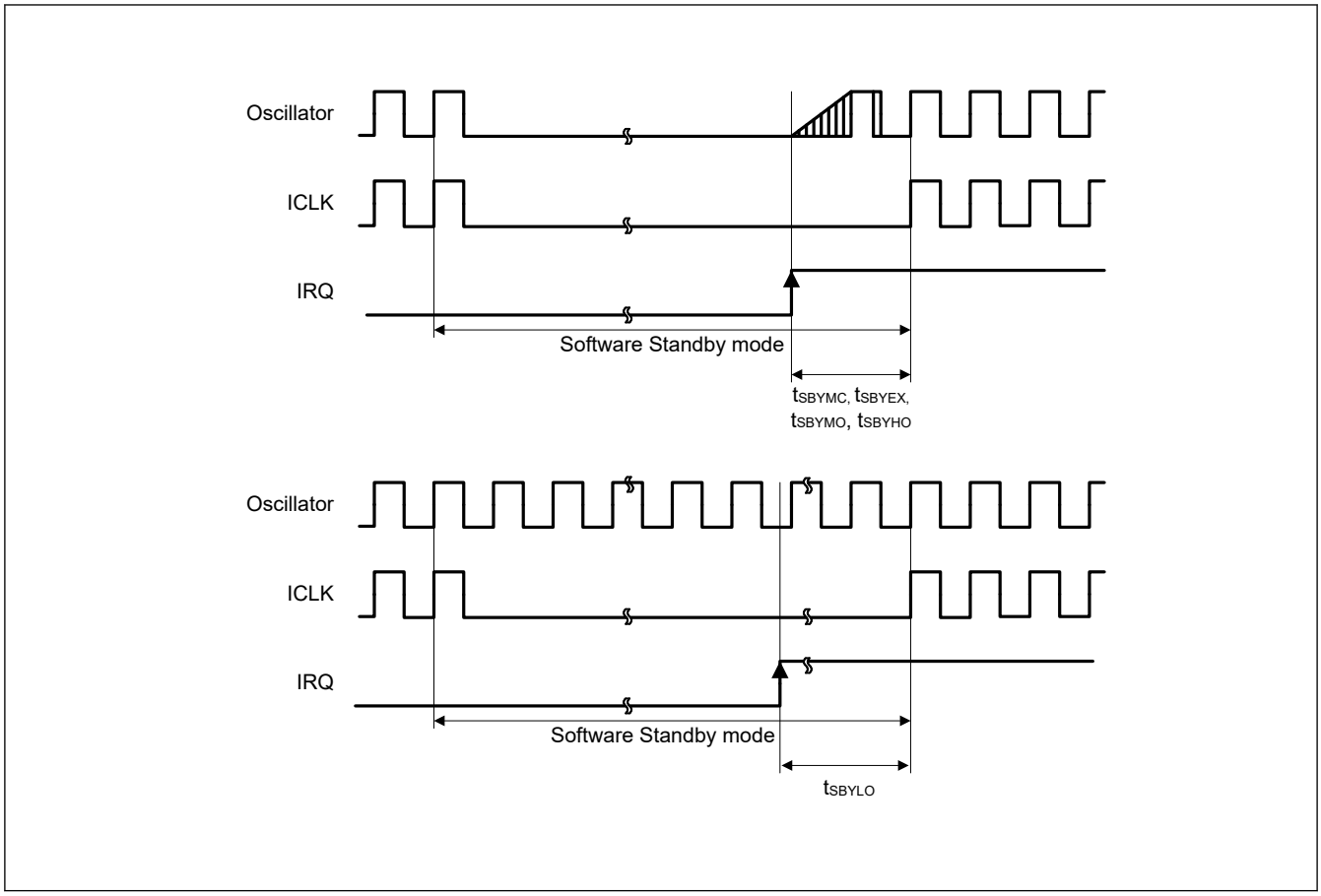
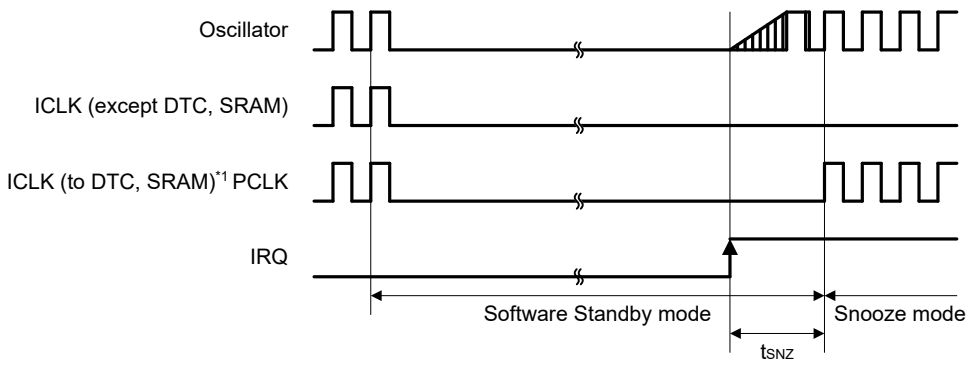


Figure 35.9 Software Standby mode cancellation timing

Table 35.25 Timing of recovery from low power modes (5)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Software Standby mode to Snooze mode	High-speed mode System clock source is HOCO	t_{SNZ}	—	6.6	8.1	µs	Figure 35.10
	Middle-speed mode System clock source is HOCO (24 MHz) VCC = 1.8 V to 5.5 V	t_{SNZ}	—	6.7	8.2	µs	
	Middle-speed mode System clock source is HOCO (24 MHz) VCC = 1.6 V to 1.8 V	t_{SNZ}	—	10.8	12.9	µs	
	Low-speed mode System clock source is MOCO (2 MHz)	t_{SNZ}	—	6.7	8.0	µs	



Note 1. When SNZCR.SNZDTCEN bit is set to 1, ICLK is supplied to DTC and SRAM.

Figure 35.10 Recovery timing from Software Standby mode to Snooze mode

35.3.5 NMI and IRQ Noise Filter

Table 35.26 NMI and IRQ noise filter

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions	
NMI pulse width	t_{NMIW}	200	—	—	ns	NMI digital filter disabled	$t_{Pcyc} \times 2 \leq 200$ ns
		$t_{Pcyc} \times 2$*1	—	—			$t_{Pcyc} \times 2 > 200$ ns
		200	—	—		NMI digital filter enabled	$t_{NMICK} \times 3 \leq 200$ ns
		$t_{NMICK} \times 3.5$*2	—	—			$t_{NMICK} \times 3 > 200$ ns
IRQ pulse width	t_{IRQW}	200	—	—	ns	IRQ digital filter disabled	$t_{Pcyc} \times 2 \leq 200$ ns
		$t_{Pcyc} \times 2$*1	—	—			$t_{Pcyc} \times 2 > 200$ ns
		200	—	—		IRQ digital filter enabled	$t_{IRQCK} \times 3 \leq 200$ ns
		$t_{IRQCK} \times 3.5$*3	—	—			$t_{IRQCK} \times 3 > 200$ ns

Note: 200 ns minimum in Software Standby mode.
Note: If the clock source is being switched it is needed to add 4 clock cycle of switched source.
Note 1. t_{Pcyc} indicates the PCLKB cycle.
Note 2. t_{NMICK} indicates the cycle of the NMI digital filter sampling clock.
Note 3. t_{IRQCK} indicates the cycle of the IRQi digital filter sampling clock (i = 0 to 7).

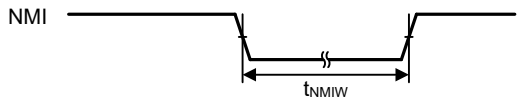


Figure 35.11 NMI interrupt input timing

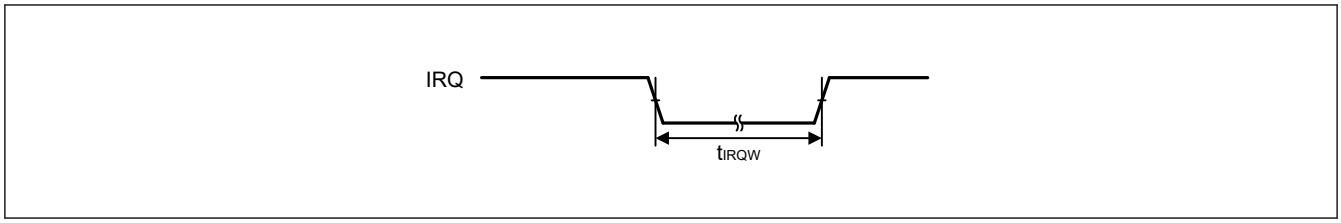


Figure 35.12 IRQ interrupt input timing

35.3.6 I/O Ports, POEG, GPT, AGTW, KINT, and ADC12 Trigger Timing

Table 35.27 I/O Ports, POEG, GPT, AGTW, KINT, and ADC12 trigger timing (1 of 3)

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
I/O Ports	Input data pulse width	2.7 V ≤ VCC ≤ 5.5 V	t_{PRW}	3	—	—	t_{Pcyc}	Figure 35.13
		2.4 V ≤ VCC < 2.7 V		4	—			
		1.6 V ≤ VCC < 2.4 V		5	—			

Table 35.27 I/O Ports, POEG, GPT, AGTW, KINT, and ADC12 trigger timing (2 of 3)

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
POEG	GTETRGn input pulse width		t_{POEW}	1.5	—	—	t_{Pcyc}	Figure 35.14
	Output disable time	Input level detection of the GTETRGn pin (via flag)	t_{POEGDI}	—	—	3PCLKB + 0.34	μs	Figure 35.15 When the digital noise filter is not in use (POEGGn.NFEN = 0 (n = A to D))
		Detection of the output stopping signal from GPT (deadtime error, simultaneous high output, or simultaneous low output)	t_{POEGDE}	—	—	0.5	μs	Figure 35.16
		Edge detection signal from a comparator	t_{POEGDC}	—	—	3PCLKB + 0.5	μs	Figure 35.17 The time is that when the noise filter for ACMPHS is not in use (COMPFIR0.CiFCK[1:0] = 00) and excludes the time for detection by ACMPHS.
		Resister setting	t_{POEGDS}	—	—	1PCLKB + 0.3	μs	Figure 35.18 Time for access to the register is not included.
		Oscillation stop detection*2	$t_{POEGDOS}$	—	≤ 1	—	μs	Figure 35.19
		Input level detection of the GTETRGn pin (direct path)	$t_{POEGDDI}$	—	—	2PCLKB + 1PCLKD + 0.34	μs	Figure 35.20
		Level detection signal from a comparator	$t_{POEGDDC}$	—	—	1PCLKD + 0.3	μs	Figure 35.21 The time is that when the noise filter for ACMPHS is not in use (COMPFIR0.CiFCK[1:0] = 00) and excludes the time for detection by ACMPHS.

Table 35.27 I/O Ports, POEG, GPT, AGTW, KINT, and ADC12 trigger timing (3 of 3)

Parameter				Symbol	Min	Typ	Max	Unit	Test conditions
GPT	Input capture pulse width	Single edge		t_{GTICW}	1.5	—	—	t_{PDcyc}	Figure 35.22
		Dual edge			2.5	—	—		
	GTIOCxY output skew (x = 0 to 3, Y = A or B)			t_{GTISK}	—	—	4	ns	Figure 35.23
	GTIOCxY output skew (x = 0 to 3, Y = A or B)				—	—	6		
	OPS output skew GTOUUP, GTOULO, GTOVUP, GTOVLO, GTOWUP, GTOWLO			t_{GTOSK}	—	—	5	ns	Figure 35.24
	External trigger input pulse width	Synchronous clock	Single-edge setting	t_{GTEW}	1.5	—	—	t_{Pcyc}	Figure 35.25
			Both-edge setting		2.5	—	—		
	Timer clock pulse width	Synchronous clock	Single-edge setting	t_{GTCKWH}, t_{GTCKWL}	1.5	—	—	t_{Pcyc}	Figure 35.26
			Both-edge setting		2.5	—	—		
GPT (PWM Delay Generation Circuit)	GTIOCxY_Z skew (x = 0 to 3, Y = A or B, Z = A to D)			t_{HRSK}*1	—	—	4.0	ns	Figure 35.27
AGTW	AGTIO, AGTEE input cycle	1.8 V ≤ VCC ≤ 5.5 V		t_{ACYC}*1	250	—	—	ns	Figure 35.28
		1.6 V ≤ VCC < 1.8 V			2000	—	—	ns	
	AGTIO, AGTEE input high-level width, low-level width	1.8 V ≤ VCC ≤ 5.5 V		t_{ACKWH}, t_{ACKWL}	100	—	—	ns	
		1.6 V ≤ VCC < 1.8 V			800	—	—	ns	
	AGTIO, AGTO, AGTOA, AGTOB output cycle	2.7 V ≤ VCC ≤ 5.5 V		t_{ACYC2}	62.5	—	—	ns	Figure 35.28
		2.4 V ≤ VCC < 2.7 V			125	—	—	ns	
		1.8 V ≤ VCC < 2.4 V			250	—	—	ns	
		1.6 V ≤ VCC < 1.8 V			500	—	—	ns	
ADC12	12-bit A/D converter trigger input pulse width			t_{TRGW}	1.5	—	—	t_{Pcyc}	Figure 35.29
KINT	KRn (n = 00 to 04) pulse width			t_{KR}	250	—	—	ns	Figure 35.30

Note 1. Constraints on input cycle: t_{Pcyc} × 2 (t_{Pcyc}: PCLKB cycle) < t_{ACYC}

Note 2. Reference value

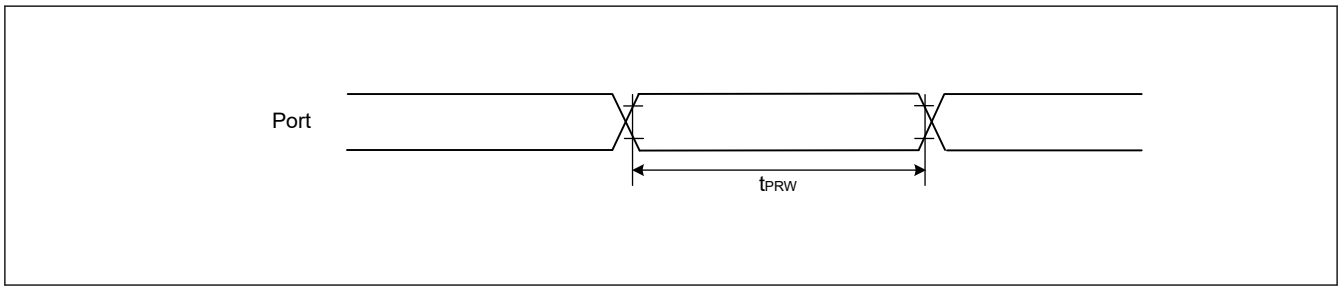


Figure 35.13 I/O ports input timing

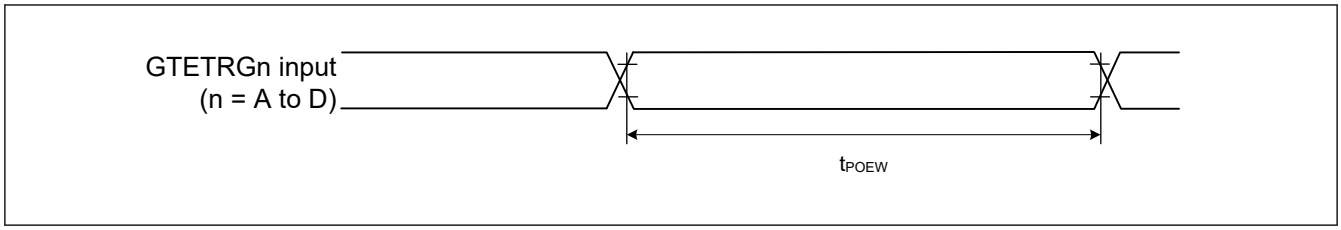


Figure 35.14 POEG input trigger timing

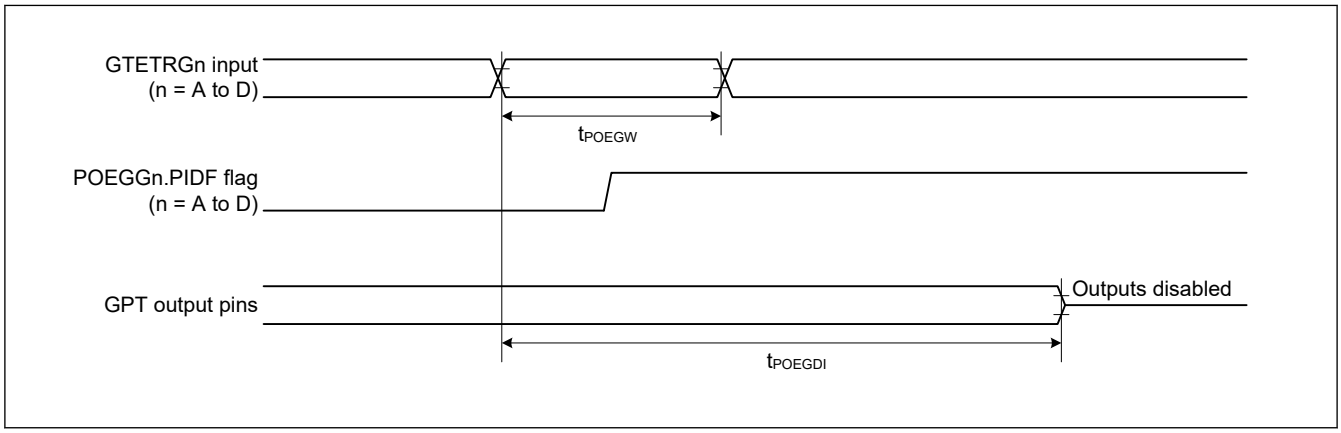
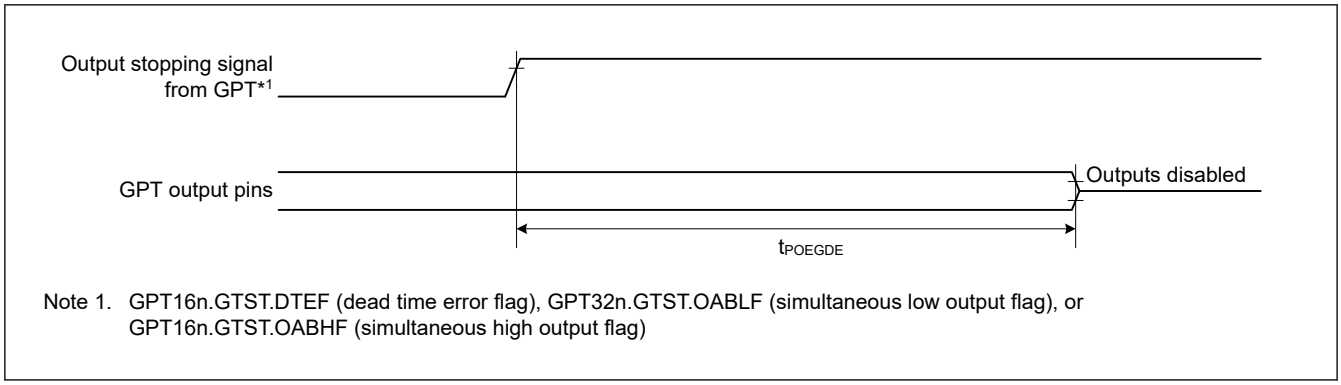


Figure 35.15 Output disable time for POEG via detection flag in response to the input level detection of the GTETRGn pin



Note 1. GPT16n.GTST.DTEF (dead time error flag), GPT32n.GTST.OABLF (simultaneous low output flag), or GPT16n.GTST.OABHF (simultaneous high output flag)

Figure 35.16 Output disable time for POEG in response to detection of the output stopping signal from GPT

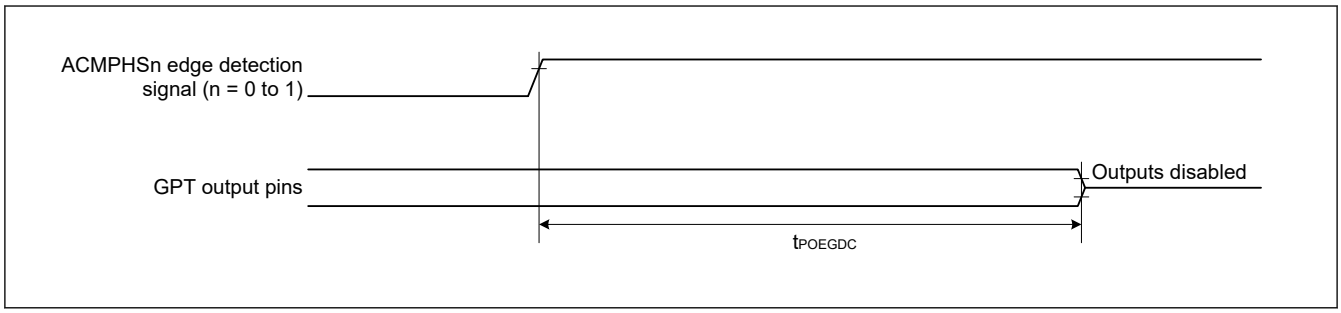


Figure 35.17 Output disable time for POEG in response to edge detection signal from ACMPHS

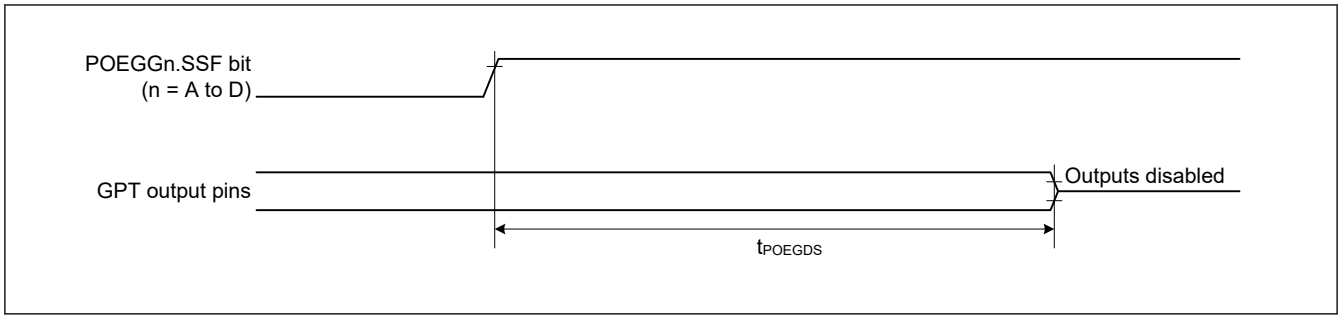


Figure 35.18 Output disable time for POEG in response to the register setting

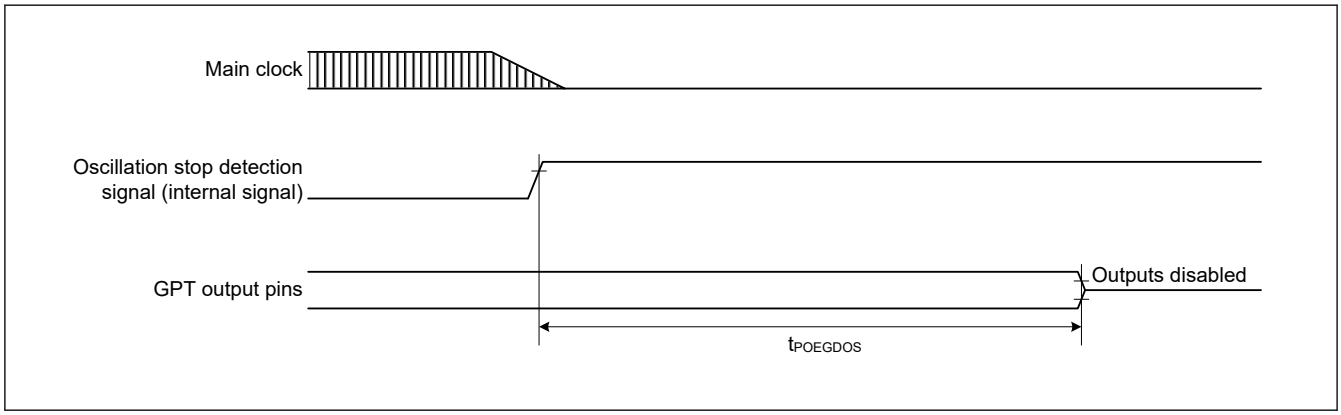


Figure 35.19 Output disable time of POEG in response to the oscillation stop detection

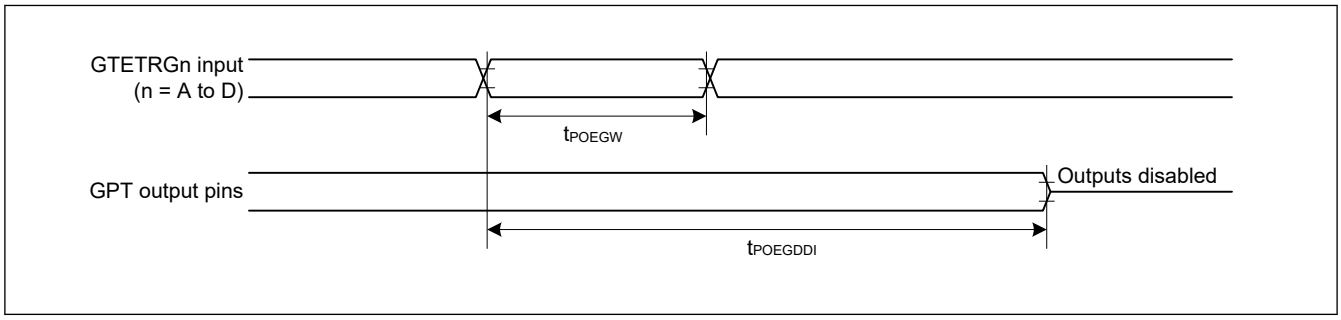


Figure 35.20 Output disable time for POEG in direct response to the input level detection of the GTETRGn pin

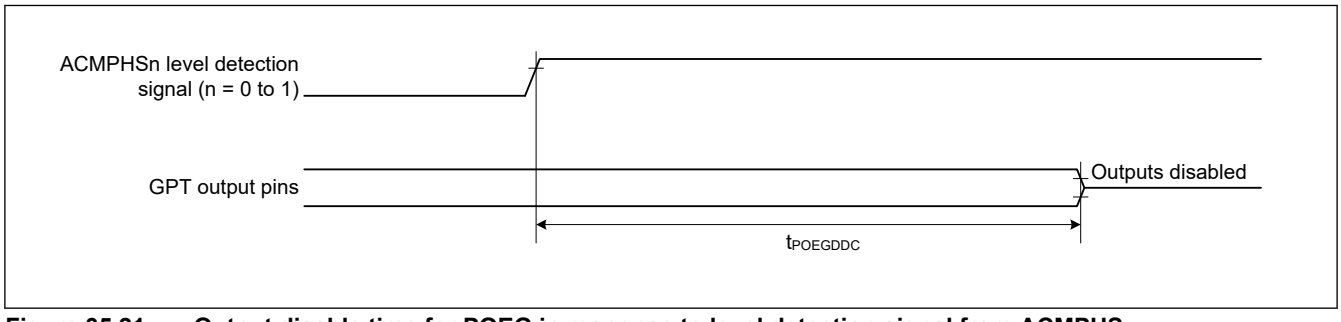


Figure 35.21 Output disable time for POEG in response to level detection signal from ACMPHS

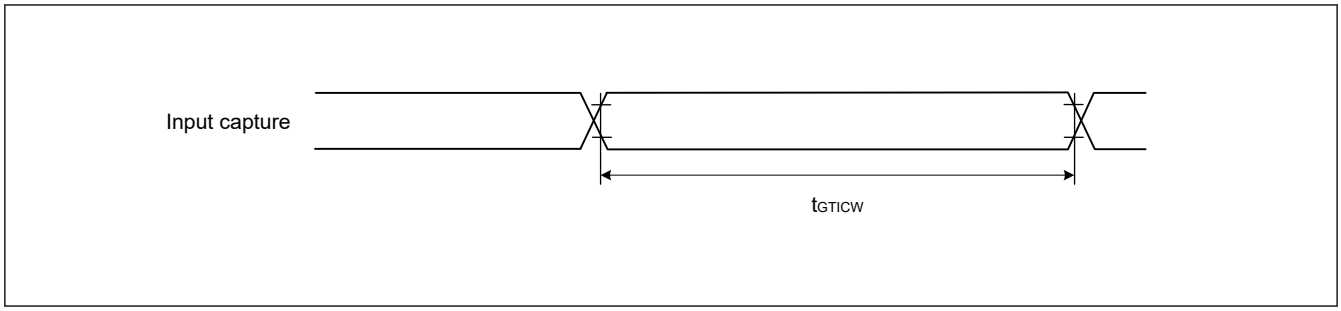


Figure 35.22 GPT input capture timing

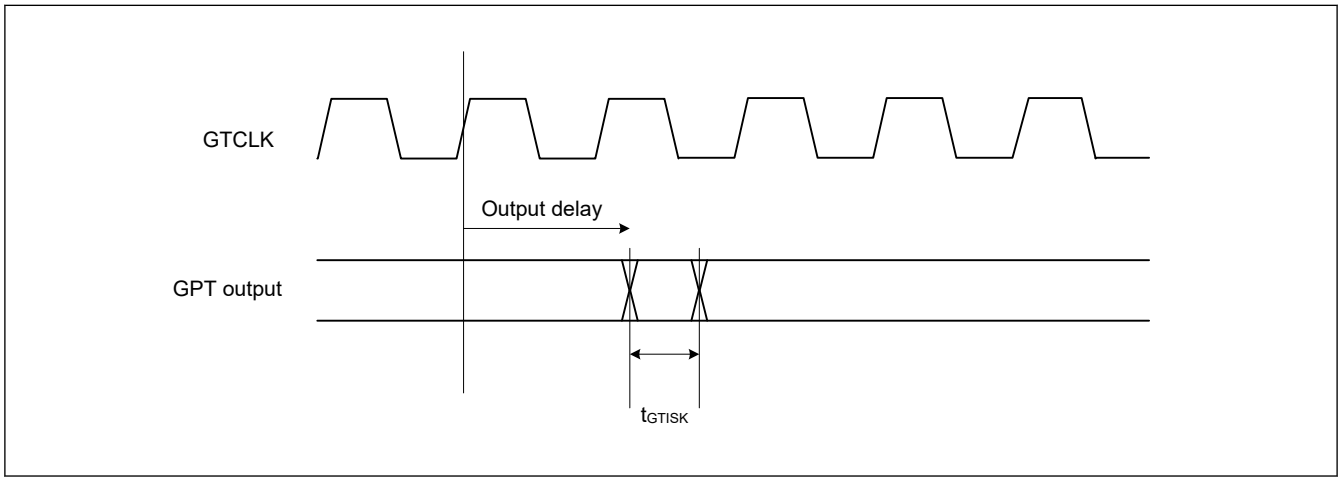


Figure 35.23 GPT output delay skew

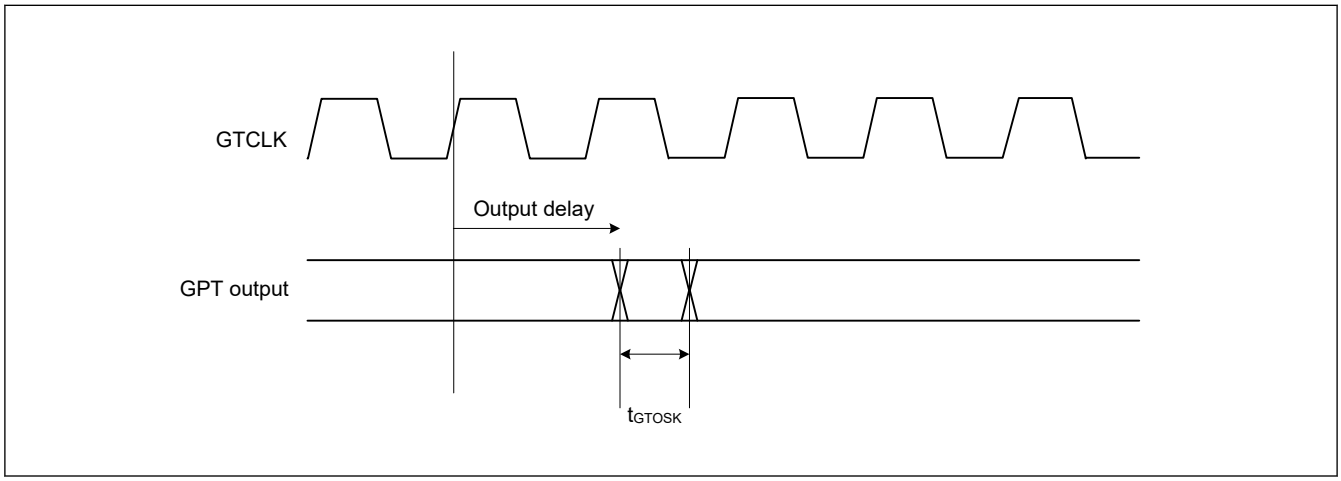


Figure 35.24 GPT output delay skew for OPS

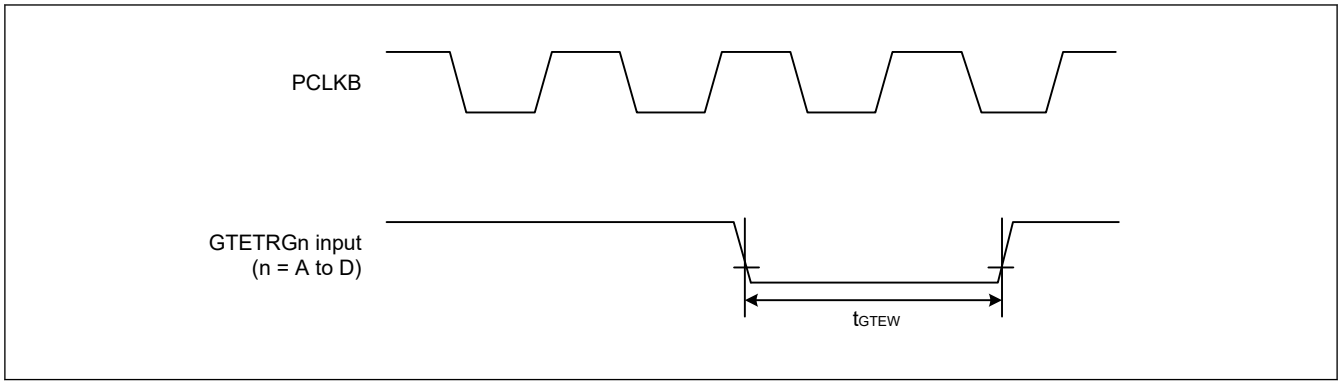


Figure 35.25 GPT external trigger input timing

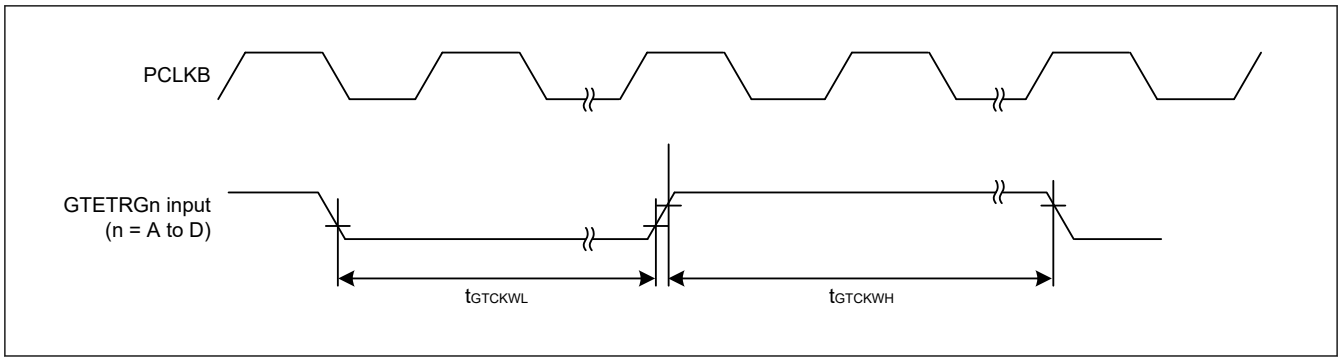


Figure 35.26 GPT clock input timing

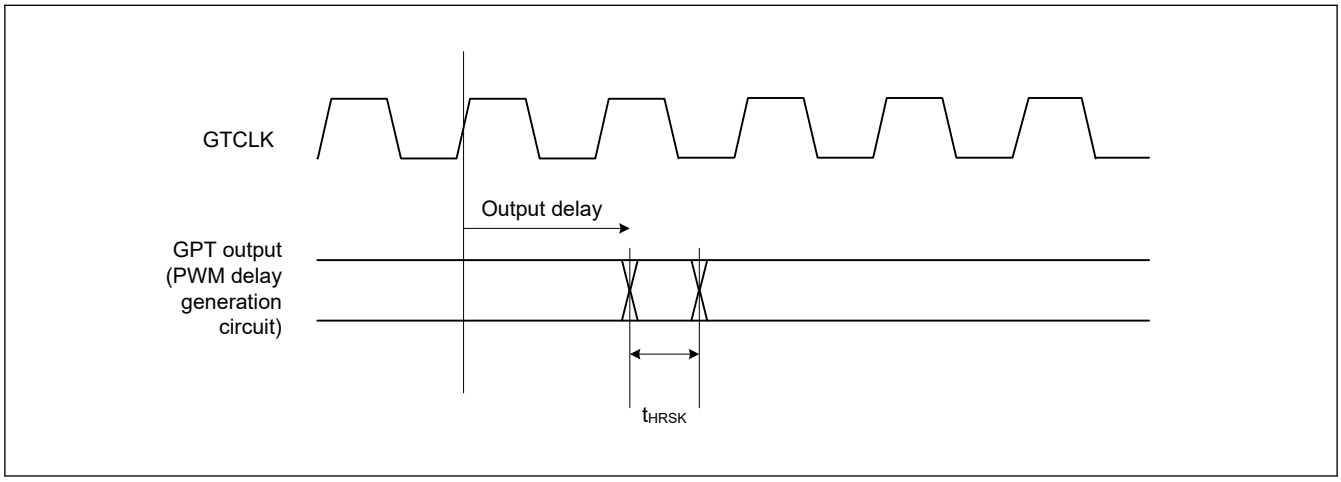


Figure 35.27 GPT (PDG) output delay skew

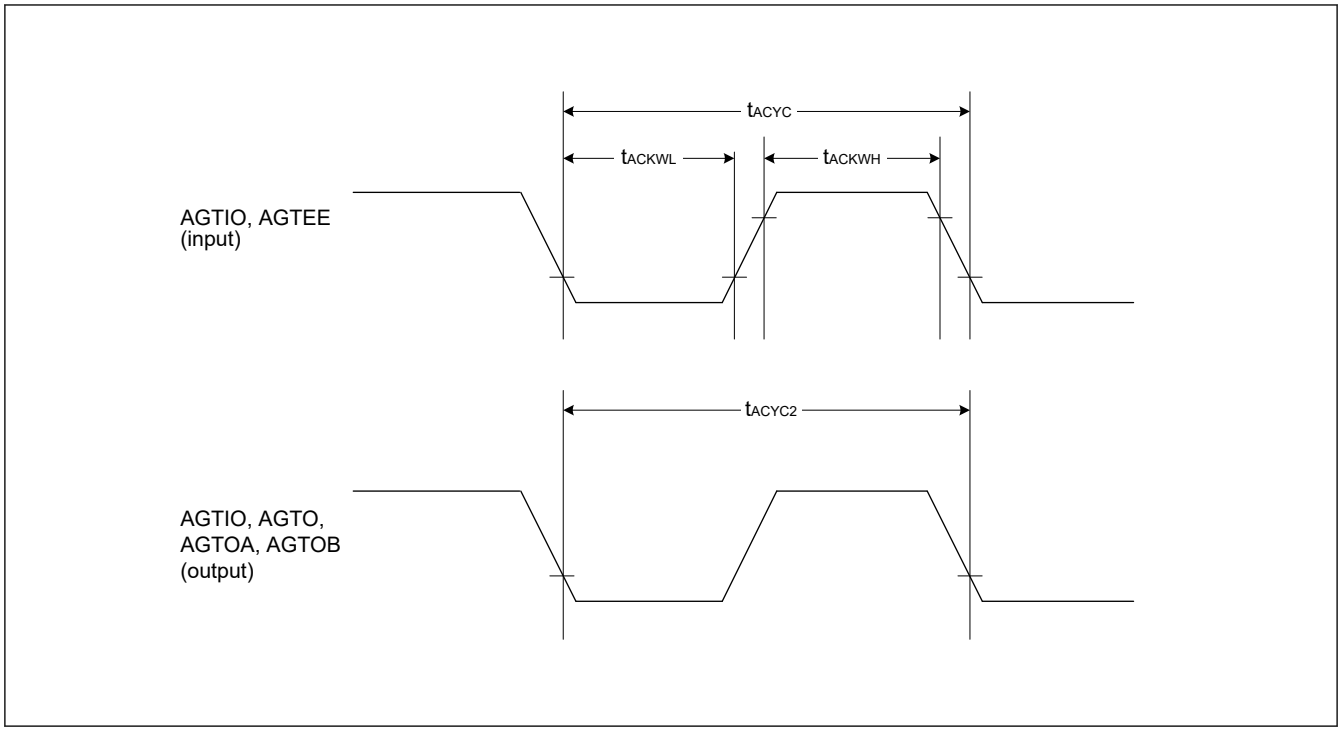


Figure 35.28 AGTW I/O timing

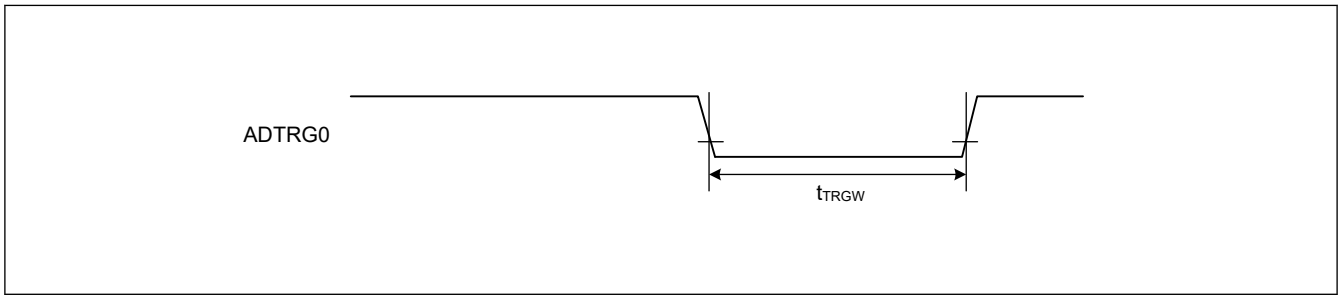


Figure 35.29 ADC12 trigger input timing

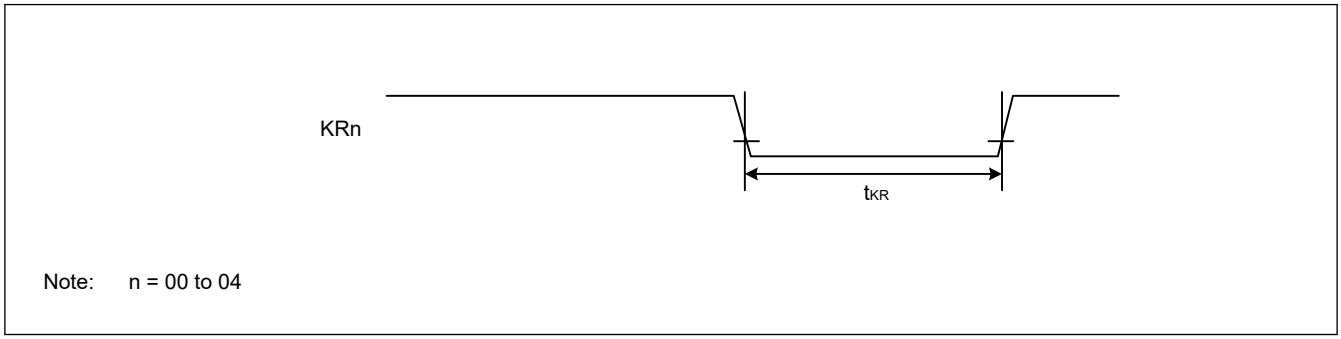


Figure 35.30 Key interrupt input timing

35.3.7 CAC Timing

Table 35.28 CAC timing

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
CAC	CACREF input pulse width	t_{Pcyc}*1 ≤ t_{CAC}*2	t_{CACREF}	4.5 × t_{CAC} + 3 × t_{Pcyc}	—	—	ns	—
		t_{Pcyc}*1 > t_{CAC}*2		5 × t_{CAC} + 6.5 × t_{Pcyc}	—	—	ns	

Note 1. t_{Pcyc}: PCLKB cycle.

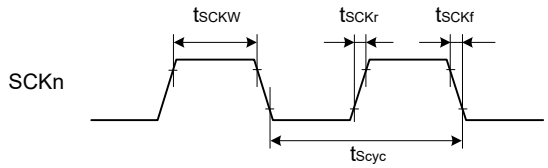
Note 2. t_{CAC}: CAC count clock source cycle.

35.3.8 SCI Timing

Table 35.29 SCI timing (1)

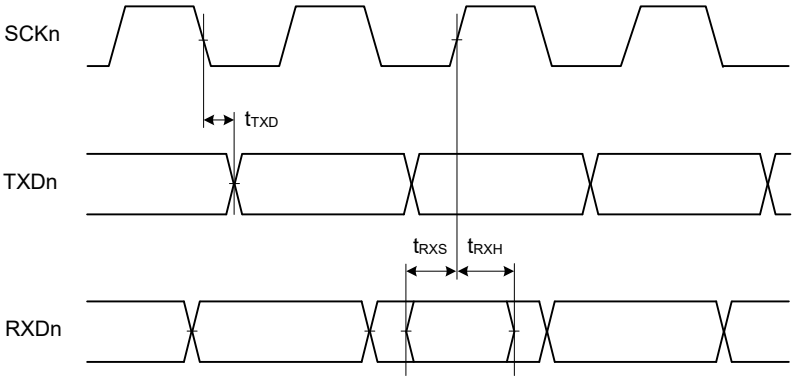
Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter				Symbol	Min	Max	Unit	Test conditions
SCI	Input clock cycle	Asynchronous	2.7 V ≤ VCC ≤ 5.5 V	t_{Scyc}	125	—	ns	Figure 35.31
			2.4 V ≤ VCC < 2.7 V		250	—		
			1.8 V ≤ VCC < 2.4 V		500	—		
			1.6 V ≤ VCC < 1.8 V		1000	—		
		Clock synchronous	2.7 V ≤ VCC ≤ 5.5 V		187.5	—		
			2.4 V ≤ VCC < 2.7 V		375	—		
			1.8 V ≤ VCC < 2.4 V		750	—		
			1.6 V ≤ VCC < 1.8 V		1500	—		
	Input clock pulse width			t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time			t_{SCKr}	—	20	ns	
	Input clock fall time			t_{SCKf}	—	20	ns	
	Output clock cycle	Asynchronous	2.7 V ≤ VCC ≤ 5.5 V	t_{Scyc}	187.5	—	ns	
			2.4 V ≤ VCC < 2.7 V		375	—		
			1.8 V ≤ VCC < 2.4 V		750	—		
			1.6 V ≤ VCC < 1.8 V		1500	—		
		Clock synchronous	2.7 V ≤ VCC ≤ 5.5 V		125	—		
			2.4 V ≤ VCC < 2.7 V		250	—		
			1.8 V ≤ VCC < 2.4 V		500	—		
			1.6 V ≤ VCC < 1.8 V		1000	—		
	Output clock pulse width			t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time		1.8 V ≤ VCC ≤ 5.5 V	t_{SCKr}	—	20	ns	
			1.6 V ≤ VCC < 1.8 V		—	30		
	Output clock fall time		1.8 V ≤ VCC ≤ 5.5 V	t_{SCKf}	—	20	ns	
			1.6 V ≤ VCC < 1.8 V		—	30		
	Transmit data delay time (master)	Clock synchronous	1.8 V ≤ VCC ≤ 5.5 V	t_{TXD}	—	40	ns	Figure 35.32
			1.6 V ≤ VCC < 1.8 V		—	45		
	Transmit data delay time (slave)	Clock synchronous	2.7 V ≤ VCC ≤ 5.5 V		—	55	ns	
			2.4 V ≤ VCC < 2.7 V		—	60		
			1.8 V ≤ VCC < 2.4 V		—	100		
			1.6 V ≤ VCC < 1.8 V		—	125		
	Receive data setup time (master)	Clock synchronous	2.7 V ≤ VCC ≤ 5.5 V	t_{RXS}	45	—	ns	
			2.4 V ≤ VCC < 2.7 V		55	—		
			1.8 V ≤ VCC < 2.4 V		90	—		
			1.6 V ≤ VCC < 1.8 V		110	—		
	Receive data setup time (slave)	Clock synchronous	2.7 V ≤ VCC ≤ 5.5 V		40	—	ns	
			1.6 V ≤ VCC < 2.7 V		45	—		
	Receive data hold time (master)	Clock synchronous		t_{RXH}	5	—	ns	
	Receive data hold time (slave)	Clock synchronous		t_{RXH}	40	—	ns	



Note: n = 0 to 2, 9

Figure 35.31 SCK clock input timing



Note: n = 0 to 2, 9

Figure 35.32 SCI input/output timing in clock synchronous mode

Table 35.30 SCI timing (2) (1 of 2)

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

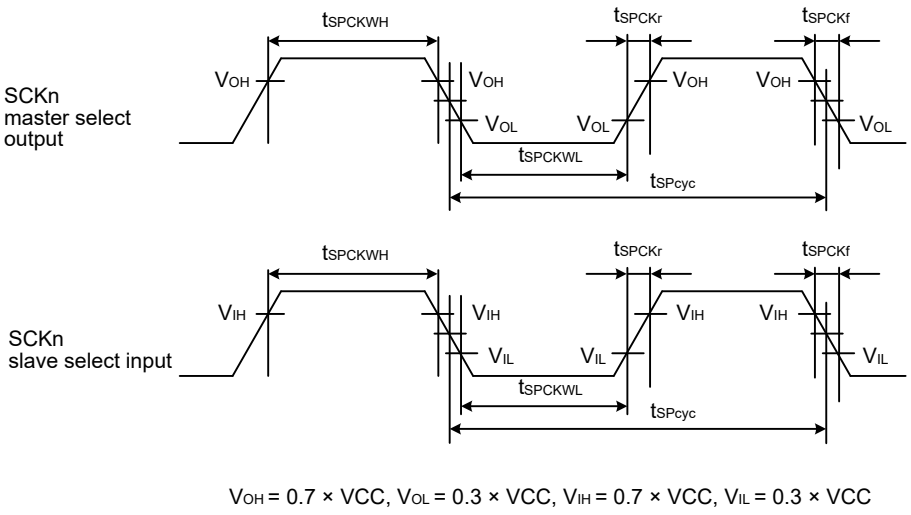
Parameter				Symbol	Min	Max	Unit*1	Test conditions
Simple SPI	SCK clock cycle output (master)		2.7 V ≤ VCC ≤ 5.5 V	t_{SPcyc}	125	—	ns	Figure 35.33
			2.4 V ≤ VCC < 2.7 V		250	—		
			1.8 V ≤ VCC < 2.4 V		500	—		
			1.6 V ≤ VCC < 1.8 V		1000	—		
	SCK clock cycle input (slave)		2.7 V ≤ VCC ≤ 5.5 V		187.5	—		
			2.4 V ≤ VCC < 2.7 V		375	—		
			1.8 V ≤ VCC < 2.4 V		750	—		
			1.6 V ≤ VCC < 1.8 V		1500	—		
	SCK clock high pulse width			t_{SPCKWH}	0.4	0.6	t_{SPcyc}	
	SCK clock low pulse width			t_{SPCKWL}	0.4	0.6	t_{SPcyc}	
	SCK clock rise and fall time		1.8 V ≤ VCC ≤ 5.5 V	t_{SPCKr}, t_{SPCKf}	—	20	ns	
			1.6 V ≤ VCC < 1.8 V		—	30		
	Data input setup time	Master	2.7 V ≤ VCC ≤ 5.5 V	t_{SU}	45	—	ns	Figure 35.34 to Figure 35.37
			2.4 V ≤ VCC < 2.7 V		55	—		
			1.8 V ≤ VCC < 2.4 V		80	—		
			1.6 V ≤ VCC < 1.8 V		110	—		
		Slave	2.7 V ≤ VCC ≤ 5.5 V		40	—		
			1.6 V ≤ VCC < 2.7 V		45	—		
	Data input hold time	Master		t_H	33.3	—	ns	
		Slave			40	—		
	SS input setup time			t_{LEAD}	1	—	t_{SPcyc}	
	SS input hold time			t_{LAG}	1	—	t_{SPcyc}	
	Data output delay time	Master	1.8 V ≤ VCC ≤ 5.5 V	t_{OD}	—	40	ns	
			1.6 V ≤ VCC < 1.8 V		—	50		
		Slave	2.4 V ≤ VCC ≤ 5.5 V		—	65		
			1.8 V ≤ VCC < 2.4 V		—	100		
			1.6 V ≤ VCC < 1.8 V		—	125		
	Data output hold time	Master	2.7 V ≤ VCC ≤ 5.5 V	t_{OH}	-10	—	ns	
			2.4 V ≤ VCC < 2.7 V		-20	—		
			1.8 V ≤ VCC < 2.4 V		-30	—		
			1.6 V ≤ VCC < 1.8 V		-40	—		
		Slave			-10	—		
	Data rise and fall time	Master	1.8 V ≤ VCC ≤ 5.5 V	t_{Dr}, t_{Df}	—	20	ns	
			1.6 V ≤ VCC < 1.8 V		—	30		
		Slave	1.8 V ≤ VCC ≤ 5.5 V		—	20		
			1.6 V ≤ VCC < 1.8 V		—	30		

Table 35.30 SCI timing (2) (2 of 2)

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter				Symbol	Min	Max	Unit*1	Test conditions
Simple SPI	Slave access time	2.4 V ≤ VCC ≤ 5.5 V		t_{SA}	—	6	t_{Pcyc}	Figure 35.37
		1.8 V ≤ VCC < 2.4 V	24 MHz ≤ PCLKB ≤ 32 MHz		—	7		
			PCLKB < 24 MHz		—	6		
		1.6 V ≤ VCC < 1.8 V			—	6		
	Slave output release time	2.4 V ≤ VCC ≤ 5.5 V		t_{REL}	—	6	t_{Pcyc}	
		1.8 V ≤ VCC < 2.4 V	24 MHz ≤ PCLKB ≤ 32 MHz		—	7		
			PCLKB < 24 MHz		—	6		
		1.6 V ≤ VCC < 1.8 V			—	6		

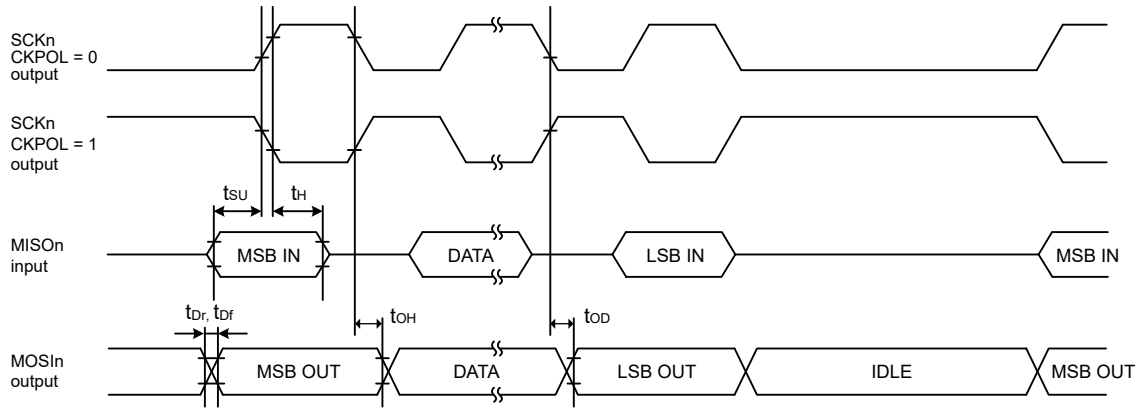
Note 1. t_{Pcyc}: PCLKB cycle.



V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC

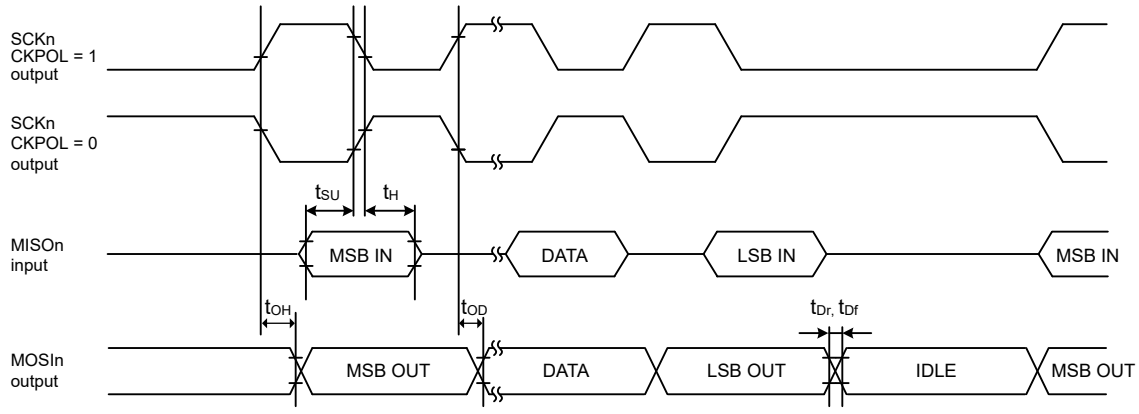
Note: n = 0 to 2, 9

Figure 35.33 SCI simple SPI mode clock timing



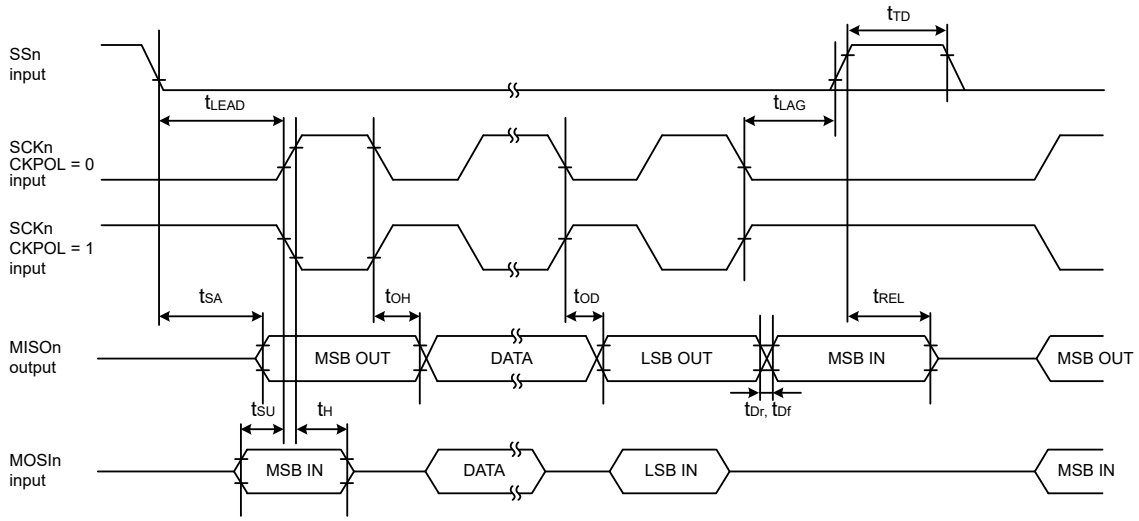
Note: n = 0 to 2, 9

Figure 35.34 SCI simple SPI mode timing (master, CKPH = 1)



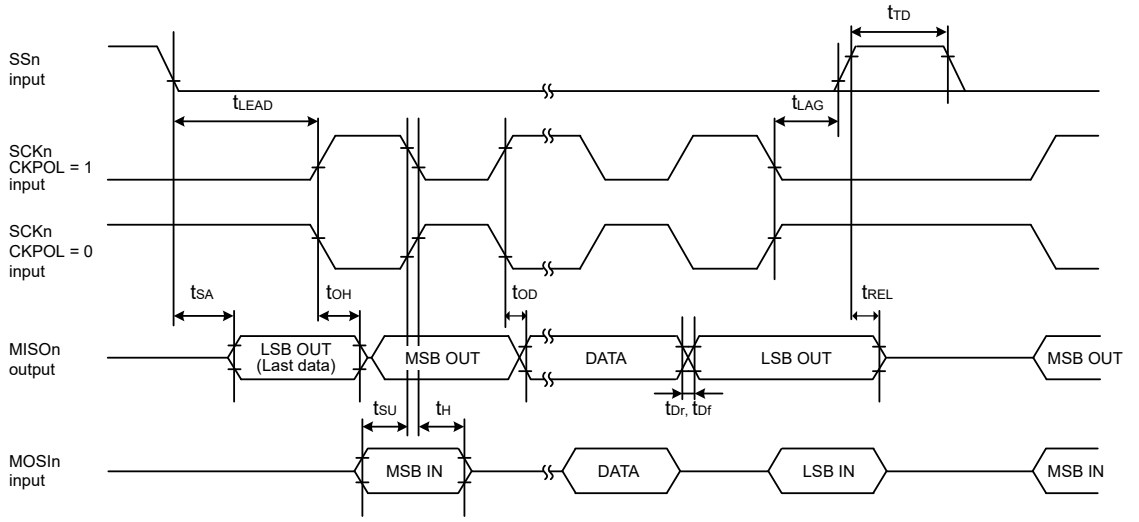
Note: n = 0 to 2, 9

Figure 35.35 SCI simple SPI mode timing (master, CKPH = 0)



Note: n = 0 to 2, 9

Figure 35.36 SCI simple SPI mode timing (slave, CKPH = 1)



Note: n = 0 to 2, 9

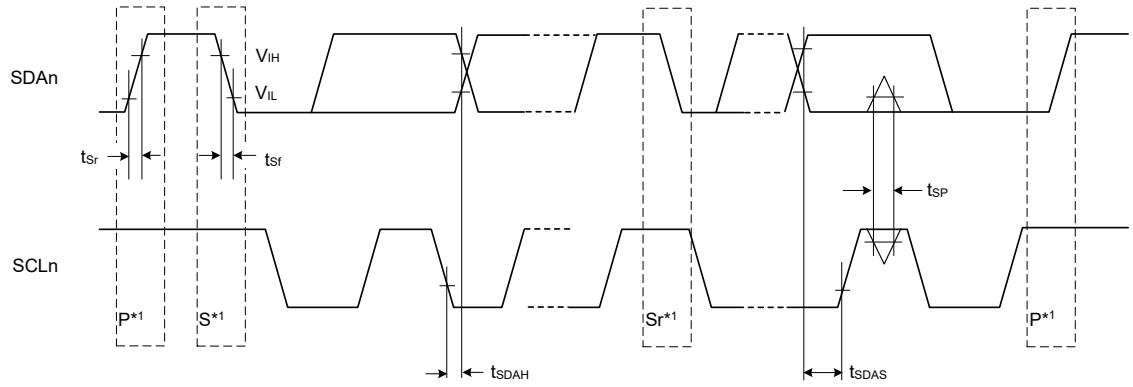
Figure 35.37 SCI simple SPI mode timing (slave, CKPH = 0)

Table 35.31 SCI timing (3)

Conditions: VCC = AVCC0 = 2.7 to 5.5 V

Parameter		Symbol	Min	Max	Unit	Test conditions
Simple IIC (Standard mode)	SDA input rise time	t_{Sr}	—	1000	ns	Figure 35.38
	SDA input fall time	t_{Sf}	—	300	ns	
	SDA input spike pulse removal time	t_{SP}	0	4 × t_{IICcyc} *1	ns	
	Data input setup time	t_{SDAS}	250	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b *2	—	400	pF	
Simple IIC (Fast mode)	SDA input rise time	t_{Sr}	—	300	ns	Figure 35.38
	SDA input fall time	t_{Sf}	—	300	ns	
	SDA input spike pulse removal time	t_{SP}	0	4 × t_{IICcyc} *1	ns	
	Data input setup time	t_{SDAS}	100	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b *2	—	400	pF	

Note 1. t_{IICcyc}: Clock cycle selected by the SMR.CKS[1:0] bits.
Note 2. C_b indicates the total capacity of the bus line.



Test conditions:
V_{IH} = VCC × 0.7, V_{IL} = VCC × 0.3
V_{OL} = 0.6 V, I_{OL} = 6 mA

Note: n = 0 to 2, 9
Note 1. S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 35.38 SCI simple IIC mode timing

35.3.9 SPI Timing

Table 35.32 SPI timing (1 of 3)

Parameter				Symbol	Min	Max	Unit*1	Test conditions
SPI	RSPCK clock cycle	Master	2.7 V ≤ VCC ≤ 5.5 V	t_{SPcyc}	62.5	—	ns	Figure 35.39 C = 30 pF
			2.4 V ≤ VCC < 2.7 V		125	—		
			1.8 V ≤ VCC < 2.4 V		250	—		
			1.6 V ≤ VCC < 1.8 V		500	—		
		Slave	2.7 V ≤ VCC ≤ 5.5 V		187.5	—		
			2.4 V ≤ VCC < 2.7 V		375	—		
			1.8 V ≤ VCC < 2.4 V		750	—		
			1.6 V ≤ VCC < 1.8 V		1500	—		
	RSPCK clock high pulse width	Master		t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	—	ns	
		Slave			$3 \times t_{Pcyc}$	—		
	RSPCK clock low pulse width	Master		t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	—	ns	
		Slave			$3 \times t_{Pcyc}$	—		
	RSPCK clock rise and fall time	Output	2.7 V ≤ VCC ≤ 5.5 V	t_{SPCKr}, t_{SPCKf}	—	10	ns	
			2.4 V ≤ VCC < 2.7 V		—	15		
			1.8 V ≤ VCC ≤ 2.4 V		—	20		
			1.6 V ≤ VCC < 1.8 V		—	30		
		Input			—	0.1	µs/V	

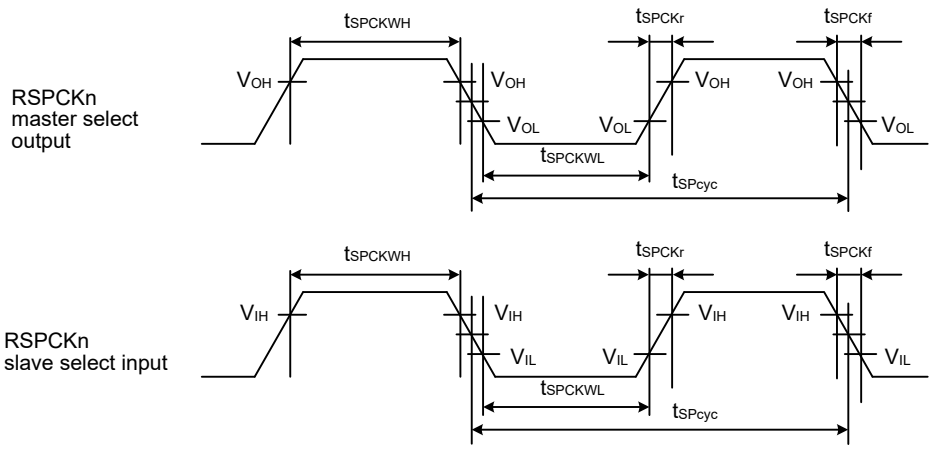
Table 35.32 SPI timing (2 of 3)

Parameter					Symbol	Min	Max	Unit*1	Test conditions
SPI	Data input setup time	Master	2.7 V ≤ VCC ≤ 5.5 V		t_{SU}	10	—	ns	Figure 35.40 to Figure 35.45 C = 30 pF
			2.4 V ≤ VCC < 2.7 V	16 MHz < PCLKB ≤ 32 MHz		30	—		
				PCLKB ≤ 16 MHz		10	—		
			1.8 V ≤ VCC < 2.4 V	16 MHz < PCLKB ≤ 32 MHz		55	—		
				8 MHz < PCLKB ≤ 16 MHz		30	—		
				PCLKB ≤ 8 MHz		10	—		
			1.6 V ≤ VCC < 1.8 V			10	—		
		Slave	2.4 V ≤ VCC ≤ 5.5 V			10	—		
			1.8 V ≤ VCC < 2.4 V			15	—		
			1.6 V ≤ VCC < 1.8 V			20	—		
	Data input hold time	Master (RSPCK is PCLKB/2)			t_{HF}	0	—	ns	
		Master (RSPCK is not PCLKB/2)			t_{H}	t_{Pcyc}	—		
		Slave			t_{H}	20	—		
SPI	SSL setup time	Master	1.8 V ≤ VCC ≤ 5.5 V		t_{LEAD}	-30 + N × t_{SPcyc}*2	—	ns	
			1.6 V ≤ VCC < 1.8 V			-50 + N × t_{SPcyc}*2	—		
		Slave				6 × t_{Pcyc}	—	ns	
	SSL hold time	Master			t_{LAG}	-30 + N × t_{SPcyc}*3	—	ns	
		Slave				6 × t_{Pcyc}	—	ns	
	Data output delay time	Master	2.7 V ≤ VCC ≤ 5.5 V		t_{OD}	—	14	ns	
			2.4 V ≤ VCC < 2.7 V			—	20		
			1.8 V ≤ VCC < 2.4 V			—	25		
			1.6 V ≤ VCC < 1.8 V			—	30		
		Slave	2.7 V ≤ VCC ≤ 5.5 V			—	50		
			2.4 V ≤ VCC < 2.7 V			—	60		
			1.8 V ≤ VCC < 2.4 V			—	85		
			1.6 V ≤ VCC < 1.8 V			—	110		
	Data output hold time	Master			t_{OH}	0	—	ns	
		Slave				0	—		
	Successive transmission delay time	Master			t_{TD}	t_{SPcyc} + 2 × t_{Pcyc}	8 × t_{SPcyc} + 2 × t_{Pcyc}	ns	
		Slave				6 × t_{Pcyc}	—		

Table 35.32 SPI timing (3 of 3)

Parameter				Symbol	Min	Max	Unit*1	Test conditions
SPI	MOSI and MISO rise and fall time	Output	2.7 V ≤ VCC ≤ 5.5 V	t_{Dr}, t_{Df}	—	10	ns	Figure 35.40 to Figure 35.45 C = 30 pF
			2.4 V ≤ VCC < 2.7 V		—	15		
			1.8 V ≤ VCC < 2.4 V		—	20		
			1.6 V ≤ VCC < 1.8 V		—	30		
		Input			—	1	μs	
	SSL rise and fall time	Output	2.7 V ≤ VCC ≤ 5.5 V	t_{SSLr}, t_{SSLf}	—	10	ns	
			2.4 V ≤ VCC < 2.7 V		—	15		
			1.8 V ≤ VCC < 2.4 V		—	20		
			1.6 V ≤ VCC < 1.8 V		—	30		
		Input			—	1	μs	
	Slave access time		2.4 V ≤ VCC ≤ 5.5 V	t_{SA}	—	2 × t_{Pcyc} + 100	ns	Figure 35.44 and Figure 35.45 C = 30 pF
			1.8 V ≤ VCC < 2.4 V		—	2 × t_{Pcyc} + 140		
			1.6 V ≤ VCC < 1.8 V		—	2 × t_{Pcyc} + 180		
	Slave output release time		2.4 V ≤ VCC ≤ 5.5 V	t_{REL}	—	2 × t_{Pcyc} + 100	ns	
			1.8 V ≤ VCC < 2.4 V		—	2 × t_{Pcyc} + 140		
			1.6 V ≤ VCC < 1.8 V		—	2 × t_{Pcyc} + 180		

Note 1. t_{Pcyc}: PCLKB cycle.
Note 2. N is set as an integer from 1 to 8 by the SPCKD register.
Note 3. N is set as an integer from 1 to 8 by the SSLND register.



V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC

Note: n = A

Figure 35.39 SPI clock timing

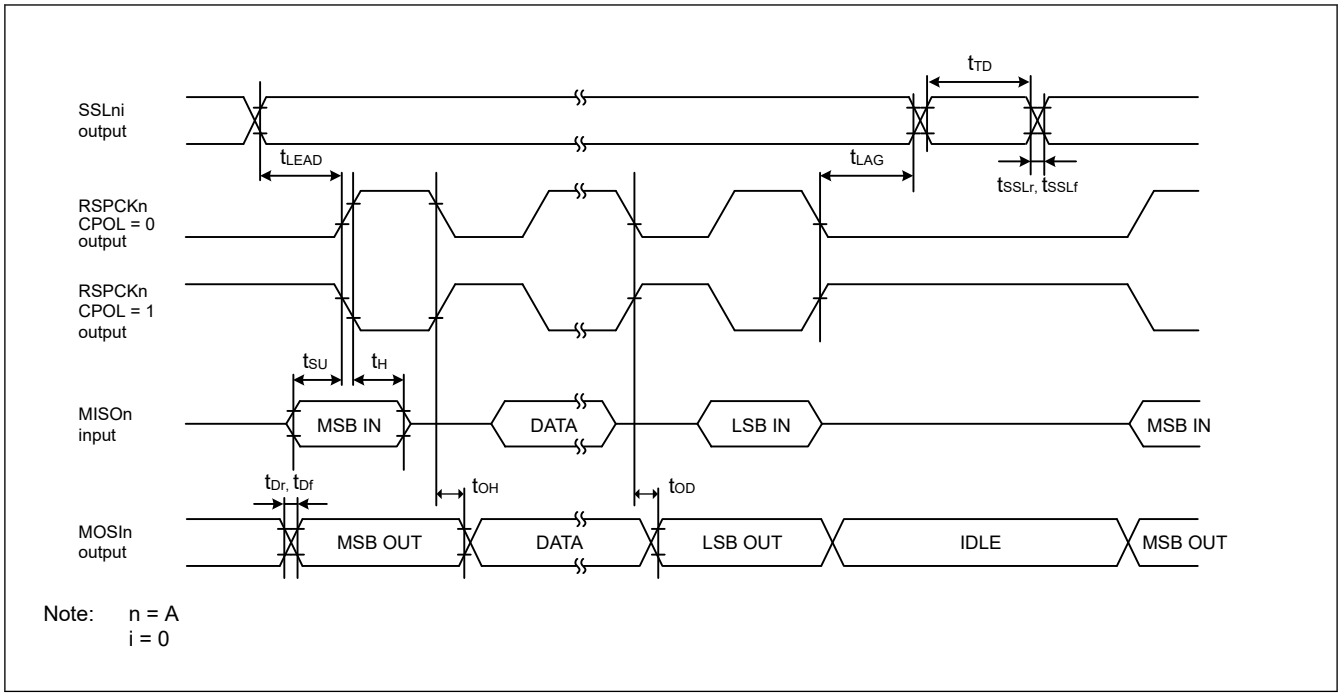


Figure 35.40 SPI timing (master, CPHA = 0) (bit rate: PCLKB division ratio is set to any value other than 1/2)

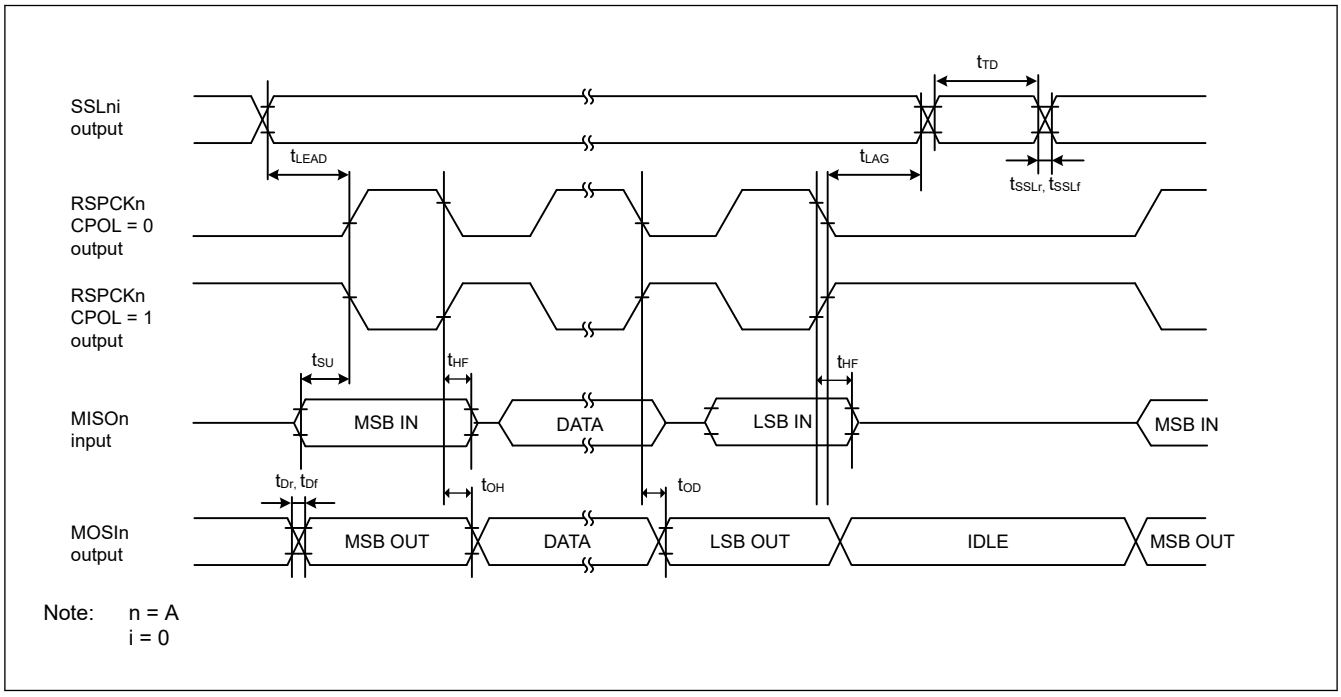
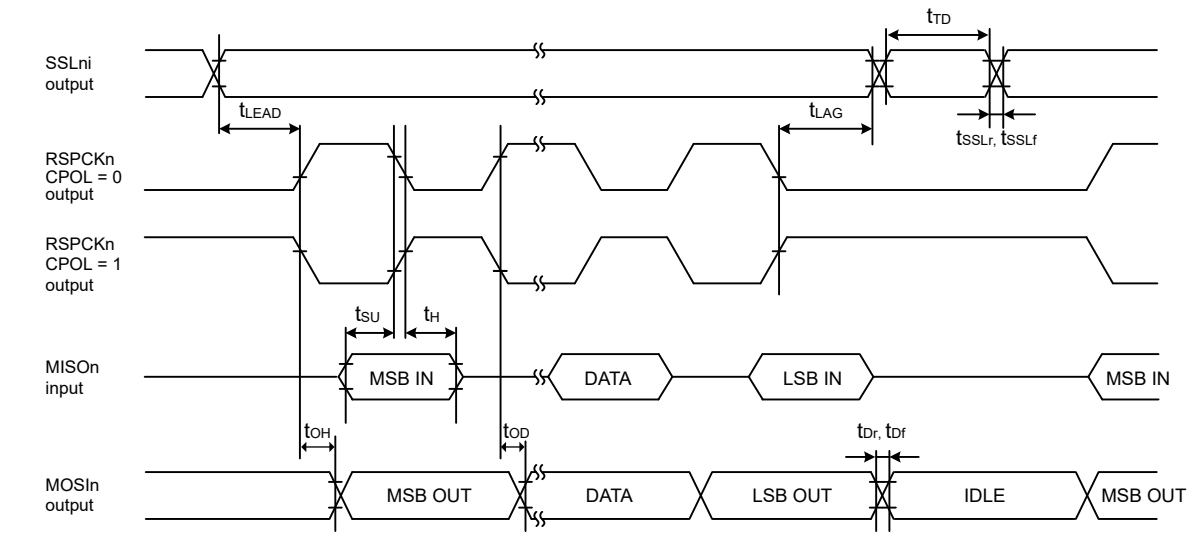
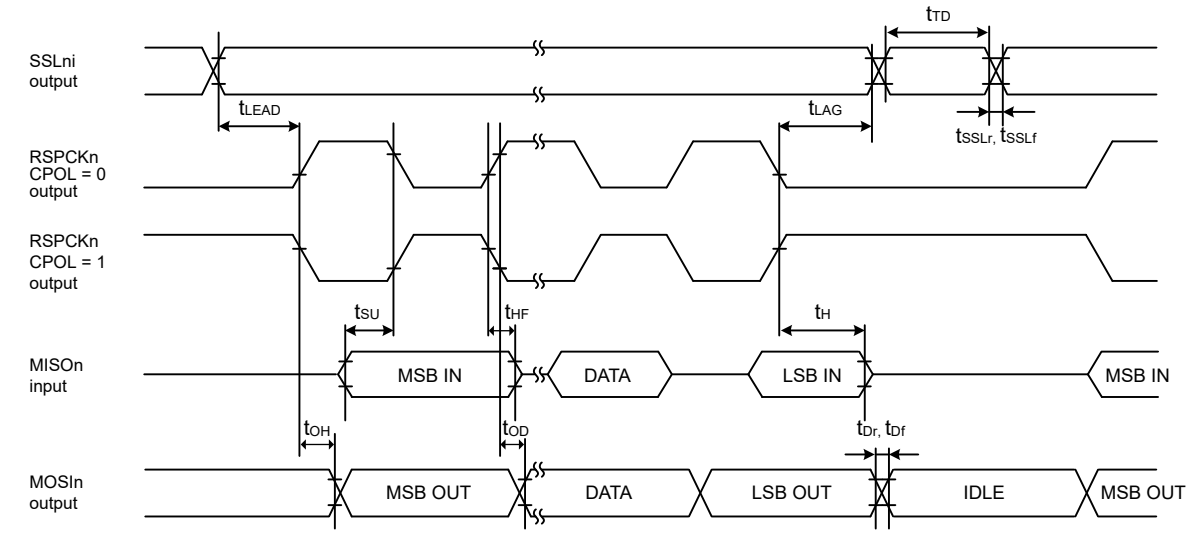


Figure 35.41 SPI timing (master, CPHA = 0) (bit rate: PCLKB division ratio is set to 1/2)



Note: n = A
i = 0

Figure 35.42 SPI timing (master, CPHA = 1) (bit rate: PCLKB division ratio is set to any value other than 1/2)



Note: n = A
i = 0

Figure 35.43 SPI timing (master, CPHA = 1) (bit rate: PCLKB division ratio is set to 1/2)

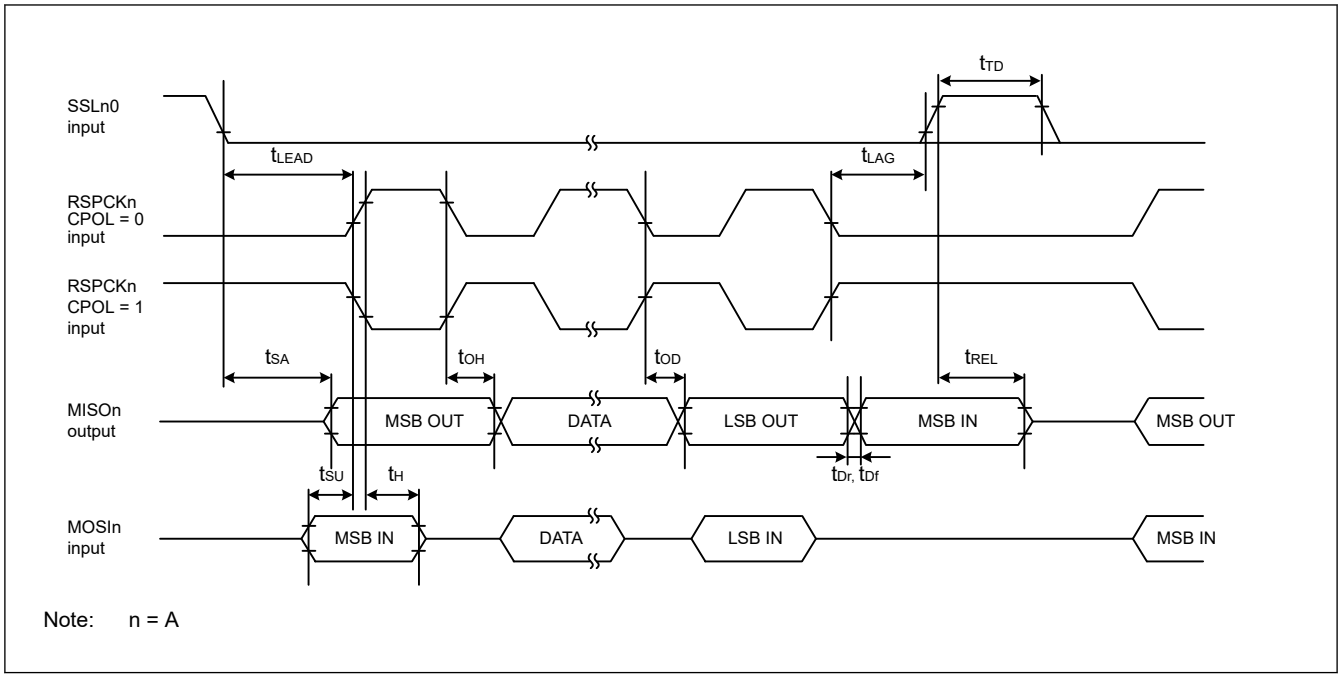


Figure 35.44 SPI timing (slave, CPHA = 0)

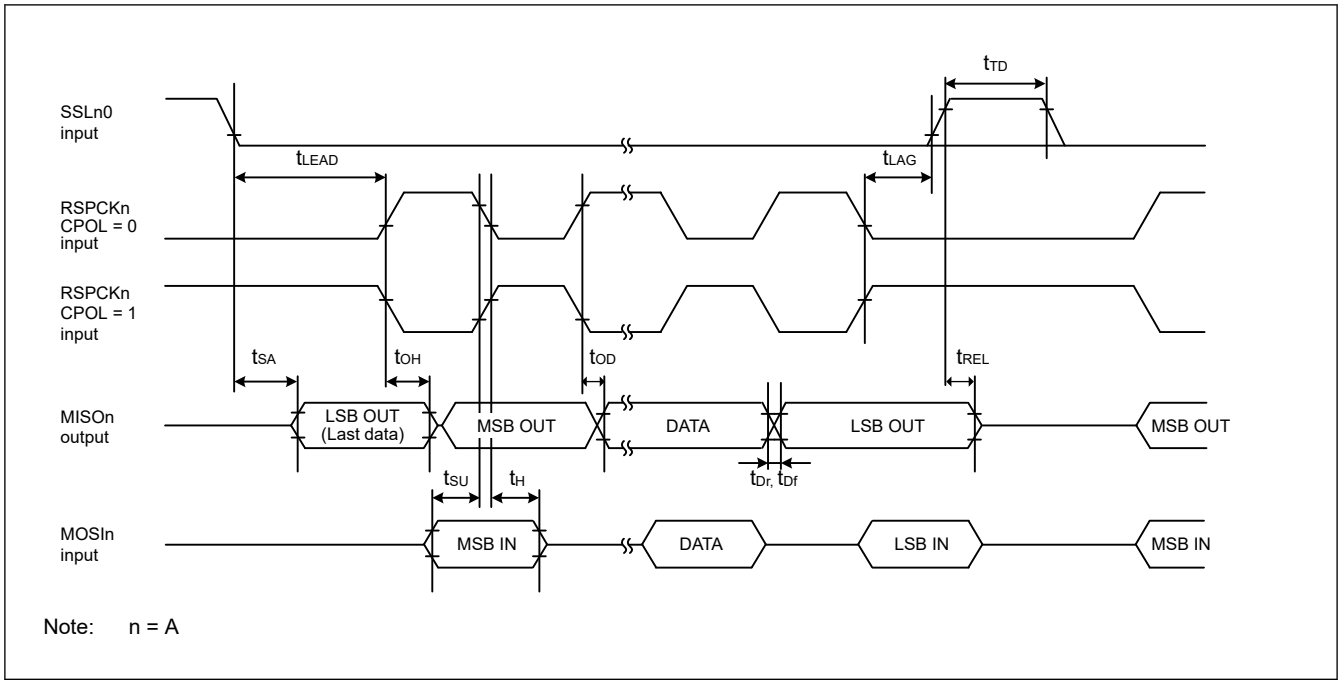


Figure 35.45 SPI timing (slave, CPHA = 1)

35.3.10 IIC Timing

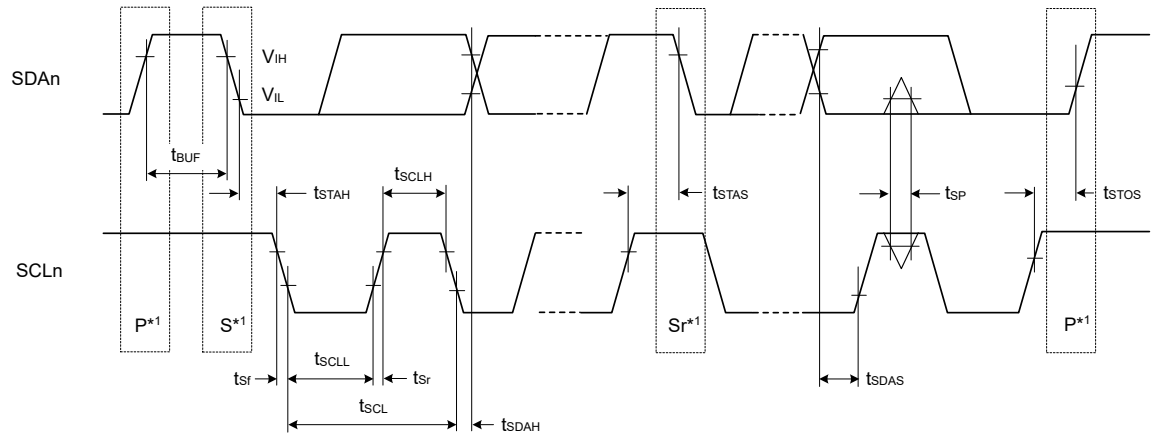
Table 35.33 IIC timing

Conditions: VCC = AVCC0 = 2.7 to 5.5 V

Parameter		Symbol	Min*1	Max	Unit	Test conditions
IIC (standard mode, SMBus)	SCL input cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 1300	—	ns	Figure 35.46
	SCL input high pulse width	t_{SCLH}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL input low pulse width	t_{SCLL}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA input rise time	t_{Sr}	—	1000	ns	
	SCL, SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA input bus free time (when wakeup function is disabled)	t_{BUF}	3 (6) × t_{IICcyc} + 300	—	ns	
	SDA input bus free time (when wakeup function is enabled)	t_{BUF}	3 (6) × t_{IICcyc} + 4 × t_{Pcyc} + 300	—	ns	
	START condition input hold time (when wakeup function is disabled)	t_{STAH}	t_{IICcyc} + 300	—	ns	
	START condition input hold time (when wakeup function is enabled)	t_{STAH}	1 (5) × t_{IICcyc} + t_{Pcyc} + 300	—	ns	
	Repeated START condition input setup time	t_{STAS}	1000	—	ns	
	STOP condition input setup time	t_{STOS}	1000	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	
IIC (Fast mode)	SCL input cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 600	—	ns	Figure 35.46
	SCL input high pulse width	t_{SCLH}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL input low pulse width	t_{SCLL}	3 (6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA input rise time	t_{Sr}	—	300	ns	
	SCL, SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}	ns	
	SDA input bus free time (When wakeup function is disabled)	t_{BUF}	3 (6) × t_{IICcyc} + 300	—	ns	
	SDA input bus free time (When wakeup function is enabled)	t_{BUF}	3 (6) × t_{IICcyc} + 4 × t_{Pcyc}+ 300	—	ns	
	START condition input hold time (When wakeup function is disabled)	t_{STAH}	t_{IICcyc} + 300	—	ns	
	START condition input hold time (When wakeup function is enabled)	t_{STAH}	1 (5) × t_{IICcyc} + t_{Pcyc} + 300	—	ns	
	Repeated START condition input setup time	t_{STAS}	300	—	ns	
	STOP condition input setup time	t_{STOS}	300	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	

Note: t_{IICcyc}: IIC internal reference clock (IICφ) cycle, t_{Pcyc}: PCLKB cycle

Note 1. Values in parentheses apply when ICMR3.NF[1:0] is set to 11b while the digital filter is enabled with ICFER.NFE set to 1.



Note: n = 0
Note 1. S, P, and Sr indicate the following conditions:
S: Start condition
P: Stop condition
Sr: Restart condition

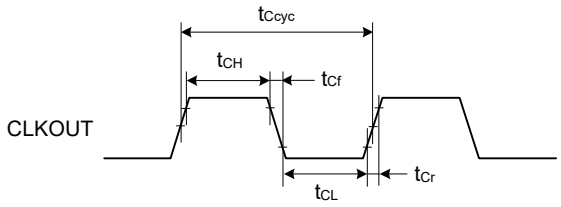
Figure 35.46 I²C bus interface input/output timing

35.3.11 CLKOUT Timing

Table 35.34 CLKOUT timing

Parameter			Symbol	Min	Max	Unit	Test conditions
CLKOUT	CLKOUT pin output cycle*1	2.7 V ≤ VCC ≤ 5.5 V	t_{Ccyc}	62.5	—	ns	Figure 35.47
		1.8 V ≤ VCC < 2.7 V		125	—		
		1.6 V ≤ VCC < 1.8 V		250	—		
	CLKOUT pin high pulse width*2	2.7 V ≤ VCC ≤ 5.5 V	t_{CH}	15	—	ns	
		1.8 V ≤ VCC < 2.7 V		30	—		
		1.6 V ≤ VCC < 1.8 V		150	—		
	CLKOUT pin low pulse width*2	2.7 V ≤ VCC ≤ 5.5 V	t_{CL}	15	—	ns	
		1.8 V ≤ VCC < 2.7 V		30	—		
		1.6 V ≤ VCC < 1.8 V		150	—		
	CLKOUT pin output rise time	2.7 V ≤ VCC ≤ 5.5 V	t_{Cr}	—	12	ns	
		1.8 V ≤ VCC < 2.7 V		—	25		
		1.6 V ≤ VCC < 1.8 V		—	50		
	CLKOUT pin output fall time	2.7 V ≤ VCC ≤ 5.5 V	t_{Cf}	—	12	ns	
		1.8 V ≤ VCC < 2.7 V		—	25		
		1.6 V ≤ VCC < 1.8 V		—	50		

Note 1. When the EXTAL external clock input or an oscillator is used with division by 1 (the CKOCR.CKOSEL[2:0] bits are 011b and the CKOCR.CKODIV[2:0] bits are 000b) to output from CLKOUT, specifications in Table 35.34 should be satisfied with 45% to 55% of input duty cycle.
Note 2. When MOCO is selected as the clock output source (the CKOCR.CKOSEL[2:0] bits are 001b), set the clock output division ratio to be divided by 2 (the CKOCR.CKODIV[2:0] bits are 001b).



Test conditions: V_{OH} = VCC × 0.7, V_{OL} = VCC × 0.3, I_{OH} = -1.0 mA, I_{OL} = 1.0 mA, C = 30 pF

Figure 35.47 CLKOUT output timing

35.4 ADC12 Characteristics

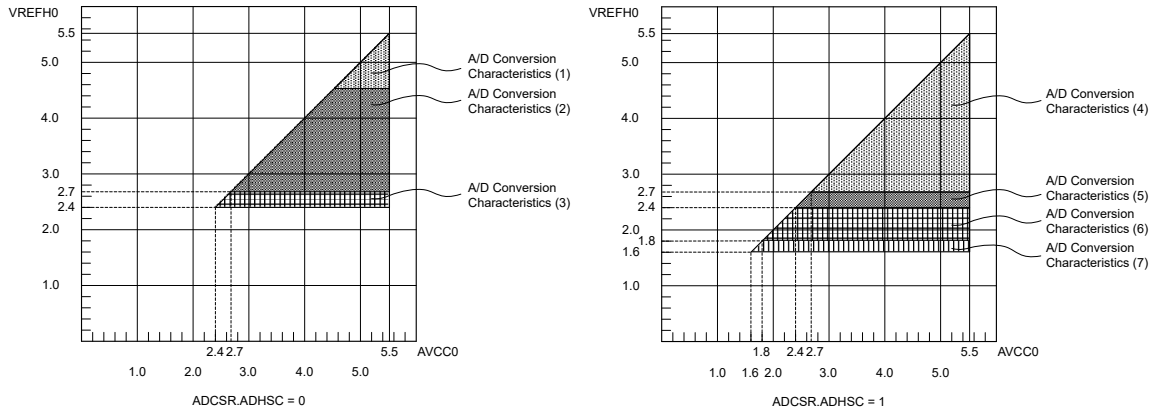


Figure 35.48 AVCC0 to VREFH0 voltage range

Table 35.35 A/D conversion characteristics (1) in high-speed A/D conversion mode (1 of 2)

Conditions: VCC = AVCC0 = VREFH0 = 4.5 to 5.5 V[*5], VSS = AVSS0 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
PCLKD (ADCLK) frequency		1	—	64	MHz	ADACSR.ADSAC = 0
				48	MHz	ADACSR.ADSAC = 1
Analog input capacitance[*2]	Cs	—	—	9[*3]	pF	High-precision channel
		—	—	10[*3]	pF	Normal-precision channel
Analog input resistance	Rs	—	—	1.3[*3]	kΩ	High-precision channel
		—	—	5.0[*3]	kΩ	Normal-precision channel
Analog input voltage range	Ain	0	—	VREFH0	V	—
Resolution		—	—	12	Bit	—

Table 35.35 A/D conversion characteristics (1) in high-speed A/D conversion mode (2 of 2)

Conditions: VCC = AVCC0 = VREFH0 = 4.5 to 5.5 V[*5], VSS = AVSS0 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
Conversion time[*1] (Operation at PCLKD = 64 MHz)	Permissible signal source impedance Max. = 0.3 kΩ	0.70 (0.211)[*4]	—	—	µs	High-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0x0D ADACSR.ADSAC = 0
		1.30 (0.406+0.211)[*4]	—	—	µs	High-precision channel dedicated sample-and-hold circuits in use (AN000 to AN002) ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0x0D ADACSR.ADSAC = 0 ADSHCR.SSTSH[7:0] = 0x1A
		1.34 (0.852)[*4]	—	—	µs	Normal-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0x36 ADACSR.ADSAC = 0
Conversion time[*1] (Operation at PCLKD = 48 MHz)	Permissible signal source impedance Max. = 0.3 kΩ	0.67 (0.219)[*4]	—	—	µs	High-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0x0A ADACSR.ADSAC = 1
		1.33 (0.417+0.219)[*4]	—	—	µs	High-precision channel dedicated sample-and-hold circuits in use (AN000 to AN002) ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0x0A ADACSR.ADSAC = 1 ADSHCR.SSTSH[7:0] = 0x14
		1.29 (0.844)[*4]	—	—	µs	Normal-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0x28 ADACSR.ADSAC = 1
Offset error		—	±1.0	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than specified
Full-scale error		—	±1.0	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than specified
Quantization error		—	±0.5	—	LSB	—
Absolute accuracy		—	±2.5	±5.0	LSB	High-precision channel
				±7	LSB	High-precision channel when the sample-and-hold circuit is in use (0.25 V ≤ VAIN ≤ AVCC0 – 0.25 V)
				±8.0	LSB	Other than specified
DNL differential nonlinearity error		—	±1.0	—	LSB	—
INL integral nonlinearity error		—	±1.5	±3.0	LSB	—
Holding characteristics of sample-and-hold circuits		—	—	10[*6]	µs	—

Note: The characteristics apply when no pin functions other than 12-bit A/D converter input are used. Absolute accuracy does not include quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.
Note 1. The conversion time is the sum of the sampling time and the comparison time. The number of sampling states is indicated for the test conditions.
Note 2. Except for I/O input capacitance (Cin), see section 35.2.4. I/O V_{OH}, V_{OL}, and Other Characteristics.
Note 3. Reference data.
Note 4. ( ) lists sampling time.
Note 5. When VREFH0 < AVCC0, the MAX. values are as follows.
Absolute accuracy/Offset error/Full-scale error:

For voltage difference between AVCC0 and VREFH0, it should be added ±0.75 LSB/V to the Max spec.
INL integral non-linearity error:
For voltage difference between AVCC0 and VREFH0, it should be added ±0.2 LSB/V to the Max spec.

Note 6. For 3 channel simultaneous sampling: When using a 3 channel S/H circuit, there is a restriction that one AD conversion should be 3.3 µs or less (3.3 µs × 3 < Max 10 µs).

Table 35.36 A/D conversion characteristics (2) in high-speed A/D conversion mode

Conditions: VCC = AVCC0 = VREFH0 = 2.7 to 5.5 V[*5], VSS = AVSS0 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
PCLKD (ADCLK) frequency		1	—	48	MHz	—
Analog input capacitance[*2]	Cs	—	—	9[*3]	pF	High-precision channel
		—	—	10[*3]	pF	Normal-precision channel
Analog input resistance	Rs	—	—	1.9[*3]	kΩ	High-precision channel
		—	—	6.0[*3]	kΩ	Normal-precision channel
Analog input voltage range	Ain	0	—	VREFH0	V	—
Resolution		—	—	12	Bit	—
Conversion time[*1] (Operation at PCLKD = 48 MHz)	Permissible signal source impedance Max. = 0.3 kΩ	0.67 (0.219)[*4]	—	—	µs	High-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0x0A ADACSR.ADSAC = 1
		1.33 (0.417+0.219)[*4]	—	—	µs	High-precision channel dedicated sample-and-hold circuits in use (AN000 to AN002) ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0x0A ADACSR.ADSAC = 1 ADSHCR.SSTSH[7:0] = 0x14
		1.29 (0.844)[*4]	—	—	µs	Normal-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0x28 ADACSR.ADSAC = 1
Offset error		—	±1.0	±5.5	LSB	High-precision channel
				±7.0	LSB	Other than specified
Full-scale error		—	±1.0	±5.5	LSB	High-precision channel
				±7.0	LSB	Other than specified
Quantization error		—	±0.5	—	LSB	—
Absolute accuracy		—	±2.5	±6.0	LSB	High-precision channel
				±8.5	LSB	High-precision channel when the sample-and-hold circuit is in use (0.25 V ≤ VAIN ≤ AVCC0 – 0.25 V)
				±9.0	LSB	Other than specified
DNL differential nonlinearity error		—	±1.0	—	LSB	—
INL integral nonlinearity error		—	±1.5	±3.0	LSB	—
Holding characteristics of sample-and-hold circuits		—	—	10[*6]	µs	—

Note: The characteristics apply when no pin functions other than 12-bit A/D converter input are used. Absolute accuracy does not include quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. The number of sampling states is indicated for the test conditions.

Note 2. Except for I/O input capacitance (Cin), see section 35.2.4. I/O V_{OH}, V_{OL}, and Other Characteristics.

Note 3. Reference data.

Note 4. ( ) lists sampling time.

Note 5. When VREFH0 < AVCC0, the MAX. values are as follows.

Absolute accuracy/Offset error/Full-scale error:
For voltage difference between AVCC0 and VREFH0, it should be added ±0.75 LSB/V to the Max spec.
INL integral non-linearity error:
For voltage difference between AVCC0 and VREFH0, it should be added ±0.2 LSB/V to the Max spec.

Note 6. For 3 channel simultaneous sampling: When using a 3 channel S/H circuit, there is a restriction that one AD conversion should be 3.3 µs or less (3.3 µs × 3 < Max 10 µs).

Table 35.37 A/D conversion characteristics (3) in high-speed A/D conversion mode

Conditions: VCC = AVCC0 = VREFH0 = 2.4 to 5.5 V[*5], VSS = AVSS0 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min		Max	Unit	Test conditions
PCLKD (ADCLK) frequency		1	—	32	MHz	—
Analog input capacitance[*2]	Cs	—	—	9[*3]	pF	High-precision channel
		—	—	10[*3]	pF	Normal-precision channel
Analog input resistance	Rs	—	—	2.2[*3]	kΩ	High-precision channel
		—	—	7.0[*3]	kΩ	Normal-precision channel
Analog input voltage range	Ain	0	—	VREFH0	V	—
Resolution		—	—	12	Bit	—
Conversion time[*1] (Operation at PCLKD = 32 MHz)	Permissible signal source impedance Max. = 1.3 kΩ	1.00 (0.328)[*4]	—	—	µs	High-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0x0A ADACSR.ADSAC = 1
		1.94 (1.266)[*4]	—	—	µs	Normal-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0x28 ADACSR.ADSAC = 1
Offset error		—	±1.0	±5.5	LSB	High-precision channel
				±7.0	LSB	Other than specified
Full-scale error		—	±1.0	±5.5	LSB	High-precision channel
				±7.0	LSB	Other than specified
Quantization error		—	±0.5	—	LSB	—
Absolute accuracy		—	±2.50	±6.0	LSB	High-precision channel
				±9.0	LSB	Other than specified
DNL differential nonlinearity error		—	±1.0	—	LSB	—
INL integral nonlinearity error		—	±1.5	±3.0	LSB	—

Note: The characteristics apply when no pin functions other than 12-bit A/D converter input are used. Absolute accuracy does not include quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. The number of sampling states is indicated for the test conditions.

Note 2. Except for I/O input capacitance (Cin), see section 35.2.4. I/O V_{OH}, V_{OL}, and Other Characteristics.

Note 3. Reference data.

Note 4. ( ) lists sampling time.

Note 5. When VREFH0 < AVCC0, the MAX. values are as follows.
Absolute accuracy/Offset error/Full-scale error:
For voltage difference between AVCC0 and VREFH0, it should be added ±0.75 LSB/V to the Max spec.
INL integral non-linearity error:
For voltage difference between AVCC0 and VREFH0, it should be added ±0.2 LSB/V to the Max spec.

Table 35.38 A/D conversion characteristics (4) in low-power A/D conversion mode (1 of 2)

Conditions: VCC = AVCC0 = VREFH0 = 2.7 to 5.5 V[*5], VSS = AVSS0 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter	Min	Typ	Max	Unit	Test conditions
PCLKD (ADCLK) frequency	1	—	24	MHz	—

Table 35.38 A/D conversion characteristics (4) in low-power A/D conversion mode (2 of 2)

Conditions: VCC = AVCC0 = VREFH0 = 2.7 to 5.5 V[*5], VSS = AVSS0 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
Analog input capacitance[*2]	Cs	—	—	9[*3]	pF	High-precision channel
		—	—	10[*3]	pF	Normal-precision channel
Analog input resistance	Rs	—	—	1.9[*3]	kΩ	High-precision channel
		—	—	6[*3]	kΩ	Normal-precision channel
Analog input voltage range	Ain	0	—	VREFH0	V	—
Resolution		—	—	12	Bit	—
Conversion time[*1] (Operation at PCLKD = 24 MHz)	Permissible signal source impedance Max. = 1.1 kΩ	1.58 (0.438)[*4]	—	—	µs	High-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 0x0A ADACSR.ADSAC = 1
		2.0 (0.854)[*4]	—	—	µs	Normal-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 0x14 ADACSR.ADSAC = 1
Offset error		—	±1.25	±6.0	LSB	High-precision channel
				±7.5	LSB	Other than specified
Full-scale error		—	±1.25	±6.0	LSB	High-precision channel
				±7.5	LSB	Other than specified
Quantization error		—	±0.5	—	LSB	—
Absolute accuracy		—	±3.25	±7.0	LSB	High-precision channel
				±10.0	LSB	Other than specified
DNL differential nonlinearity error		—	±1.5	—	LSB	—
INL integral nonlinearity error		—	±1.75	±4.0	LSB	—

Note: The characteristics apply when no pin functions other than 12-bit A/D converter input are used. Absolute accuracy does not include quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.
Note 1. The conversion time is the sum of the sampling time and the comparison time. The number of sampling states is indicated for the test conditions.
Note 2. Except for I/O input capacitance (Cin), see section 35.2.4. I/O V_{OH}, V_{OL}, and Other Characteristics.
Note 3. Reference data.
Note 4. ( ) lists sampling time.
Note 5. When VREFH0 < AVCC0, the MAX. values are as follows.
Absolute accuracy/Offset error/Full-scale error:
For voltage difference between AVCC0 and VREFH0, it should be added ±0.75 LSB/V to the Max spec.
INL integral non-linearity error:
For voltage difference between AVCC0 and VREFH0, it should be added ±0.2 LSB/V to the Max spec.

Table 35.39 A/D conversion characteristics (5) in low-power A/D conversion mode (1 of 2)

Conditions: VCC = AVCC0 = VREFH0 = 2.4 to 5.5 V[*5], VSS = AVSS0 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
PCLKD (ADCLK) frequency		1	—	16	MHz	—
Analog input capacitance[*2]	Cs	—	—	9[*3]	pF	High-precision channel
		—	—	10[*3]	pF	Normal-precision channel
Analog input resistance	Rs	—	—	2.2[*3]	kΩ	High-precision channel
		—	—	7[*3]	kΩ	Normal-precision channel
Analog input voltage range	Ain	0	—	VREFH0	V	—
Resolution		—	—	12	Bit	—

Table 35.39 A/D conversion characteristics (5) in low-power A/D conversion mode (2 of 2)

Conditions: VCC = AVCC0 = VREFH0 = 2.4 to 5.5 V[*5], VSS = AVSS0 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
Conversion time[*1] (Operation at PCLKD = 16 MHz)	Permissible signal source impedance Max. = 2.2 kΩ	2.38 (0.656)[*4]	—	—	µs	High-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 0x0A ADACSR.ADSAC = 1
		3.0 (1.281)[*4]	—	—	µs	Normal-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 0x14 ADACSR.ADSAC = 1
Offset error		—	±1.25	±6.0	LSB	High-precision channel
				±7.5	LSB	Other than specified
Full-scale error		—	±1.25	±6.0	LSB	High-precision channel
				±7.5	LSB	Other than specified
Quantization error		—	±0.5	—	LSB	—
Absolute accuracy		—	±3.25	±7.0	LSB	High-precision channel
				±10.0	LSB	Other than specified
DNL differential nonlinearity error		—	±1.5	—	LSB	—
INL integral nonlinearity error		—	±1.75	±4.0	LSB	—

Note: The characteristics apply when no pin functions other than 12-bit A/D converter input are used. Absolute accuracy does not include quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.
Note 1. The conversion time is the sum of the sampling time and the comparison time. The number of sampling states is indicated for the test conditions.
Note 2. Except for I/O input capacitance (Cin), see section 35.2.4. I/O V_{OH}, V_{OL}, and Other Characteristics.
Note 3. Reference data.
Note 4. ( ) lists sampling time.
Note 5. When VREFH0 < AVCC0, the MAX. values are as follows.
Absolute accuracy/Offset error/Full-scale error:
For voltage difference between AVCC0 and VREFH0, it should be added ±0.75 LSB/V to the Max spec.
INL integral non-linearity error:
For voltage difference between AVCC0 and VREFH0, it should be added ±0.2 LSB/V to the Max spec.

Table 35.40 A/D conversion characteristics (6) in low-power A/D conversion mode (1 of 2)

Conditions: VCC = AVCC0 = VREFH0 = 1.8 to 5.5 V[*5], VSS = AVSS0 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
PCLKD (ADCLK) frequency		1	—	8	MHz	—
Analog input capacitance[*2]	Cs	—	—	9[*3]	pF	High-precision channel
		—	—	10[*3]	pF	Normal-precision channel
Analog input resistance	Rs	—	—	6[*3]	kΩ	High-precision channel
		—	—	14[*3]	kΩ	Normal-precision channel
Analog input voltage range	Ain	0	—	VREFH0	V	—
Resolution		—	—	12	Bit	—
Conversion time[*1] (Operation at PCLKD = 8 MHz)	Permissible signal source impedance Max. = 5 kΩ	4.75 (1.313)[*4]	—	—	µs	High-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 0x0A ADACSR.ADSAC = 1
		6.0 (2.563)[*4]	—	—	µs	Normal-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 0x14 ADACSR.ADSAC = 1

Table 35.40 A/D conversion characteristics (6) in low-power A/D conversion mode (2 of 2)

Conditions: VCC = AVCC0 = VREFH0 = 1.8 to 5.5 V[*5], VSS = AVSS0 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter	Min	Typ	Max	Unit	Test conditions
Offset error	—	±1.25	±7.5	LSB	High-precision channel
			±10.0	LSB	Other than specified
Full-scale error	—	±1.5	±7.5	LSB	High-precision channel
			±10.0	LSB	Other than specified
Quantization error	—	±0.5	—	LSB	—
Absolute accuracy	—	±3.75	±9.5	LSB	High-precision channel
			±13.5	LSB	Other than specified
DNL differential nonlinearity error	—	±2.0	—	LSB	—
INL integral nonlinearity error	—	±2.25	±4.5	LSB	—

Note: The characteristics apply when no pin functions other than 12-bit A/D converter input are used. Absolute accuracy does not include quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.
Note 1. The conversion time is the sum of the sampling time and the comparison time. The number of sampling states is indicated for the test conditions.
Note 2. Except for I/O input capacitance (Cin), see section 35.2.4. I/O V_{OH}, V_{OL}, and Other Characteristics.
Note 3. Reference data.
Note 4. ( ) lists sampling time.
Note 5. When VREFH0 < AVCC0, the MAX. values are as follows.
Absolute accuracy/Offset error/Full-scale error:
For voltage difference between AVCC0 and VREFH0, it should be added ±0.75 LSB/V to the Max spec.
INL integral non-linearity error:
For voltage difference between AVCC0 and VREFH0, it should be added ±0.2 LSB/V to the Max spec.

Table 35.41 A/D conversion characteristics (7) in low-power A/D conversion mode (1 of 2)

Conditions: VCC = AVCC0 = VREFH0 = 1.6 to 5.5 V[*5], VSS = AVSS0 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
PCLKD (ADCLK) frequency		1	—	4	MHz	—
Analog input capacitance[*2]	Cs	—	—	9[*3]	pF	High-precision channel
		—	—	10[*3]	pF	Normal-precision channel
Analog input resistance	Rs	—	—	12[*3]	kΩ	High-precision channel
		—	—	28[*3]	kΩ	Normal-precision channel
Analog input voltage range	Ain	0	—	VREFH0	V	—
Resolution		—	—	12	Bit	—
Conversion time[*1] (Operation at PCLKD = 4 MHz)	Permissible signal source impedance Max. = 9.9 kΩ	9.5 (2.625)[*4]	—	—	µs	High-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 0x0A ADACSR.ADSAC = 1
		12.0 (5.125)[*4]	—	—	µs	Normal-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 0x14 ADACSR.ADSAC = 1
Offset error		—	±1.25	±7.5	LSB	High-precision channel
				±10.0	LSB	Other than specified
Full-scale error		—	±1.5	±7.5	LSB	High-precision channel
				±10.0	LSB	Other than specified
Quantization error		—	±0.5	—	LSB	—

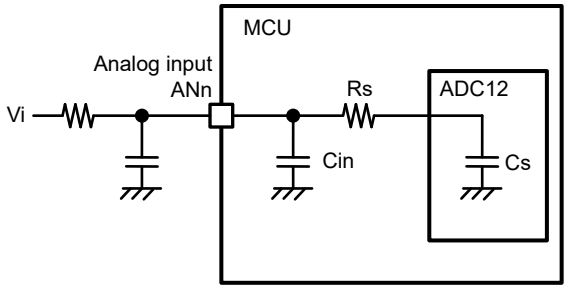
Table 35.41 A/D conversion characteristics (7) in low-power A/D conversion mode (2 of 2)

Conditions: VCC = AVCC0 = VREFH0 = 1.6 to 5.5 V[*5], VSS = AVSS0 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter	Min	Typ	Max	Unit	Test conditions
Absolute accuracy	—	±3.75	±9.5	LSB	High-precision channel
			±13.5	LSB	Other than specified
DNL differential nonlinearity error	—	±2.0	—	LSB	—
INL integral nonlinearity error	—	±2.25	±4.5	LSB	—

Note: The characteristics apply when no pin functions other than 12-bit A/D converter input are used. Absolute accuracy does not include quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.
Note 1. The conversion time is the sum of the sampling time and the comparison time. The number of sampling states is indicated for the test conditions.
Note 2. Except for I/O input capacitance (Cin), see section 35.2.4. I/O V_{OH}, V_{OL}, and Other Characteristics.
Note 3. Reference data.
Note 4. ( ) lists sampling time.
Note 5. When VREFH0 < AVCC0, the MAX. values are as follows.
Absolute accuracy/Offset error/Full-scale error:
For voltage difference between AVCC0 and VREFH0, it should be added ±0.75 LSB/V to the Max spec.
INL integral non-linearity error:
For voltage difference between AVCC0 and VREFH0, it should be added ±0.2 LSB/V to the Max spec.

Figure 35.49 shows the equivalent circuit for analog input.



Note: Terminal leakage current is not shown in this figure.

Figure 35.49 Equivalent circuit for analog input

Table 35.42 12-bit A/D converter channel classification

Classification	Channel	Conditions	Remarks
High-precision channel	AN000 to AN002, AN005 to AN010	AVCC0 = 1.6 to 5.5 V	Pins AN000 to AN002, AN005 to AN010 cannot be used as general I/O when the A/D converter is in use.
Normal-precision channel	AN019 to AN022		
Internal reference voltage input channel	Internal reference voltage	AVCC0 = 1.8 to 5.5 V	—
Temperature sensor input channel	Temperature sensor output	AVCC0 = 1.8 to 5.5 V	—

Table 35.43 A/D internal reference voltage characteristics

Conditions: VCC = AVCC0 = VREFH0 = 1.8 to 5.5 V[*1]

Parameter	Min	Typ	Max	Unit	Test conditions
Internal reference voltage input channel[*2]	1.42	1.48	1.54	V	—
PCLKD (ADCLK) frequency[*3]	1	—	2	MHz	—
Sampling time[*4]	5.0	—	—	µs	—

Note 1. The internal reference voltage cannot be selected for input channels when AVCC0 < 1.8 V.
Note 2. The 12-bit A/D internal reference voltage indicates the voltage when the internal reference voltage is input to the 12-bit A/D converter.
Note 3. When the internal reference voltage is selected as the high-potential reference voltage.
Note 4. When the internal reference voltage is converted.

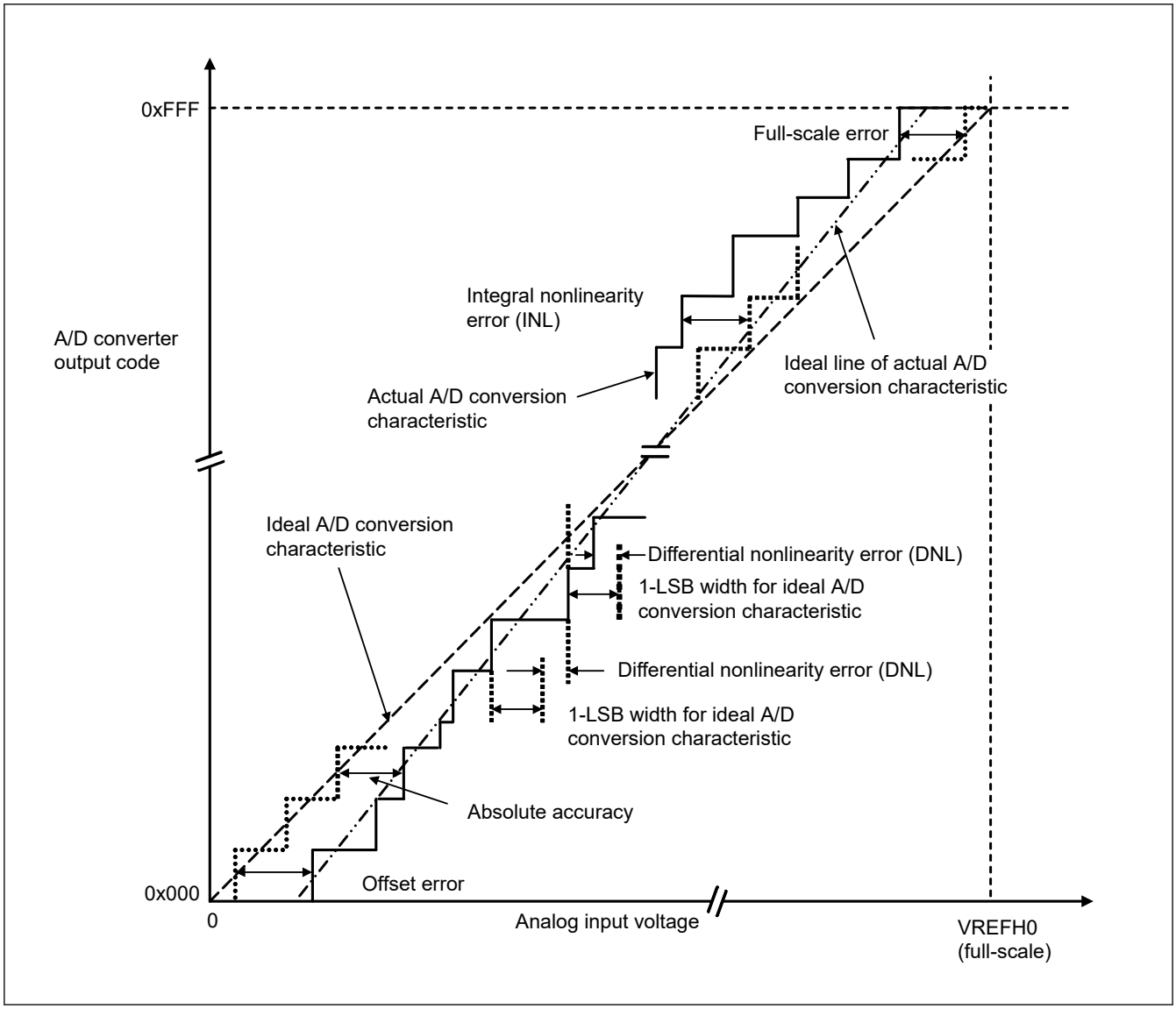


Figure 35.50 Illustration of 12-bit A/D converter characteristic terms

Absolute accuracy

Absolute accuracy is the difference between output code based on the theoretical A/D conversion characteristics, and the actual A/D conversion result. When measuring absolute accuracy, the voltage at the midpoint of the width of the analog input voltage (1-LSB width), which can meet the expectation of outputting an equal code based on the theoretical A/D conversion characteristics, is used as the analog input voltage. For example, if 12-bit resolution is used and the reference voltage VREFH0 = 3.072 V, then 1-LSB width becomes 0.75 mV, and 0 mV, 0.75 mV, and 1.5 mV are used as the analog input voltages. If analog input voltage is 6 mV, an absolute accuracy of ±5 LSB means that the actual A/D conversion result is in the range of 0x003 to 0x00D, though an output code of 0x008 can be expected from the theoretical A/D conversion characteristics.

Integral nonlinearity error (INL)

Integral nonlinearity error is the maximum deviation between the ideal line when the measured offset and full-scale errors are zeroed, and the actual output code.

Differential nonlinearity error (DNL)

Differential nonlinearity error is the difference between 1-LSB width based on the ideal A/D conversion characteristics and the width of the actual output code.

Offset error

Offset error is the difference between the transition point of the ideal first output code and the actual first output code.

Full-scale error

Full-scale error is the difference between the transition point of the ideal last output code and the actual last output code.

35.5 TSN Characteristics

Table 35.44 TSN characteristics

Conditions: VCC = AVCC0 = 1.8 to 5.5 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Relative accuracy	—	—	± 1.5	—	°C	2.4 V or above
		—	± 2.0	—	°C	Below 2.4 V
Temperature slope	—	—	-3.3	—	mV/°C	—
Output voltage (at 25°C)	—	—	1.05	—	V	VCC = 3.3 V
Temperature sensor start time	t_{START}	—	—	5	µs	—
Sampling time	—	5	—	—	µs	

35.6 OSC Stop Detect Characteristics

Table 35.45 Oscillation stop detection circuit characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Detection time	t_{dr}	—	—	1	ms	Figure 35.51

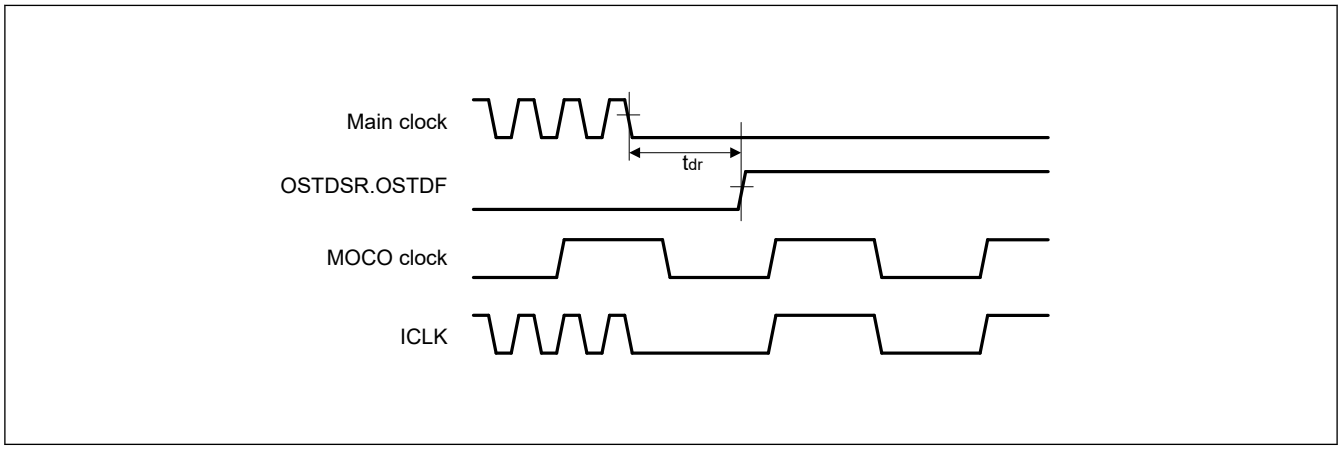


Figure 35.51 Oscillation stop detection timing

35.7 POR and LVD Characteristics

Table 35.46 Power-on reset circuit and voltage detection circuit characteristics (1) (1 of 2)

Parameter			Symbol	Min	Typ	Max	Unit	Test Conditions
Voltage detection level*1	Power-on reset (POR)	When power supply rise	V_{POR}	1.47	1.51	1.55	V	Figure 35.52
		When power supply fall	V_{PDR}	1.46	1.50	1.54		Figure 35.53
	Voltage detection circuit (LVD0)*2	When power supply rise	V_{det0_0}	3.74	3.91	4.06	V	Figure 35.54 At falling edge VCC
		When power supply fall		3.68	3.85	4.00		
		When power supply rise	V_{det0_1}	2.73	2.9	3.01		
		When power supply fall		2.68	2.85	2.96		
		When power supply rise	V_{det0_2}	2.44	2.59	2.70		
		When power supply fall		2.38	2.53	2.64		
		When power supply rise	V_{det0_3}	1.83	1.95	2.07		
		When power supply fall		1.78	1.90	2.02		
		When power supply rise	V_{det0_4}	1.66	1.75	1.88		
		When power supply fall		1.60	1.69	1.82		
Voltage detection level*1	Voltage detection circuit (LVD1)*3	When power supply rise	V_{det1_0}	4.23	4.39	4.55	V	Figure 35.55 At falling edge VCC
		When power supply fall		4.13	4.29	4.45		
		When power supply rise	V_{det1_1}	4.07	4.25	4.39		
		When power supply fall		3.98	4.16	4.30		
		When power supply rise	V_{det1_2}	3.97	4.14	4.29		
		When power supply fall		3.86	4.03	4.18		
		When power supply rise	V_{det1_3}	3.74	3.92	4.06		
		When power supply fall		3.68	3.86	4.00		
		When power supply rise	V_{det1_4}	3.05	3.17	3.29		
		When power supply fall		2.98	3.10	3.22		
		When power supply rise	V_{det1_5}	2.95	3.06	3.17		
		When power supply fall		2.89	3.00	3.11		
		When power supply rise	V_{det1_6}	2.86	2.97	3.08		
		When power supply fall		2.79	2.90	3.01		
		When power supply rise	V_{det1_7}	2.74	2.85	2.96		
		When power supply fall		2.68	2.79	2.90		

Table 35.46 Power-on reset circuit and voltage detection circuit characteristics (1) (2 of 2)

Parameter			Symbol	Min	Typ	Max	Unit	Test Conditions
Voltage detection level*1	Voltage detection circuit (LVD1)*3	When power supply rise	V_{det1_8}	2.63	2.75	2.85	V	Figure 35.55 At falling edge VCC
		When power supply fall		2.58	2.68	2.78		
		When power supply rise	V_{det1_9}	2.54	2.64	2.75		
		When power supply fall		2.48	2.58	2.68		
		When power supply rise	V_{det1_A}	2.43	2.53	2.63		
		When power supply fall		2.38	2.48	2.58		
		When power supply rise	V_{det1_B}	2.16	2.26	2.36		
		When power supply fall		2.10	2.20	2.30		
		When power supply rise	V_{det1_C}	1.88	2	2.09		
		When power supply fall		1.84	1.96	2.05		
		When power supply rise	V_{det1_D}	1.78	1.9	1.99		
		When power supply fall		1.74	1.86	1.95		
		When power supply rise	V_{det1_E}	1.67	1.79	1.88		
		When power supply fall		1.63	1.75	1.84		
		When power supply rise	V_{det1_F}	1.65	1.7	1.78		
		When power supply fall		1.60	1.65	1.73		
Voltage detection level*1	Voltage detection circuit (LVD2)*4	When power supply rise	V_{det2_0}	4.20	4.40	4.57	V	Figure 35.56 At falling edge VCC
		When power supply fall		4.11	4.31	4.48		
		When power supply rise	V_{det2_1}	4.05	4.25	4.42		
		When power supply fall		3.97	4.17	4.34		
		When power supply rise	V_{det2_2}	3.91	4.11	4.28		
		When power supply fall		3.83	4.03	4.20		
		When power supply rise	V_{det2_3}	3.71	3.91	4.08		
		When power supply fall		3.64	3.84	4.01		

Note 1. These characteristics apply when noise is not superimposed on the power supply. When a setting causes this voltage detection level to overlap with that of the voltage detection circuit, it cannot be specified whether LVD1 or LVD2 is used for voltage detection.
Note 2. # in the symbol $V_{det0_\#}$ denotes the value of the OFS1.VDSEL0[2:0] bits.
Note 3. # in the symbol $V_{det1_\#}$ denotes the value of the LVDLVLR.LVD1LVL[4:0] bits.
Note 4. # in the symbol $V_{det2_\#}$ denotes the value of the LVDLVLR.LVD2LVL[2:0] bits.

Table 35.47 Power-on reset circuit and voltage detection circuit characteristics (2) (1 of 2)

Parameter		Symbol	Min	Typ	Max	Unit	Test Conditions
Wait time after power-on reset cancellation	LVD0: enable	t_{POR}	—	4.3	—	ms	—
	LVD0: disable	t_{POR}	—	3.7	—	ms	—
Wait time after voltage monitor 0, 1, 2 reset cancellation	LVD0: enable*1	$t_{LVD0,1,2}$	—	1.4	—	ms	—
	LVD0: disable*2	$t_{LVD1,2}$	—	0.7	—	ms	—
Power-on reset response delay time*3		t_{det}	—	—	500	µs	Figure 35.52, Figure 35.53
LVD0 response delay time*3		t_{det}	—	—	500	µs	Figure 35.54
LVD1 response delay time*3		t_{det}	—	—	350	µs	Figure 35.55
LVD2 response delay time*3		t_{det}	—	—	600	µs	Figure 35.56
Minimum VCC down time		t_{VOFF}	500	—	—	µs	Figure 35.52, VCC = 1.0 V or above
Power-on reset enable time		$t_{W\ (POR)}$	1	—	—	ms	Figure 35.53, VCC = below 1.0 V

Table 35.47 Power-on reset circuit and voltage detection circuit characteristics (2) (2 of 2)

Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions
LVD1 operation stabilization time (after LVD1 is enabled)	$T_{d\ (E-A)}$	—	—	300	µs	Figure 35.55
LVD2 operation stabilization time (after LVD2 is enabled)	$T_{d\ (E-A)}$	—	—	1200	µs	Figure 35.56
Hysteresis width (POR)	V_{PORH}	—	10	—	mV	—
Hysteresis width (LVD0, LVD1 and LVD2)	V_{LVH}	—	60	—	mV	LVD0 selected
		—	110	—		V_{det1_0} to V_{det1_2} selected
		—	70	—		V_{det1_3} to V_{det1_9} selected
		—	60	—		V_{det1_A} to V_{det1_B} selected
		—	50	—		V_{det1_C} to V_{det1_F} selected
		—	90	—		LVD2 selected

Note 1. When OFS1.LVDAS = 0.
Note 2. When OFS1.LVDAS = 1.
Note 3. The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels V_{POR}, V_{det0}, V_{det1}, and V_{det2} for the POR/LVD.

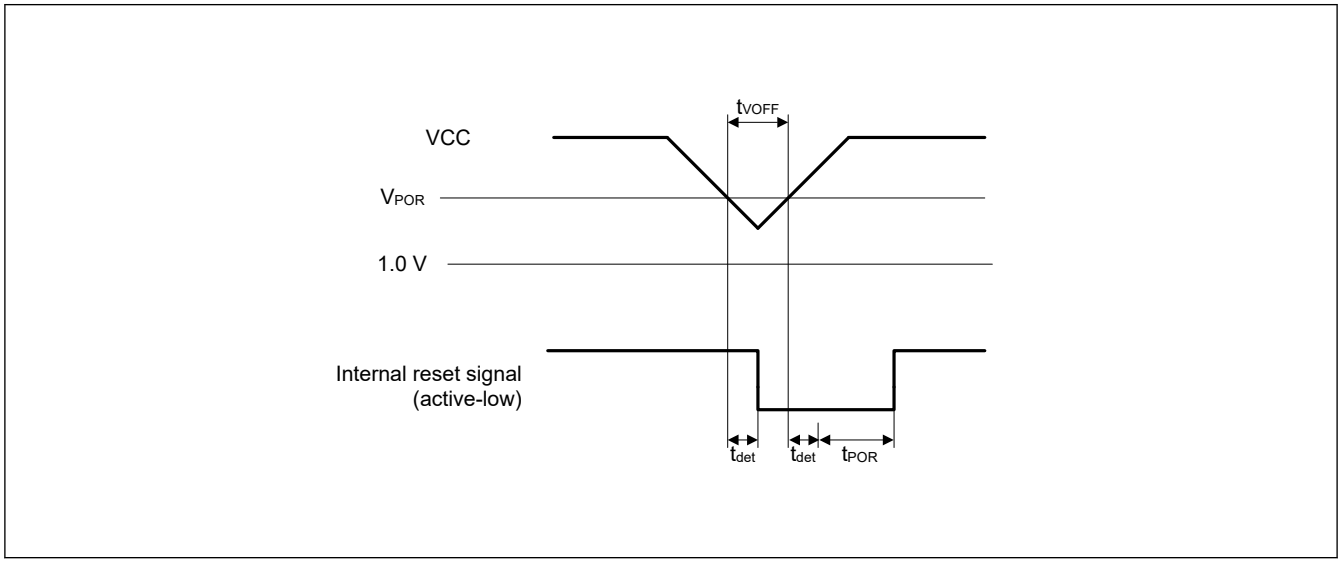
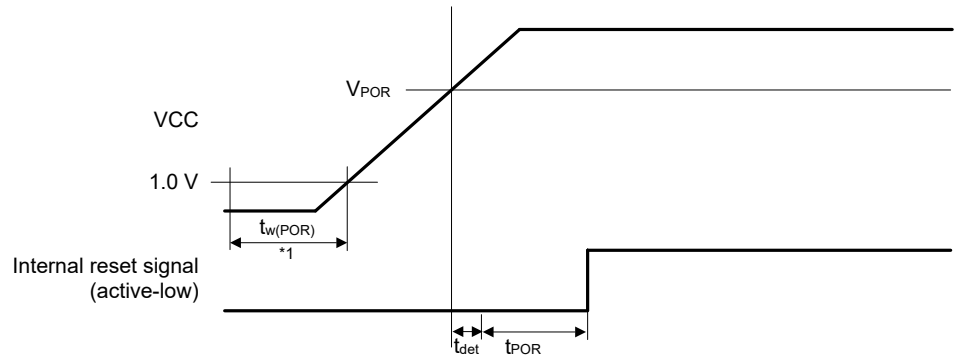


Figure 35.52 Voltage detection reset timing



Note 1. $t_{w(POR)}$ is the time required for a power-on reset to be enabled while the external power VCC is being held below the valid voltage (1.0 V).
When VCC turns on, maintain $t_{w(POR)}$ for 1.0 ms or more.

Figure 35.53 Power-on reset timing

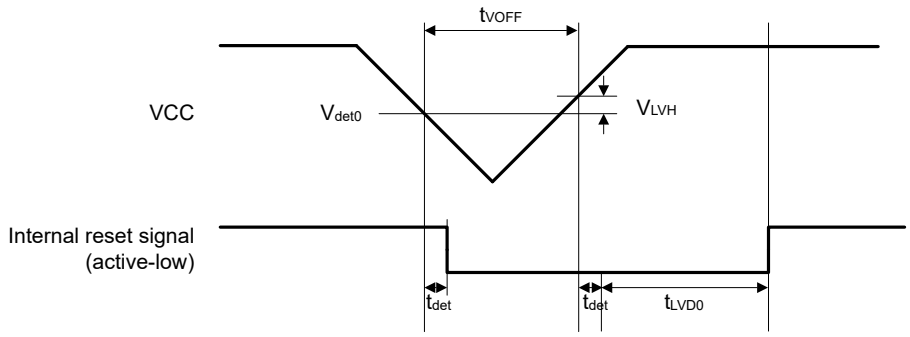


Figure 35.54 Voltage detection circuit timing (V_{det0})

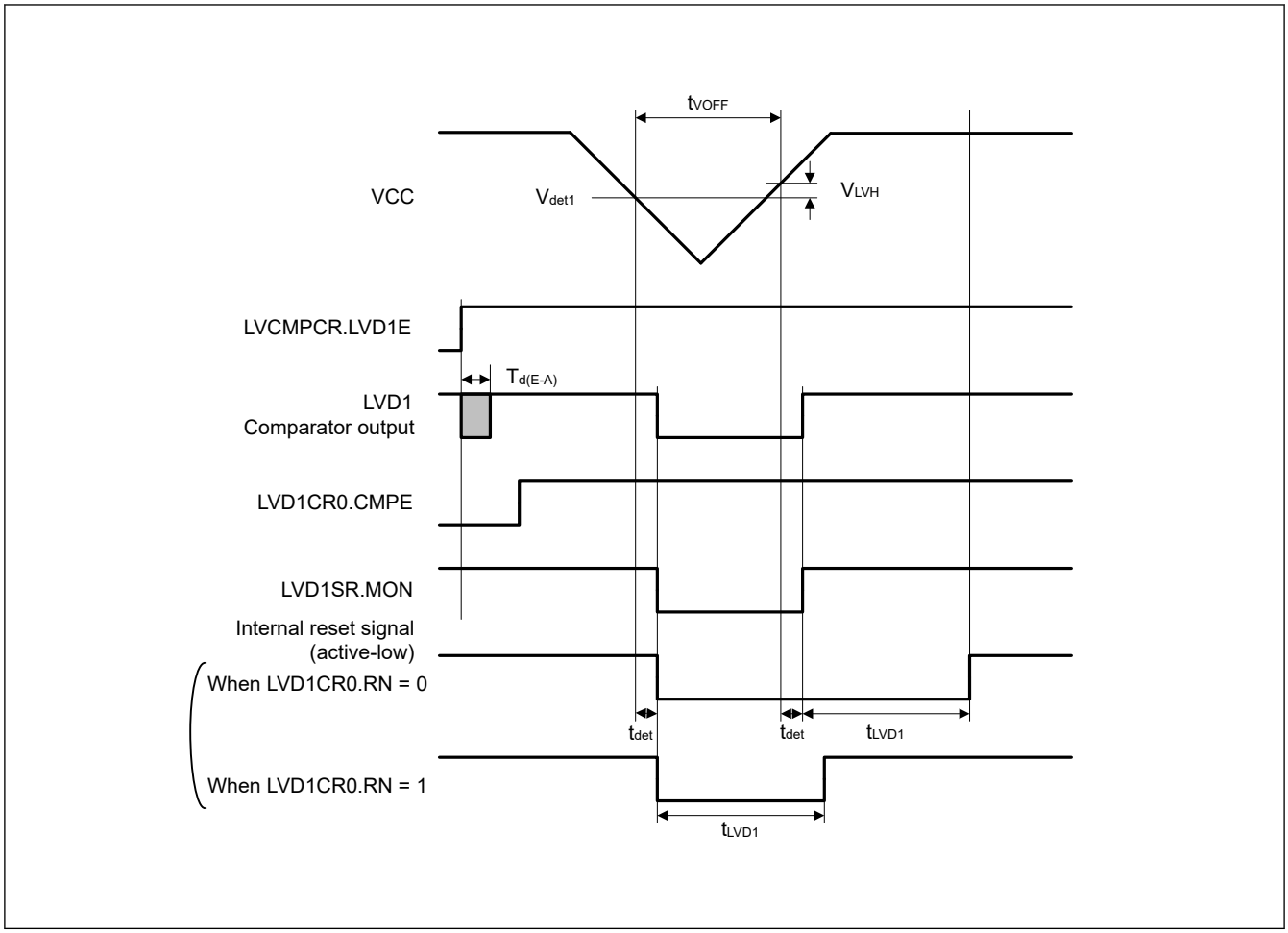


Figure 35.55 Voltage detection circuit timing (V_{det1})

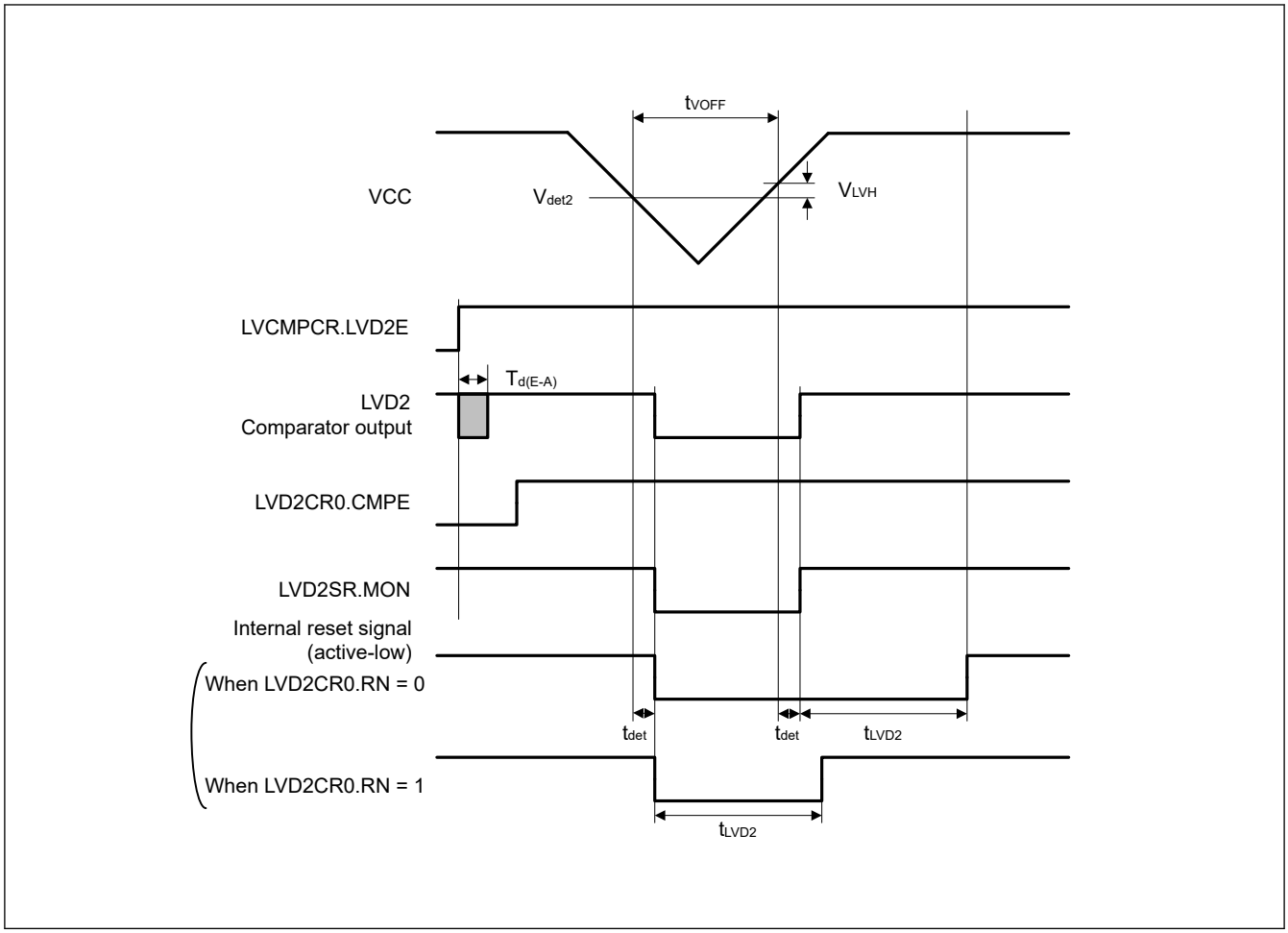


Figure 35.56 Voltage detection circuit timing (V_{det2})

35.8 Comparator Characteristics

Table 35.48 ACMPHS characteristics

Conditions: VCC = AVCC0 = 2.7 to 5.5 V, VSS = AVSS0 = 0 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Reference voltage range	VREF	0		VCC	V	—
Input voltage range	VI	0		VCC	V	—
Internal reference voltage	—	1.34	1.44	1.54	V	—
Output delay time*1	Td		50	100	ns	VI = VREF ± 100 mV
Offset voltage	—		±5	±40	mV	—
Operation stabilization wait time	Tcmp	1			µs	—
Input channel switch stabilization wait time*2	—	0.3	—	—	µs	—

Note 1. This value is the internal propagation delay.
Note 2. The time from the operation permission signal switching of the comparator (CMPnEN = 0 to 1) until the comparator is satisfied with the DC / AC characteristics.

35.9 Flash Memory Characteristics

35.9.1 Code Flash Memory Characteristics

Table 35.49 Code flash characteristics (1)

Parameter		Symbol	Min	Typ	Max	Unit	Conditions
Reprogramming/erasure cycle*1		N_{PEC}	10000	—	—	Times	—
Data hold time	After 1000 times N_{PEC}	t_{DRP}	20 *2	—	—	Year	T_a = +105°C
	After 10000 times NPEC		10 *2	—	—	Year	T_a = +125°C

Note 1. The reprogram/erase cycle is the number of erasures for each block. When the reprogram/erase cycle is n times (n = 10,000), erasing can be performed n times for each block. For instance, when 4-byte programming is performed 512 times for different addresses in 2-KB blocks, and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasure is not enabled (overwriting is prohibited).

Note 2. This result is obtained from reliability testing.

Table 35.50 Code flash characteristics (2)

High-speed operating mode
Conditions: VCC = AVCC0 = 1.8 to 5.5 V

Parameter		Symbol	ICLK = 1 MHz			ICLK = 64 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	4-byte	t_{P4}	—	86	732	—	34	321	µs
Erasure time	2-KB	t_{E2K}	—	12.5	355	—	5.6	215	ms
Blank check time	4-byte	t_{BC4}	—	—	46.5	—	—	8.3	µs
	2-KB	t_{BC2K}	—	—	3681	—	—	231	µs
Erase suspended time		t_{SED}	—	—	22.3	—	—	10.5	µs
Access window information program Start-up area selection and security setting time		t_{AWSSAS}	—	21.2	570	—	11.4	423	ms
OCD/serial programmer ID setting time*1		t_{OSIS}	—	84.7	2280	—	45.3	1690	ms
Flash memory mode transition wait time 1		t_{DIS}	2	—	—	2	—	—	µs
Flash memory mode transition wait time 2		t_{MS}	15	—	—	15	—	—	µs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.

Note: The lower-limit frequency of ICLK is 1 MHz during programming or erasing the flash memory. When using ICLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.

Note: The frequency accuracy of ICLK must be ± 1.0% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.

Note 1. Total time of four commands.

Table 35.51 Code flash characteristics (3) (1 of 2)

Middle-speed operating mode
Conditions: VCC = AVCC0 = 1.8 to 5.5 V

Parameter		Symbol	ICLK = 1 MHz			ICLK = 24 MHz*2			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	4-byte	t_{P4}	—	86	732	—	39	356	µs
Erasure time	2-KB	t_{E2K}	—	12.5	355	—	6.2	227	ms
Blank check time	4-byte	t_{BC4}	—	—	46.5	—	—	11.3	µs
	2-KB	t_{BC2K}	—	—	3681	—	—	534	µs
Erase suspended time		t_{SED}	—	—	22.3	—	—	11.7	µs
Access window information program Start-up area selection and security setting time		t_{AWSSAS}	—	21.2	570	—	12.2	435	ms

Table 35.51 Code flash characteristics (3) (2 of 2)

Middle-speed operating mode
Conditions: VCC = AVCC0 = 1.8 to 5.5 V

Parameter	Symbol	ICLK = 1 MHz			ICLK = 24 MHz*2			Unit
		Min	Typ	Max	Min	Typ	Max	
OCD/serial programmer ID setting time*1	t_{OSIS}	—	84.7	2280	—	48.7	1740	ms
Flash memory mode transition wait time 1	t_{DIS}	2	—	—	2	—	—	µs
Flash memory mode transition wait time 2	t_{MS}	15	—	—	15	—	—	µs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of ICLK is 1 MHz during programming or erasing the flash memory. When using ICLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of ICLK must be ± 1.0% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.
Note 1. Total time of four commands.
Note 2. When 1.8 V ≤ VCC = AVCC0 ≤ 5.5 V

Table 35.52 Code flash characteristics (4)

Low-speed operating mode
Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter		Symbol	ICLK = 1 MHz			ICLK = 2 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	4-byte	t_{P4}	—	86	732	—	57	502	µs
Erasure time	2-KB	t_{E2K}	—	12.5	355	—	8.8	280	ms
Blank check time	4-byte	t_{BC4}	—	—	46.5	—	—	23.3	µs
	2-KB	t_{BC2K}	—	—	3681	—	—	1841	µs
Erase suspended time		t_{SED}	—	—	22.3	—	—	16.2	µs
Access window information program Start-up area selection and security setting time		t_{AWSSAS}	—	21.2	570	—	15.9	491	ms
OCD/serial programmer ID setting time*1		t_{OSIS}	—	84.7	2280	—	63.5	1964	ms
Flash memory mode transition wait time 1		t_{DIS}	2	—	—	2	—	—	µs
Flash memory mode transition wait time 2		t_{MS}	15	—	—	15	—	—	µs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of ICLK is 1 MHz during programming or erasing the flash memory. When using ICLK at below 4 MHz, the frequency can be set to 1 MHz or 2 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of ICLK must be ± 1.0% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.
Note 1. Total time of four commands.

35.9.2 Data Flash Memory Characteristics

Table 35.53 Data flash characteristics (1) (1 of 2)

Parameter	Symbol	Min	Typ	Max	Unit	Conditions
Reprogramming/erasure cycle*1	N_{DPEC}	100000	1000000	—	Times	—

Table 35.53 Data flash characteristics (1) (2 of 2)

Parameter		Symbol	Min	Typ	Max	Unit	Conditions
Data hold time	After 10000 times of N_{DPEC}	t_{DDRP}	20 *2	—	—	Year	Ta = +105°C
	After 10000 times of N_{DPEC}		10	—	—	Year	Ta = +125°C
	After 100000 times of N_{DPEC}		5 *2	—	—	Year	Ta = +105°C
	After 1000000 times of N_{DPEC}		—	1 *2	—	Year	Ta = +25°C

Note 1. The reprogram/erase cycle is the number of erasure for each block. When the reprogram/erase cycle is n times (n = 100,000), erasing can be performed n times for each block. For instance, when 1-byte programming is performed 1,024 times for different addresses in 1-KB blocks, and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasure is not enabled. (overwriting is prohibited.)

Note 2. This result is obtained from reliability testing.

Table 35.54 Data flash characteristics (2)

High-speed operating mode
Conditions: VCC = AVCC0 = 1.8 to 5.5 V

Parameter		Symbol	ICLK = 1 MHz			ICLK = 64 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	1-byte	t_{DP1}	—	45	404	—	34	321	µs
Erasure time	1-KB	t_{DE1K}	—	8.8	280	—	6.1	224	ms
Blank check time	1-byte	t_{DBC1}	—	—	15.2	—	—	8.3	µs
	1-KB	t_{DBC1K}	—	—	1832	—	—	444	µs
Suspended time during erasing		t_{DSED}	—	—	13.2	—	—	10.5	µs
Data flash STOP recovery time		t_{DSTOP}	250	—	—	250	—	—	ns

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.

Note: The lower-limit frequency of ICLK is 1 MHz during programming or erasing the flash memory. When using ICLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.

Note: The frequency accuracy of ICLK must be ± 1.0% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.

Table 35.55 Data flash characteristics (3)

Middle-speed operating mode
Conditions: VCC = AVCC0 = 1.8 to 5.5 V

Parameter		Symbol	ICLK = 1 MHz			ICLK = 24 MHz*1			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	1-byte	t_{DP1}	—	45	404	—	39	356	µs
Erasure time	1-KB	t_{DE1K}	—	8.8	280	—	7.3	248	ms
Blank check time	1-byte	t_{DBC1}	—	—	15.2	—	—	11.3	µs
	1-KB	t_{DBC1K}	—	—	1.84	—	—	1.06	ms
Suspended time during erasing		t_{DSED}	—	—	13.2	—	—	11.7	µs
Data flash STOP recovery time		t_{DSTOP}	250	—	—	250	—	—	ns

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.

Note: The lower-limit frequency of ICLK is 1 MHz during programming or erasing the flash memory. When using ICLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.

Note: The frequency accuracy of ICLK must be ± 1.0% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.

Note 1. When 1.8 V ≤ VCC = AVCC0 ≤ 5.5 V

Table 35.56 Data flash characteristics (4)

Low-speed operating mode
Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter		Symbol	ICLK = 1 MHz			ICLK = 2 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	1-byte	t_{DP1}	—	86	732	—	57	502	µs
Erasure time	1-KB	t_{DE1K}	—	19.7	504	—	12.4	354	ms
Blank check time	1-byte	t_{DBC1}	—	—	46.5	—	—	23.3	µs
	1-KB	t_{DBC1K}	—	—	7.3	—	—	3.66	ms
Suspended time during erasing		t_{DSED}	—	—	22.3	—	—	16.2	µs
Data flash STOP recovery time		t_{DSTOP}	250	—	—	250	—	—	ns

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of ICLK is 1 MHz during programming or erasing the flash memory. When using ICLK at below 2 MHz, the frequency can be set to 1 MHz or 2 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of ICLK must be ± 1.0% during programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.

35.10 Serial Wire Debug (SWD)

Table 35.57 SWD characteristics (1)

Conditions: VCC = AVCC0 = 2.4 to 5.5 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
SWCLK clock cycle time	$t_{SWCKcyc}$	80	—	—	ns	Figure 35.57
SWCLK clock high pulse width	t_{SWCKH}	35	—	—	ns	
SWCLK clock low pulse width	t_{SWCKL}	35	—	—	ns	
SWCLK clock rise time	t_{SWCKr}	—	—	5	ns	
SWCLK clock fall time	t_{SWCKf}	—	—	5	ns	
SWDIO setup time	t_{SWDS}	16	—	—	ns	Figure 35.58
SWDIO hold time	t_{SWDH}	16	—	—	ns	
SWDIO data delay time	t_{SWDD}	2	—	70	ns	

Table 35.58 SWD characteristics (2)

Conditions: VCC = AVCC0 = 1.6 to 2.4 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
SWCLK clock cycle time	$t_{SWCKcyc}$	250	—	—	ns	Figure 35.57
SWCLK clock high pulse width	t_{SWCKH}	120	—	—	ns	
SWCLK clock low pulse width	t_{SWCKL}	120	—	—	ns	
SWCLK clock rise time	t_{SWCKr}	—	—	5	ns	
SWCLK clock fall time	t_{SWCKf}	—	—	5	ns	
SWDIO setup time	t_{SWDS}	50	—	—	ns	Figure 35.58
SWDIO hold time	t_{SWDH}	50	—	—	ns	
SWDIO data delay time	t_{SWDD}	2	—	170	ns	

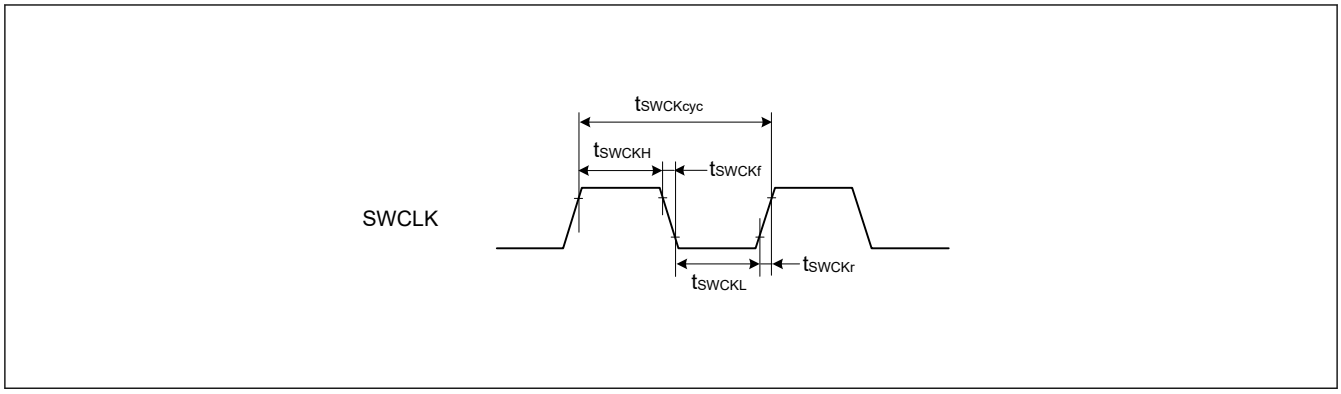


Figure 35.57 SWD SWCLK timing

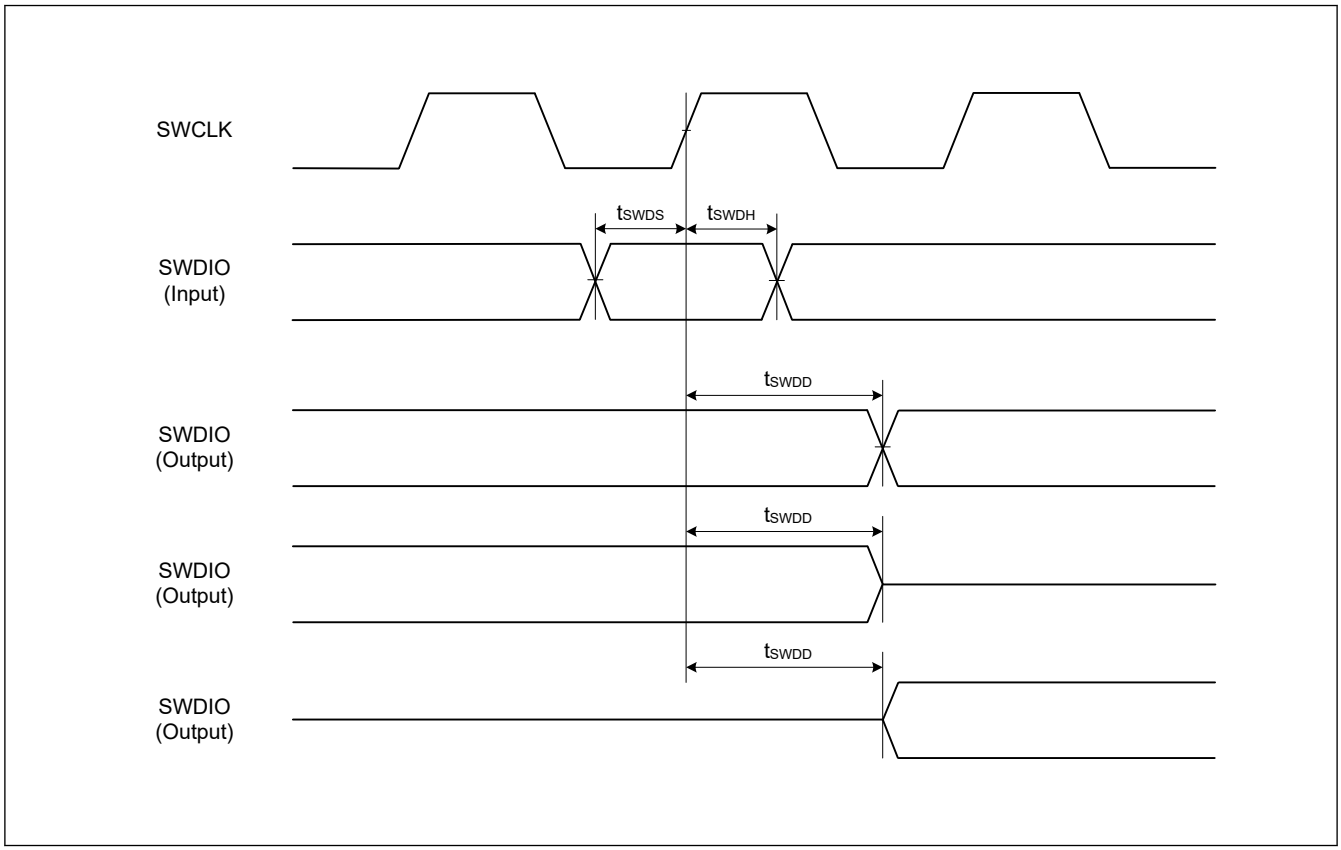


Figure 35.58 SWD input/output timing

Appendix 1. Port States in each Processing Mode

Table A1.1 Port states in each processing mode (1 of 2)

Port name	Reset	Software Standby Mode
P000/AN008	Hi-Z	Keep-O
P001/AN009/IRQ7_A	Hi-Z	Keep-O*1
P002/AN010	Hi-Z	Keep-O
P010/AN005	Hi-Z	Keep-O
P011/AN006	Hi-Z	Keep-O
P012/AN007	Hi-Z	Keep-O
P013/AN000/CMPIN03/GTIW_A/IRQ1_C	Hi-Z	Keep-O*1
P014/AN001/CMPIN02/GTIV_A/IRQ3_B	Hi-Z	Keep-O*1
P015/AN002/CMPIN01/GTIU_A/IRQ2_C	Hi-Z	Keep-O*1
P100/AGTIO0_A/GTOVLO_B/GTIOC1B_B/ RXD0_A/MISO0_A/SCL0_A/SCL0_D/ SCK1_A/MISOA_A/KRM00/AN022/IRQ0_C	Hi-Z	[AGTIO0_A output selected] AGTIO0_A output*2 [Other than the above] Keep-O*1
P101/AGTEE0/GTOVUP_B/GTIOC1A_B/ TXD0_A/MOSI0_A/SDA0_A/MOSIA_A/ SDA0_C/CTS1_RTS1_A/KRM01/AN021/ IRQ2_A	Hi-Z	Keep-O*1
P102/ADTRG0_A/AGTO0/GTOWLO_B/ GTIOC2B_A/AN020/CMPREF0/IRQ1_A/ SCK0_A/TXD2_D/MOSI2_D/SDA2_D/ RSPCKA_A/KRM02	Hi-Z	[AGTO0 selected] AGTO0 output*2 [Other than the above] Keep-O*1
P103/GTOWUP_B/GTIOC2A_A/RXD2_D/ MISO2_D/SCL2_DAN019/CMPIN1/IRQ4_B/ GTCPPO3/CTS0_RTS0_A/SSLA0_A/ KRM03	Hi-Z	Keep-O*1
P104/GTETRGB_B/RXD0_C/MISO0_C/ SCL0_C/SSLA1_A/KRM04/IRQ1_B	Hi-Z	Keep-O*1
P108/SWDIO/AGTEE1_B/GTETRGB_C/ GTIOC3B_A/RXD1/MISO1/SCL1/IRQ6_C/ GTCPPO2/CTS9_RTS9_B	Pull-up	Keep-O*1
P109/AGTOA1_B/GTOWUP_A/ GTIOC2A_B/SCK1_E/TXD9_B/MOSI9_B/ SDA9_B/CLKOUT_B	Hi-Z	[AGTOA1_B selected] AGTOA1_B output*2 [CLKOUT selected] CLKOUT output [Other than the above] Keep-O
P110/AGTIO1_A/GTOWLO_A/GTIOC2B_B/ VCOUT/CTS2_RTS2_B/RXD9_B/SCL9_B/ MISO9_B/IRQ3_A	Hi-Z	[AGTIO1 selected] AGTIO1 output*2 [ACMPHS selected] VCOUT output [Other than the above] Keep-O*1
P111/AGTOA0/GTIOC1A_C/SCK2_B/ SCK9_B/IRQ4_A	Hi-Z	[AGTOA0 selected] AGTOA0 output*2 [Other than the above] Keep-O*1
P112/AGTOB0/GTIOC1B_C/SCK1_D	Hi-Z	[AGTOB0 selected] AGTOB0 output*2 [Other than the above] Keep-O
P200/NMI	Hi-Z	Hi-Z

Table A1.1 Port states in each processing mode (2 of 2)

Port name	Reset	Software Standby Mode
P201/MD	Pull-up	Keep-O
P206/GTIU_B/TXD0_C/MOSI0_C/SDA0_C/IRQ0_B	Hi-Z	Keep-O*1
P207	Hi-Z	Keep-O
P208/AGTOB0_A/GTOUUP_B/GTIOC0A_B	Hi-Z	[AGTOB0_A selected] AGTOB0_A output*2 [Other than the above] Keep-O
P212/EXTAL/AGTEE1_A/GTOULO_A/GTIOC0B_A/IRQ7_C	Hi-Z	Keep-O*1
P213/XTAL/GTOUUP_A/GTIOC0A_A/IRQ2_B	Hi-Z	Keep-O*1
P300/SWCLK/GTETRGA_C/GTIOC3A_A/TXD1/MOSI1/SDA1/IRQ5_B/GTCPPO1	Pull-up	Keep-O*1
P301/AGTIO0_D/GTETRGB_A/RXD2_A/MISO2_A/SCL2_A/CTS9_RTS9_D/IRQ6_A	Hi-Z	[AGTIO0_D output selected] AGTIO0_D output*2 [Other than the above] Keep-O*1
P302/GTETRGA_D/TXD2_A/MOSI2_A/SDA2_A/IRQ5_A	Hi-Z	Keep-O*1
P400/CACREF_C/AGTIO1_C/GTIOC3A_B/SCK0_B/SCK1_B/SCL0_A/IRQ0_A	Hi-Z	[AGTIO1_C output selected] AGTIO1_C output*2 [Other than the above] Keep-O*1
P401/GTETRGA_B/GTIOC3B_B/CTS0_RTS0_B/SDA0_A/IRQ5_C	Hi-Z	Keep-O*1
P402/GTADSM0	Hi-Z	Keep-O
P403/GTADSM1	Hi-Z	Keep-O
P407/ADTRG0_B/AGTIO0_C/CMPREF1/GTOULO_B/GTIOC0B_B/GTCPPO0/CTS0_RTS0_D/SDA0_B	Hi-Z	[AGTIO0_C output selected] AGTIO0_C output*2 [Other than the above] Keep-O*1
P408/GTIW_B/CTS1_RTS1_D/SCL0_C/IRQ7_B	Hi-Z	Keep-O*1
P409/GTIV_B/IRQ6_B	Hi-Z	Keep-O*1
P500/GTETRGA_A/GTIOC2A_C	Hi-Z	Keep-O
P913/AGTIO1_F/GTOVLO_A/GTIOC1B_A	Hi-Z	[AGTIO1_F output selected] AGTIO1_F output*2 [Other than the above] Keep-O
P914/AGTOA1_A/GTOVUP_A/GTIOC1A_A	Hi-Z	[AGTOA1_A output selected] AGTOA1_A output*2 [Other than the above] Keep-O
P915	Hi-Z	Keep-O

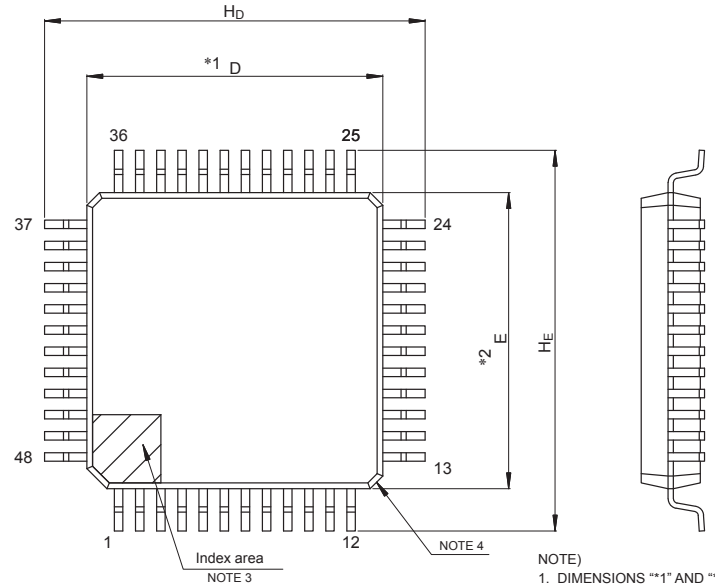
Note: Hi-Z: High-impedance
Keep-O: Output pins retain their previous values. Input pins become high-impedance.
Note 1. Input is enabled if the pin is specified as the software standby canceling source while it is used as an external interrupt pin.
Note 2. AGTIO output is enabled while LOCO is selected as a count source.

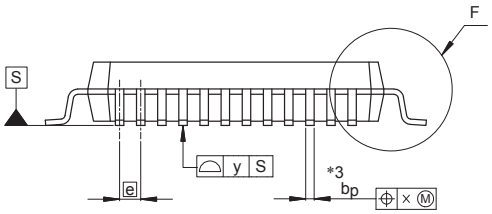
Appendix 2. Package Dimensions

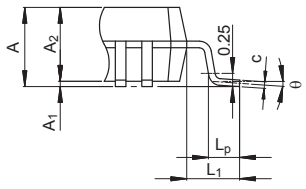
Information on the latest version of the package dimensions or mountings is displayed in “Packages” on the Renesas Electronics Corporation website.

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP48-7x7-0.50	PLQP0048KB-B	—	0.2

Unit: mm







Detail F

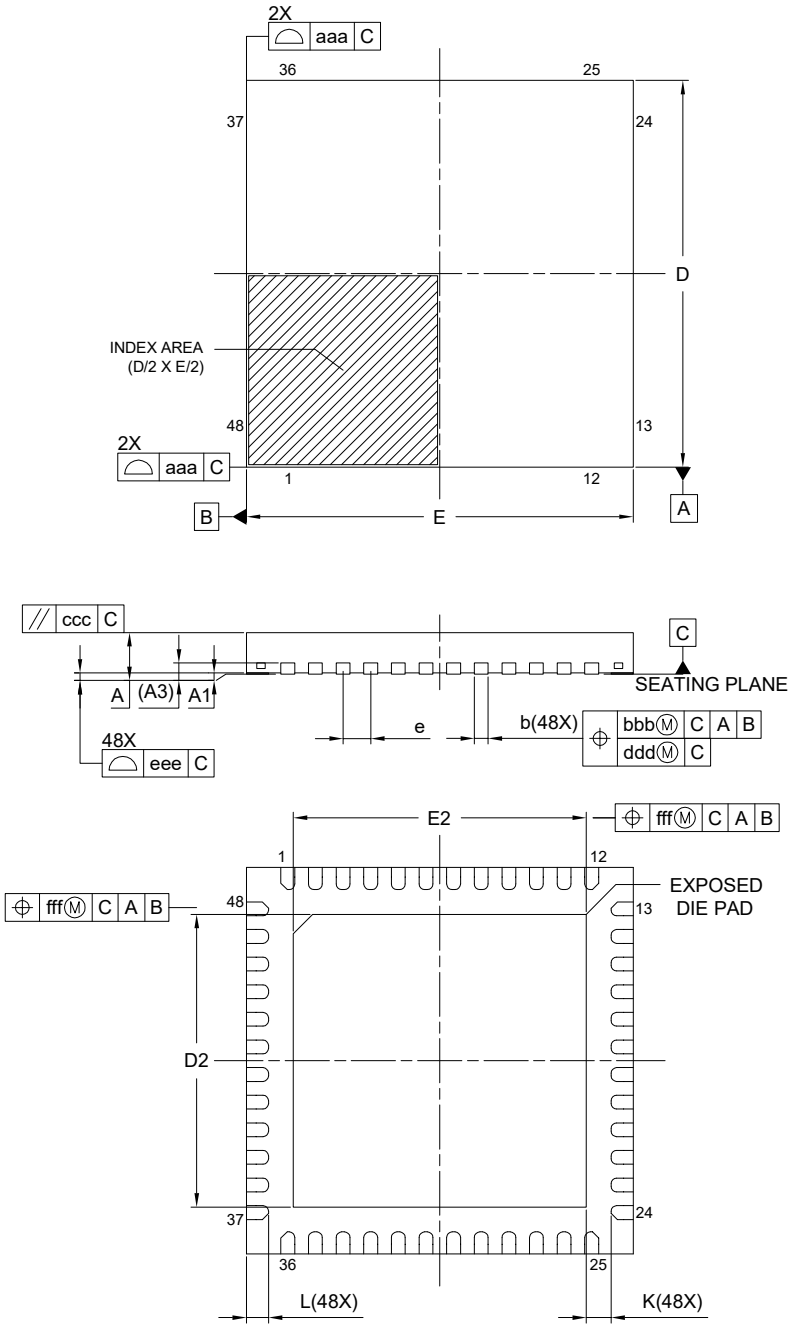
NOTE)
1. DIMENSIONS “*1” AND “*2” DO NOT INCLUDE MOLD FLASH.
2. DIMENSION “*3” DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A_2	—	1.4	—
H_D	8.8	9.0	9.2
H_E	8.8	9.0	9.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.17	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

© 2015 Renesas Electronics Corporation. All rights reserved.

Figure A2.1 LQFP 48-pin

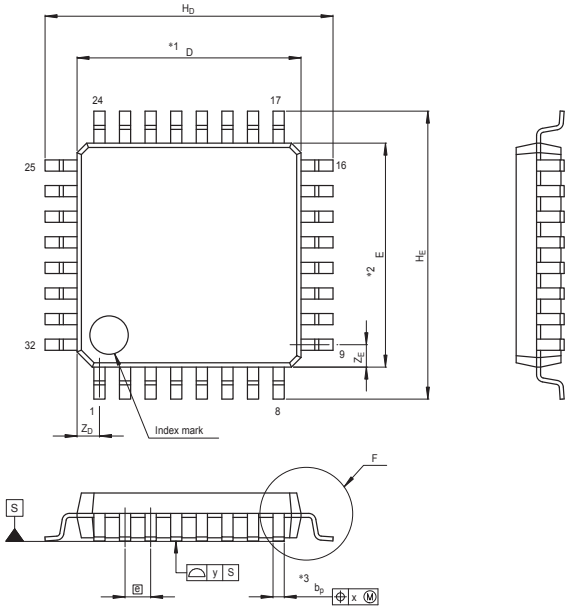
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN048-7x7-0.50	PWQN0048KC-A	0.13 g

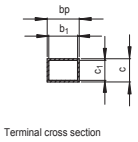


Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.20	0.25	0.30
D		7.00 BSC	
E		7.00 BSC	
e		0.50 BSC	
L	0.30	0.40	0.50
K	0.20	—	—
D_2	5.25	5.30	5.35
E_2	5.25	5.30	5.35
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

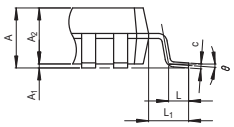
Figure A2.2 HWQFN 48-pin

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LQFP32-7x7-0.80	PLQP0032GB-A	32P6U-A	0.2g





Terminal cross section



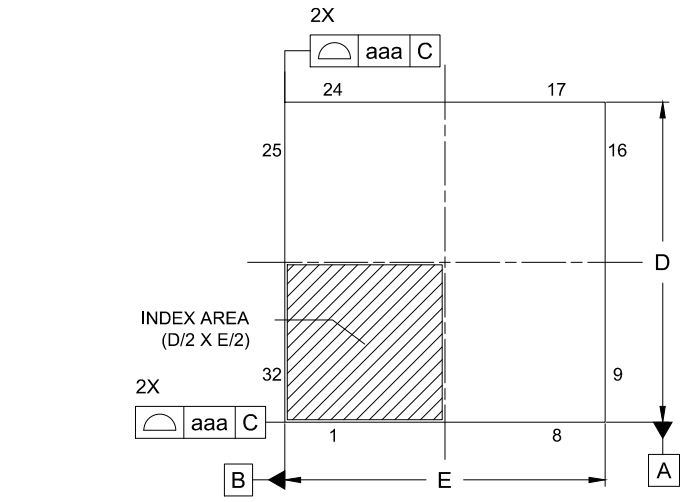
Detail F

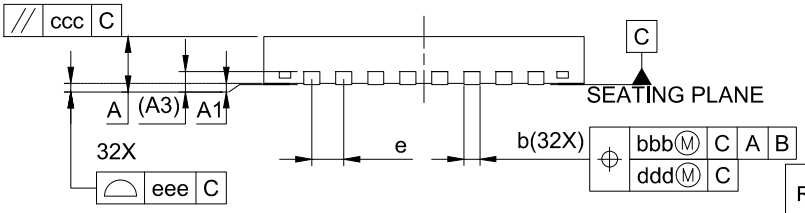
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

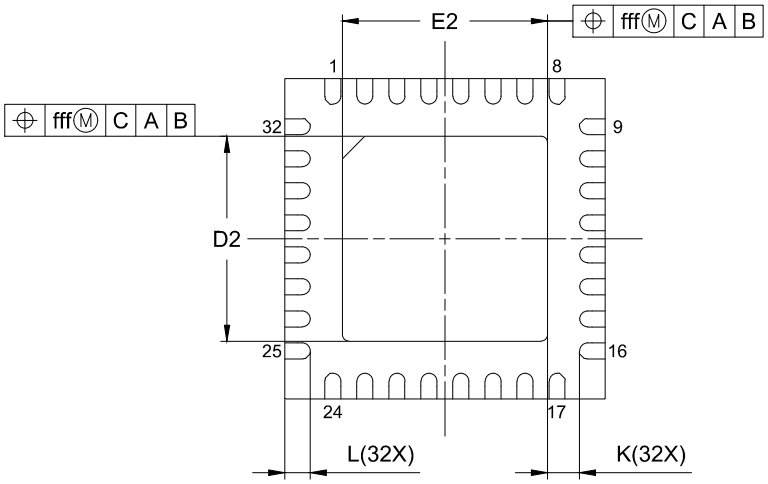
Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A_2	—	1.4	—
H_D	8.8	9.0	9.2
H_E	8.8	9.0	9.2
A	—	—	1.7
A_1	0	0.1	0.2
b_p	0.32	0.37	0.42
b_1	—	0.35	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.8	—
x	—	—	0.20
y	—	—	0.10
Z_D	—	0.7	—
Z_E	—	0.7	—
L	0.3	0.5	0.7
L_1	—	1.0	—

Figure A2.3 LQFP 32-pin

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN032-5x5-0.50	PWQN0032KE-A	0.06



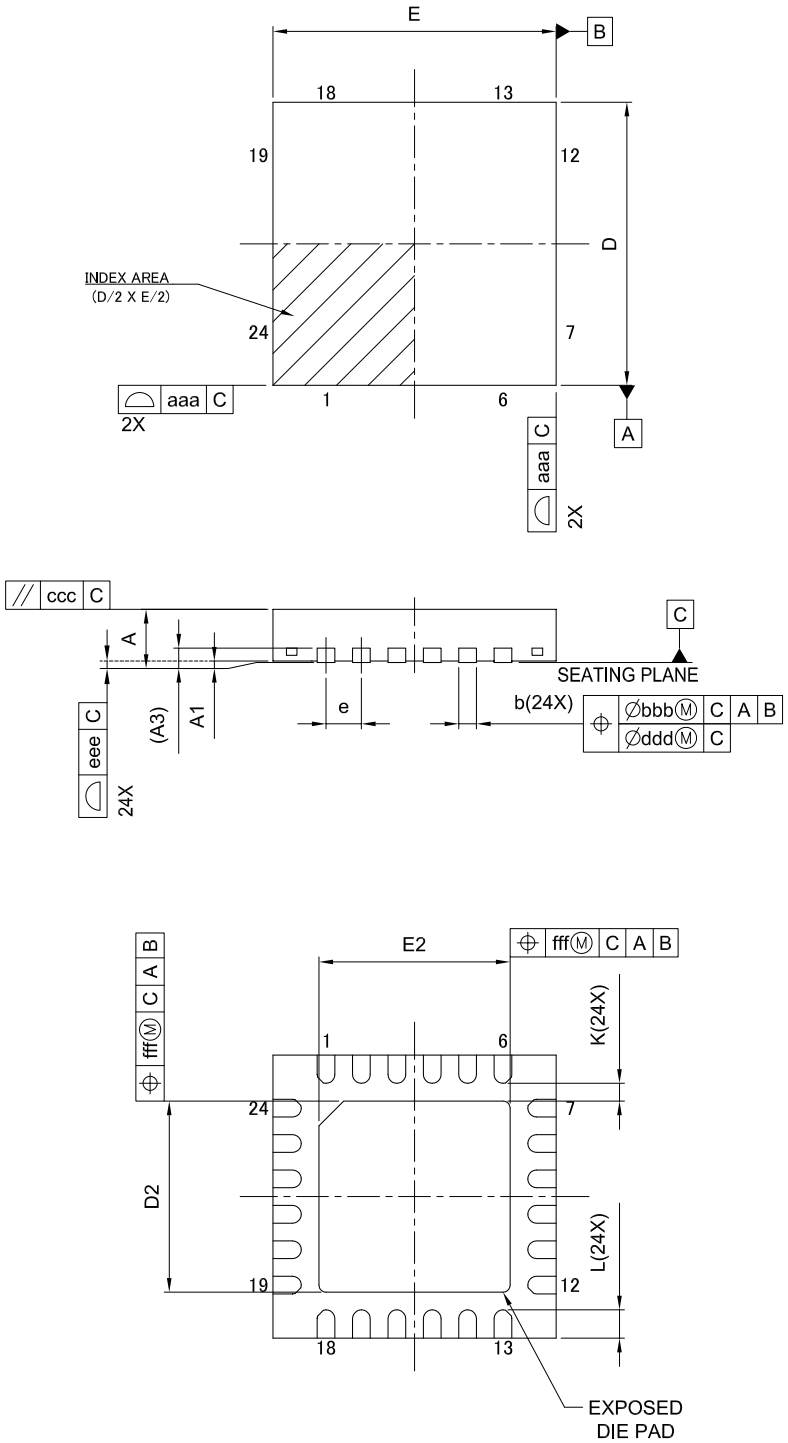




Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.18	0.25	0.30
D		5.00 BSC	
E		5.00 BSC	
e		0.50 BSC	
L	0.35	0.40	0.45
K	0.20	—	—
D_2	3.15	3.20	3.25
E_2	3.15	3.20	3.25
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

Figure A2.4 HWQFN 32-pin

JEITA Package Code	RENESAS Code	MASS(Typ.)[g]
P-HWQFN24-4×4-0.50	PWQN0024KG-A	0.04



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.18	0.25	0.30
D		4.00 BSC	
E		4.00 BSC	
e		0.50 BSC	
L	0.35	0.40	0.45
K	0.20	—	—
D_2	2.65	2.70	2.75
E_2	2.65	2.70	2.75
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

Figure A2.5 HWQFN 24-pin

Appendix 3. I/O Registers

This appendix describes I/O register addresses, access cycles, and reset values by function.

3.1 Peripheral Base Addresses

This section provides the base addresses for peripherals described in this manual.

Table A3.1 shows the name, description, and the base address of each peripheral.

Table A3.1 Peripheral base address (1 of 2)

Name	Description	Base address
RMPU	Renesas Memory Protection Unit	0x4000_0000
SRAM	SRAM Control	0x4000_2000
BUS	BUS Control	0x4000_3000
DTC	Data Transfer Controller	0x4000_5400
ICU	Interrupt Controller Unit	0x4000_6000
DBG	Debug Function	0x4001_B000
SYSC	System Control	0x4001_E000
PORT0	Port 0 Control Registers	0x4004_0000
PORT1	Port 1 Control Registers	0x4004_0020
PORT2	Port 2 Control Registers	0x4004_0040
PORT3	Port 3 Control Registers	0x4004_0060
PORT4	Port 4 Control Registers	0x4004_0080
PORT5	Port 5 Control Registers	0x4004_00A0
PORT9	Port 9 Control Registers	0x4004_0120
PFS	Pmn Pin Function Control Register	0x4004_0800
ELC	Event Link Controller	0x4004_1000
POEG	Port Output Enable Module for GPT	0x4004_2000
WDT	Watchdog Timer	0x4004_4200
IWDT	Independent Watchdog Timer	0x4004_4400
CAC	Clock Frequency Accuracy Measurement Circuit	0x4004_4600
MSTP	Module Stop Control B, C, D, Low Speed Module R/W Disable Control	0x4004_7000
IIC0	Inter-Integrated Circuit 0	0x4005_3000
IIC0WU	Inter-Integrated Circuit 0 Wakeup Unit	0x4005_3014
DOC	Data Operation Circuit	0x4005_4100
ADC120	12-bit A/D Converter	0x4005_C000
SCI0	Serial Communication Interface 0	0x4007_0000
SCI1	Serial Communication Interface 1	0x4007_0020
SCI2	Serial Communication Interface 2	0x4007_0040
SCI9	Serial Communication Interface 9	0x4007_0120
SPI0	Serial Peripheral Interface 0	0x4007_2000
CRC	Cyclic Redundancy Check	0x4007_4000
KINT	Key Interrupt Function	0x4008_0000
AGTW0	Low Power Asynchronous General Purpose Timer 0	0x4008_4000
AGTW1	Low Power Asynchronous General Purpose Timer 1	0x4008_4100
ACMPHS	High-Speed Analog Comparator	0x4008_5E00

Table A3.1 Peripheral base address (2 of 2)

Name	Description	Base address
GPT160	General PWM 16-bit Timer 0	0x4008_9000
GPT161	General PWM 16-bit Timer 1	0x4008_9100
GPT162	General PWM 16-bit Timer 2	0x4008_9200
GPT163	General PWM 16-bit Timer 3	0x4008_9300
GPT_OPS	Output Phase Switching Controller	0x4008_9A00
FLCN	Flash I/O Registers, Memory Wait Cycle Control, Temperature Sensor Calibration Data	0x407E_C000

Note: Name = Peripheral name
Description = Peripheral functionality
Base address = Lowest reserved address or address used by the peripheral

3.2 Access Cycles

This section provides access cycle information for the I/O registers described in this manual.

The following information applies to Table A3.2:

- Registers are grouped by associated module.
- The number of access cycles indicates the number of cycles based on the specified reference clock.
- In the internal I/O area, reserved addresses that are not allocated to registers must not be accessed, otherwise operations cannot be guaranteed.
- The number of I/O access cycles depends on bus cycles of the internal peripheral bus, divided clock synchronization cycles, and wait cycles of each module. Divided clock synchronization cycles differ depending on the frequency ratio between ICLK and PCLK.
- When the frequency of ICLK is equal to that of PCLK, the number of divided clock synchronization cycles is always constant.
- When the frequency of ICLK is greater than that of PCLK, at least 1 PCLK cycle is added to the number of divided clock synchronization cycles.

Note: This applies to the number of cycles when access from the CPU does not conflict with the instruction fetching to the external memory or bus access from other bus master such as DTC.

Table A3.2 shows the register access cycles for non-GPT modules.

Table A3.2 Access cycles for non-GPT modules (1 of 2)

Peripherals	Address		Number of access cycles				Cycle unit	Related function
			ICLK = PCLK		ICLK > PCLK*1			
	From	To	Read	Write	Read	Write		
MPU, SRAM, BUS, DTC, ICU, CPU_DBG	0x4000_2000	0x4001_BFFF	3				ICLK	Memory Protection Unit, SRAM, Buses, Data Transfer Controller, Interrupt Controller, CPU, Flash Memory
SYSC	0x4001_E000	0x4001_E6FF	4				ICLK	Low Power Modes, Resets, Low Voltage Detection, Clock Generation Circuit, Register Write Protection
PORTn, PFS, ELC, POEG, WDT, IWDT, CAC, MSTP	0x4004_0000	0x4004_7FFF	3		2 to 3		PCLKB	I/O Ports, Event Link Controller, Port Output Enable for GPT, Watchdog Timer, Independent Watchdog Timer, Clock Frequency Accuracy Measurement Circuit, Module Stop Control

Table A3.2 Access cycles for non-GPT modules (2 of 2)

Peripherals	Address		Number of access cycles				Cycle unit	Related function
			ICLK = PCLK		ICLK > PCLK*1			
	From	To	Read	Write	Read	Write		
IICn (n = 0), IIC0WU, DOC, ADC12	0x4005_0000	0x4005_EFFF	3		2 to 3		PCLKB	I²C Bus Interface, Data Operation Circuit, 12-bit A/D Converter
SCIn (n = 0 to 2, 9*2)	0x4007_0000	0x4007_0EFF	5		2 to 3		PCLKB	Serial Communications Interface
SPIn (n = 0)*3	0x4007_2000	0x4007_2FFF	5		2 to 3		PCLKB	Serial Peripheral Interface
CRC	0x4007_4000	0x4007_4FFF	3		2 to 3		PCLKB	CRC Calculator
GPT16n (n = 0 to 3), GPT_OPS	0x4008_9000	0x4008_9FFF	See Table A3.3.				PCLKB	General PWM Timer
KINT	0x4008_0000	0x4008_2FFF	3		2 to 3		PCLKB	Key interrupt Function
AGTWn	0x4008_4000	0x4008_4FFF	3		2 to 3		PCLKB	Low Power Asynchronous General Purpose Timer
ACMPHS	0x4008_5E00	0x4008_6FFF	3		2 to 3		PCLKB	High-speed Analog Comparator
FLCN	0x407E_C000	0x407E_FFFF	7		7		ICLK	Data Flash, Temperature Sensor, Capacitive Sensing Unit 2, Flash Control

Note 1. If the number of PCLK cycles is non-integer (for example 1.5), the minimum value is without the decimal point, and the maximum value is rounded up to the decimal point. For example, 1.5 to 2.5 is 1 to 3.
Note 2. Regarding n = 0, when accessing a 16-bit register (FTDRHL, FRDRHL, FCR, FDR, LSR, and CDR), access is 2 cycles more than the value shown in Table A3.2. When accessing an 8-bit register (FTDRH, FTDRL, FRDRH, and FRDRL), the access cycles are as shown in Table A3.2.
Note 3. When accessing the 32-bit register (SPDR), access is 2 cycles more than the value in Table A3.2. When accessing an 8-bit or 16-bit register (SPDR_HA), the access cycles are as shown in Table A3.2.

Table A3.3 shows register access cycles for GPT modules.

Table A3.3 Access cycles for GPT modules

Frequency ratio between ICLK and PCLK	Number of access cycles		Cycle unit
	Read	Write	
ICLK > PCLKD = PCLKB	5 to 6	3 to 4	PCLKB
ICLK > PCLKD > PCLKB	3 to 4	2 to 3	PCLKB
PCLKD = ICLK = PCLKB	6	4	PCLKB
PCLKD = ICLK > PCLKB	2 to 3	1 to 2	PCLKB
PCLKD > ICLK = PCLKB	4	3	PCLKB
PCLKD > ICLK > PCLKB	2 to 3	1 to 2	PCLKB

Revision History

Revision 1.00 — Jun 25, 2025

Initial release

Renesas RA2T1 Group User’s Manual: Hardware

Publication Date: Rev.1.00 Jun 25, 2025

Published by: Renesas Electronics Corporation

32-Bit MCU
Renesas RA2T1 Group

Renesas Electronics Corporation

R01UH1089EJ0100