User's Manual

S3A7 Microcontroller Group

User's Manual

Renesas Synergy™ Platform

Synergy Microcontrollers
S3 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.1.40 Oct 2018

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 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

General Precautions

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 describes the functions and electrical characteristics of the Renesas Synergy™ Microcontroller.

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 Synergy 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 for the Renesas Synergy Microcontroller. Before using any of these documents, visit renesassynergy.com/docs for the most up-to-date version of the document.

Component	Document type	Description
Microcontrollers	Datasheet	Features, overview, and electrical characteristics of the MCU
	User's Manual: Microcontrollers	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	Datasheet	Functional descriptions and specific performance data for software modules that are included in Renesas Synergy Software Package (SSP)
	User's Manual: Software	API reference including SSP architecture 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 Kit (DK), Starter Kit (SK), Promotion Kit (PK), Target Board Kit (TB), 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.
1Fh	Hexadecimal number. For example, the hexadecimal equivalent of the number 31 is described 1Fh. In some cases, a hexadecimal number is shown with the prefix 0x, based on C/C++ formatting.
1234	Decimal number. Decimal numbers are generally shown without a suffix.

5. Typographic Notation

The following typographic notation is used throughout this manual:

Example	Description
ICU.NMICR.NMIMD	Periods separate a function module symbol (ICU), register symbol (NMICR), and bit field symbol (NMIMD)
ICU.NMICR	A period separates a function module symbol (ICU) and register symbol (NMICR)
NMICR.NMIMD	A period separates a register symbol (NMICR) and bit field symbol (NMIMD)
NFCLKSEL[1:0]	In a register bit name, the bit range enclosed in square brackets indicates the number of bits in the field at this location. In this example, NFCLKSEL[1:0] represents a 2-bit field at the specified location in the NMI Pin Interrupt Control Register (NMICR).

6. Unit Prefix

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

Prefix	Description
b	Bit
B	Byte. This unit prefix is generally used for memory specification of the MCU and address space.
k	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	1024 = 2^{10}. This unit prefix is used to denote 1024 (2^{10}) not 1000 (10^3) throughout this manual.

7. Special Terms

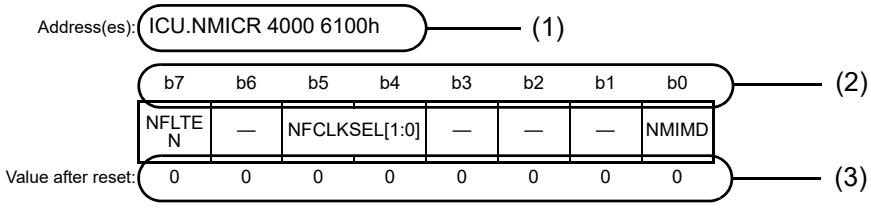
The following terms have special meanings:

Term	Description
NC	Not connected pin. NC means the pin is not connected to the MCU.
Hi-Z	High impedance

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.

X.X.X NMI Pin Interrupt Control Register (NMICR)



(4) (5) (6)

Bit	Symbol	Bit name	Description	R/W
b0	NMIMD	NMI Detection Set	0: Falling edge 1: Rising edge.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	NFCLKSEL[1:0]	NMI Digital Filter Sampling Clock Select	b5 b4 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	NFLTEN	NMI Digital Filter Enable	0: Disable the digital filter 1: Enable the digital filter.	R/W

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

Function module symbol, register symbol, and address assignment of this register are generally expressed. ICU.NMICR 4000 6100h means NMI Pin Interrupt Control Register (NMICR) of Interrupt Controller Unit (ICU) is assigned to address 4000 6100h.

(2) Bit number

This number indicates the bit number. These bits are shown in order from b31 to b0 for a 32-bit register, from b15 to b0 for a 16-bit register, and from b7 to b0 for an 8-bit register.

(3) Value after reset

This symbol or number indicates the value of each bit after a 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) Bit symbol

Bit symbol indicates the short name of the bit field. Reserved bit is expressed with a —.

(5) Bit name

Bit name indicates the full name of the bit field.

(6) R/W

The R/W column indicates access type: whether the bit field is read or write.
R/W: The bit field is read and write.
R/(W): The bit field is read and write. But writing to this bit field has some limitations. For details on the limitations, see the description or notes of respective registers.
R: The bit field is read-only. Writing to this bit field has no effect.
W: The bit field is write-only. The read value is undefined.

9. Abbreviations

Abbreviations used in this manual 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
ECC	Elliptic Curve Cryptography
ETB	Embedded Trace Buffer
ETM	Embedded Trace Macrocell
FLL	Frequency Locked Loop
FPU	Floating-Point Unit
GSM	Global System for Mobile communications
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

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.

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

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. Website and Support

Visit the following vanity URLs to learn about key elements of the Synergy Platform, download components and related documentation, and get support.

Synergy Software	renesassynergy.com/software
Synergy Software Package	renesassynergy.com/ssp
Software add-ons	renesassynergy.com/addons
Software glossary	renesassynergy.com/softwareglossary
Development tools	renesassynergy.com/tools
Synergy Hardware	renesassynergy.com/hardware
Microcontrollers	renesassynergy.com/mcus
MCU glossary	renesassynergy.com/mcuglossary
Parametric search	renesassynergy.com/parametric
Kits	renesassynergy.com/kits
Synergy Solutions Gallery	renesassynergy.com/solutionsgallery
Partner projects	renesassynergy.com/partnerprojects
Application projects	renesassynergy.com/applicationprojects
Self-service support resources:	
Documentation	renesassynergy.com/docs
Knowledgebase	renesassynergy.com/knowledgebase
Forums	renesassynergy.com/forum
Training	renesassynergy.com/training
Videos	renesassynergy.com/videos
Chat and web ticket	renesassynergy.com/support

12. Feedback on the Product

If you have any comments or suggestions about this product, go to renesassynergy.com/support.

13. Feedback on Content

If you have any comments on the document such as general suggestions for improvements, go to renesassynergy.com/support, and provide:
- The title of the Renesas Synergy document
- The document number
- If applicable, the page number(s) to which your comments refer
- A detailed explanation of your comments.

Contents

Features 48

1. Overview 49
   1.1 Function Outline 49
   1.2 Block Diagram 55
   1.3 Part Numbering 56
   1.4 Function Comparison 57
   1.5 Pin Functions 58
   1.6 Pin Assignments 62
   1.7 Pin Lists 69
2. CPU 74
   2.1 Overview 74
      2.1.1 CPU 74
      2.1.2 Debug 74
      2.1.3 Operating Frequency 75
   2.2 MCU Implementation Options 76
   2.3 Trace Interface 77
   2.4 JTAG/SWD Interface 77
   2.5 Debug Mode 77
      2.5.1 Debug Mode Definition 77
      2.5.2 Debug Mode Effects 77
   2.6 Programmers Model 78
      2.6.1 Address Spaces 78
      2.6.2 Cortex-M4 Peripheral Address Map 79
      2.6.3 CoreSight ROM Table 79
      2.6.4 DBGREG Module 80
      2.6.5 OCDREG Module 82
   2.7 CoreSight ATB Funnel 85
   2.8 Flash Patch and Break Unit 85
   2.9 SysTick System Timer 85
   2.10 CoreSight Time Stamp Generator 85
   2.11 OCD Emulator Connection 86
      2.11.1 DBGEN 86
      2.11.2 Unlock ID Code 86
      2.11.3 Restrictions on Connecting an OCD Emulator 86
   2.12 References 88
3. Operating Modes 89
   3.1 Overview 89
   3.2 Operating Mode Details 89
      3.2.1 Single-Chip Mode 89

3.2.2 SCI Boot Mode ... 89
3.2.3 USB Boot Mode ... 89
3.3 Operating Mode Transitions ... 89
3.3.1 Operating Mode Transitions as Determined by the Mode-Setting Pin ... 89
4. Address Space ... 90
4.1 Address Space ... 90
4.2 External Address Space ... 91
5. Memory Mirror Function (MMF) ... 92
5.1 Overview ... 92
5.2 Register Descriptions ... 92
5.2.1 MemMirror Special Function Register (MMSFR) ... 92
5.2.2 MemMirror Enable Register (MMEN) ... 93
5.3 Operation ... 94
5.3.1 MMF ... 94
5.3.2 Setting Example ... 96
6. Resets ... 97
6.1 Overview ... 97
6.2 Register Descriptions ... 101
6.2.1 Reset Status Register 0 (RSTSR0) ... 101
6.2.2 Reset Status Register 1 (RSTSR1) ... 102
6.2.3 Reset Status Register 2 (RSTSR2) ... 104
6.3 Operation ... 105
6.3.1 RES Pin Reset ... 105
6.3.2 Power-On Reset ... 105
6.3.3 Voltage Monitor Reset ... 106
6.3.4 Independent Watchdog Timer Reset ... 107
6.3.5 Watchdog Timer Reset ... 107
6.3.6 Software Reset ... 107
6.3.7 Determination of Cold/Warm Start ... 107
6.3.8 Determination of Reset Generation Source ... 108
7. Option-Setting Memory ... 109
7.1 Overview ... 109
7.2 Register Descriptions ... 110
7.2.1 Option Function Select Register 0 (OFS0) ... 110
7.2.2 Option Function Select Register 1 (OFS1) ... 113
7.2.3 MPU Registers ... 114
7.2.4 Access Window Setting Control Register (AWSC) ... 114
7.2.5 Access Window Setting Register (AWS) ... 115
7.2.6 OCD/Serial Programmer ID Setting Register (OSIS) ... 116
7.3 Setting Option-Setting Memory ... 117
7.3.1 Allocation of Data in Option-Setting Memory ... 117

7.3.2 Setting Data for Programming Option-Setting Memory 117
7.4 Usage Note 117
7.4.1 Data for Programming Reserved Areas and Reserved Bits in the Option-Setting Memory 117
8. Low Voltage Detection (LVD) 118
8.1 Overview 118
8.2 Register Descriptions 120
8.2.1 Voltage Monitor 1 Circuit Control Register 1 (LVD1CR1) 120
8.2.2 Voltage Monitor 1 Circuit Status Register (LVD1SR) 121
8.2.3 Voltage Monitor 2 Circuit Control Register 1 (LVD2CR1) 121
8.2.4 Voltage Monitor 2 Circuit Status Register (LVD2SR) 122
8.2.5 Voltage Monitor Circuit Control Register (LVCMPCR) 122
8.2.6 Voltage Detection Level Select Register (LVDLVLR) 123
8.2.7 Voltage Monitor 1 Circuit Control Register 0 (LVD1CR0) 124
8.2.8 Voltage Monitor 2 Circuit Control Register 0 (LVD2CR0) 124
8.3 VCC Input Voltage Monitor 125
8.3.1 Monitoring V_{det0} 125
8.3.2 Monitoring V_{det1} 125
8.3.3 Monitoring Vdet2 125
8.4 Reset from Voltage Monitor 0 126
8.5 Interrupt and Reset from Voltage Monitor 1 126
8.6 Interrupt and Reset from Voltage Monitor 2 128
8.7 Event Link Output 130
8.7.1 Interrupt Handling and Event Linking 130
9. Clock Generation Circuit 132
9.1 Overview 132
9.2 Register Descriptions 135
9.2.1 System Clock Division Control Register (SCKDIVCR) 135
9.2.2 System Clock Source Control Register (SCKSCR) 138
9.2.3 PLL Clock Control Register 2 (PLLCCR2) 139
9.2.4 PLL Control Register (PLLCR) 140
9.2.5 External Bus Clock Control Register (BCKCR) 141
9.2.6 Memory Wait Cycle Control Register (MEMWAIT) 141
9.2.7 Main Clock Oscillator Control Register (MOSCCR) 144
9.2.8 Sub-Clock Oscillator Control Register (SOSCCR) 145
9.2.9 Low-Speed On-Chip Oscillator Control Register (LOCOCR) 146
9.2.10 High-Speed On-Chip Oscillator Control Register (HOCOCR) 147
9.2.11 Middle-Speed On-Chip Oscillator Control Register (MOCOCR) 148
9.2.12 Oscillation Stabilization Flag Register (OSCSF) 148
9.2.13 Oscillation Stop Detection Control Register (OSTDCR) 150
9.2.14 Oscillation Stop Detection Status Register (OSTDSR) 151

9.2.15 Main Clock Oscillator Wait Control Register (MOSCWTCR) 152
9.2.16 High-Speed On-Chip Oscillator Wait Control Register (HOCOWTCR) 153
9.2.17 Main Clock Oscillator Mode Oscillation Control Register (MOMCR) 154
9.2.18 Sub-Clock Oscillator Mode Control Register (SOMCR) 154
9.2.19 Segment LCD Source Clock Control Register (SLCDSCKCR) 155
9.2.20 Clock Out Control Register (CKOCR) 156
9.2.21 External Bus Clock Output Control Register (EBCKOCR) 157
9.2.22 LOCO User Trimming Control Register (LOCOUTCR) 157
9.2.23 MOCO User Trimming Control Register (MOCOUTCR) 158
9.2.24 HOCO User Trimming Control Register (HOCOUTCR) 158
9.2.25 Trace Clock Control Register (TRCKCR) 159
9.3 Main Clock Oscillator 159
9.3.1 Connecting Crystal Resonator 159
9.3.2 External Clock Input 160
9.3.3 Notes on External Clock Input 160
9.4 Sub-Clock Oscillator 160
9.4.1 Connecting a 32.768-kHz Crystal Resonator 160
9.5 Oscillation Stop Detection Function 161
9.5.1 Oscillation Stop Detection and Operation after Detection 161
9.5.2 Oscillation Stop Detection Interrupts 162
9.6 PLL Circuit 163
9.7 Internal Clock 163
9.7.1 System Clock (ICLK) 163
9.7.2 Peripheral Module Clock (PCLKA, PCLKB, PCLKC, PCLKD) 164
9.7.3 Flash Interface Clock (FCLK) 164
9.7.4 External Bus Clock (BCLK) 164
9.7.5 USB Clock (UCLK) 164
9.7.6 CAN Clock (CANMCLK) 164
9.7.7 CAC Clock (CACCLK) 165
9.7.8 RTC-Dedicated Clock (RTCSCLK, RTCLCLK) 165
9.7.9 IWDT-Dedicated Clock (IWDTCLK) 165
9.7.10 AGT-Dedicated Clock (AGTSCLK, AGTLCLK) 165
9.7.11 SysTick Timer-Dedicated Clock (SYSTICCLK) 165
9.7.12 Segment LCDC Source Clock (LCDSRCCLK) 165
9.7.13 Clock/Buzzer Output Clock (CLKOUT) 165
9.7.14 JTAG Clock (JTAGTCK) 165
9.8 Usage Notes 166
9.8.1 Notes on Clock Generation Circuit 166
9.8.2 Notes on Resonator 166
9.8.3 Notes on Board Design 166
9.8.4 Notes on Resonator Connect Pin 166

10. Clock Frequency Accuracy Measurement Circuit (CAC) 167
10.1 Overview 167
10.2 Register Descriptions 168
10.2.1 CAC Control Register 0 (CACR0) 168
10.2.2 CAC Control Register 1 (CACR1) 169
10.2.3 CAC Control Register 2 (CACR2) 170
10.2.4 CAC Interrupt Control Register (CAICR) 171
10.2.5 CAC Status Register (CASTR) 172
10.2.6 CAC Upper-Limit Value Setting Register (CAULVR) 173
10.2.7 CAC Lower-Limit Value Setting Register (CALLVR) 173
10.2.8 CAC Counter Buffer Register (CACNTBR) 173
10.3 Operation 174
10.3.1 Measuring Clock Frequency 174
10.3.2 Digital Filtering of Signals on CACREF Pin 175
10.4 Interrupt Requests 175
10.5 Usage Note 175
10.5.1 Module Stop Function Setting 175
11. Low Power Modes 176
11.1 Overview 176
11.2 Register Descriptions 180
11.2.1 Standby Control Register (SBYCR) 180
11.2.2 Module Stop Control Register A (MSTPCRA) 181
11.2.3 Module Stop Control Register B (MSTPCRB) 181
11.2.4 Module Stop Control Register C (MSTPCRC) 183
11.2.5 Module Stop Control Register D (MSTPCRD) 184
11.2.6 Operating Power Control Register (OPCCR) 185
11.2.7 Sub Operating Power Control Register (SOPCCR) 186
11.2.8 Snooze Control Register (SNZCR) 187
11.2.9 Snooze End Control Register (SNZEDCR) 188
11.2.10 Snooze Request Control Register (SNZREQCR) 189
11.2.11 Flash Operation Control Register (FLSTOP) 191
11.2.12 Power Save Memory Control Register (PSMCR) 191
11.2.13 System Control OCD Control Register (SYOCDCR) 192
11.3 Reducing Power Consumption by Switching Clock Signals 192
11.4 Module-Stop Function 192
11.5 Function for Lower Operating Power Consumption 193
11.5.1 Setting Operating Power Control Mode 193
11.5.2 Operating range 195
11.6 Sleep Mode 198
11.6.1 Transition to Sleep Mode 198
11.6.2 Canceling Sleep Mode 198

11.7 Software Standby Mode 199
11.7.1 Transition to Software Standby Mode 199
11.7.2 Canceling Software Standby Mode 199
11.7.3 Software Standby Mode Operation Example 200
11.8 Snooze Mode 201
11.8.1 Transition to Snooze Mode 201
11.8.2 Canceling Snooze Mode 202
11.8.3 Return to Software Standby Mode 202
11.8.4 Snooze Operation Example 204
11.9 Usage Notes 207
11.9.1 Register Access 207
11.9.2 I/O Port States 208
11.9.3 Module-Stop State of DMAC and DTC 208
11.9.4 Internal Interrupt Sources 209
11.9.5 Transition to Low Power Modes 209
11.9.6 Timing of WFI Instruction 209
11.9.7 Writing WDT/IWDT Registers by DMAC or DTC in Sleep Mode or Snooze Mode 209
11.9.8 Oscillators in Snooze Mode 209
11.9.9 Snooze Mode Entry by RXD0 Falling Edge 209
11.9.10 Using SCI0 in Snooze Mode 209
11.9.11 Conditions of A/D Conversion Start in Snooze Mode 209
11.9.12 Conditions of CTSU in Snooze Mode 209
11.9.13 ELC Event in Snooze Mode 210
11.9.14 Module-Stop Function for ADC140 210
11.9.15 Module-Stop Function for an Unused Circuit 210
12. Battery Backup Function 211
12.1 Overview 211
12.1.1 Features of Battery Backup Function 211
12.1.2 Battery Power Supply Switch 211
12.1.3 VBATT Pin Low Voltage Detection 211
12.1.4 VBATT_R Low Voltage Detection 211
12.1.5 Backup Registers 211
12.1.6 VBATT Wakeup Control Function 211
12.1.7 Time Capture Pin Detection 212
12.2 Register Descriptions 213
12.2.1 VBATT Control Register 1 (VBTCR1) 213
12.2.2 VBATT Control Register 2 (VBTCR2) 214
12.2.3 VBATT Status Register (VBTSR) 214
12.2.4 VBATT Comparator Control register (VBTCMPCR) 215
12.2.5 VBATT Pin Low Voltage Detect Interrupt Control Register (VBTLVDICR) 216
12.2.6 VBATT Backup Register (VBTBKR[n]) (n = 0 to 511) 216

12.2.7 VBATT Wakeup Control Register (VBTWCTLR) 217
12.2.8 VBATT Wakeup I/O 0 Output Trigger Select Register (VBTWCH0OTSR) 218
12.2.9 VBATT Wakeup I/O 1 Output Trigger Select Register (VBTWCH1OTSR) 219
12.2.10 VBATT Wakeup I/O 2 Output Trigger Select Register (VBTWCH2OTSR) 220
12.2.11 VBATT Input Control Register (VBTICTLR) 221
12.2.12 VBATT Output Control Register (VBTOCTLR) 222
12.2.13 VBATT Wakeup Trigger Source Enable Register (VBTWTER) 223
12.2.14 VBATT Wakeup Trigger Source Edge Register (VBTWEGR) 223
12.2.15 VBATT Wakeup Trigger Source Flag Register (VBTWFR) 224
12.3 Operation 225
12.3.1 Battery Backup Function 225
12.3.2 VBATT Battery Power Supply Switch Usage 227
12.3.3 VBATT Pin Low Voltage Detection Procedures 227
12.3.4 VBATT Backup Register Usage 228
12.3.5 VBATT Wakeup Control Function Usage 228
12.4 Usage Note 230
13. Register Write Protection 231
13.1 Overview 231
13.2 Register Descriptions 231
13.2.1 Protect Register (PRCR) 231
14. Interrupt Controller Unit (ICU) 232
14.1 Overview 232
14.2 Register Descriptions 234
14.2.1 IRQ Control Register i (IRQCRi) (i = 0 to 15) 234
14.2.2 Non-Maskable Interrupt Status Register (NMISR) 235
14.2.3 Non-Maskable Interrupt Enable Register (NMIER) 238
14.2.4 Non-Maskable Interrupt Status Clear Register (NMICLR) 239
14.2.5 NMI Pin Interrupt Control Register (NMICR) 241
14.2.6 ICU Event Link Setting Register n (IELSRn) (n= 0 to 63) 242
14.2.7 DMAC Event Link Setting Register n (DELSRn) 243
14.2.8 SYS Event Link Setting Register (SELSR0) 243
14.2.9 Wake Up Interrupt Enable Register (WUPEN) 244
14.3 Vector Table 246
14.3.1 Interrupt Vector Table 246
14.3.2 Event Number 248
14.4 Interrupt Operation 254
14.4.1 Detecting Interrupts 254
14.4.2 Selecting Interrupt Request Destinations 255
14.4.3 Digital Filter 256
14.4.4 External Pin Interrupts 257
14.5 Non-maskable Interrupt Operation 257

14.6 Return from Low Power Mode 258
14.6.1 Return from Sleep mode 258
14.6.2 Return from Software Standby mode 258
14.6.3 Return from Snooze mode 258
14.7 Using the WFI instruction with Non-maskable Interrupts 259
14.8 Reference 259
15. Buses 260
15.1 Overview 260
15.2 Description of Buses 261
15.2.1 Main Buses 261
15.2.2 Slave Interface 261
15.2.3 External Bus 262
15.2.4 Parallel Operation 263
15.2.5 Bus Settings 263
15.2.6 Restrictions 264
15.3 Register Descriptions 264
15.3.1 CSn Control Register (CSnCR) (n = 0 to 3) 264
15.3.2 CSn Recovery Cycle Register (CSnREC) (n = 0 to 3) 265
15.3.3 CS Recovery Cycle Insertion Enable Register (CSRECEN) 266
15.3.4 CSn Mode Register (CSnMOD) (n = 0 to 3) 267
15.3.5 CSn Wait Control Register 1 (CSnWCR1) (n = 0 to 3) 268
15.3.6 CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 3) 270
15.3.7 Master Bus Control Register (BUSMCNT<master>) 272
15.3.8 Slave Bus Control Register (BUSSCNT<slave>) 273
15.3.9 Bus Error Address Register (BUSnERRADD) (n = 1 to 4) 274
15.3.10 Bus Error Status Register (BUSnERRSTAT) (n = 1 to 4) 275
15.4 Endianness and Data Alignment 276
15.4.1 Data Alignment Control for the CS Areas 276
15.5 Operation of CS Area Controller 278
15.5.1 Separate Bus 278
15.5.2 External Wait Function 287
15.5.3 Insertion of Recovery Cycles 289
15.5.4 No Access State 291
15.5.5 Write Buffer Function (External Bus) 291
15.5.6 Constraints 292
15.6 Bus Error Monitoring Section 292
15.6.1 Error Type that Occurs by Bus 292
15.6.2 Operation when a Bus Error Occurs 293
15.6.3 Conditions Leading to Illegal Address Access Errors 293
15.6.4 Timeout 294
15.7 Notes on using Flash Cache 294

15.8 References 294
16. Memory Protection Unit (MPU) 295
16.1 Overview 295
16.2 CPU Stack Pointer Monitor 295
16.2.1 Protection of Registers 297
16.2.2 Overflow/Underflow Error 297
16.2.3 Register Descriptions 297
16.3 Arm MPU 301
16.4 Bus Master MPU 302
16.4.1 Register Descriptions 303
16.4.2 Functions 307
16.5 Bus Slave MPU 309
16.5.1 Register Descriptions 310
16.5.2 Functions 319
16.6 Security MPU 319
16.6.1 Register Descriptions (Option-Setting Memory) 320
16.6.2 Memory Protection 322
16.6.3 Usage Notes 323
16.7 References 323
17. DMA Controller (DMAC) 324
17.1 Overview 324
17.2 Register Descriptions 326
17.2.1 DMA Source Address Register (DMSAR) 326
17.2.2 DMA Destination Address Register (DMDAR) 326
17.2.3 DMA Transfer Count Register (DMCRA) 327
17.2.4 DMA Block Transfer Count Register (DMCRB) 328
17.2.5 DMA Transfer Mode Register (DMTMD) 328
17.2.6 DMA Interrupt Setting Register (DMINT) 329
17.2.7 DMA Address Mode Register (DMAMD) 330
17.2.8 DMA Offset Register (DMOFR) 332
17.2.9 DMA Transfer Enable Register (DMCNT) 332
17.2.10 DMA Software Start Register (DMREQ) 333
17.2.11 DMA Status Register (DMSTS) 333
17.2.12 DMACA Module Activation Register (DMAST) 335
17.3 Operation 335
17.3.1 Transfer Mode 335
17.3.2 Extended Repeat Area Function 338
17.3.3 Address Update Function Using Offset 339
17.3.4 Activation Sources 343
17.3.5 Operation Timing 343
17.3.6 Execution Cycles of DMAC 344

17.3.7 Activating DMAC 345
17.3.8 Starting DMA Transfer 346
17.3.9 Registers during DMA Transfer 346
17.3.10 Channel Priority 347
17.4 Ending DMA Transfer 347
17.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations 347
17.4.2 Transfer End by Repeat Size End Interrupt 347
17.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow 347
17.4.4 Precautions in the end of DMA transfer 348
17.5 Interrupts 349
17.6 Event Link 350
17.7 Low Power Consumption Function 350
17.8 Usage Notes 351
17.8.1 DMA Transfer to External Devices 351
17.8.2 Access to Registers during DMA Transfer 351
17.8.3 DMA Transfer to Reserved Areas 351
17.8.4 Setting of DMAC Event Link Setting Register of the Interrupt Controller Unit (ICU.DELSRn) 351
17.8.5 Suspending or Restarting DMA Activation 351
18. Data Transfer Controller (DTC) 352
18.1 Overview 352
18.2 Register Descriptions 354
18.2.1 DTC Mode Register A (MRA) 354
18.2.2 DTC Mode Register B (MRB) 355
18.2.3 DTC Transfer Source Register (SAR) 356
18.2.4 DTC Transfer Destination Register (DAR) 356
18.2.5 DTC Transfer Count Register A (CRA) 357
18.2.6 DTC Transfer Count Register B (CRB) 358
18.2.7 DTC Control Register (DTCCR) 358
18.2.8 DTC Vector Base Register (DTCVBR) 359
18.2.9 DTC Module Start Register (DTCST) 359
18.2.10 DTC Status Register (DTCSTS) 360
18.3 Activation Sources 360
18.3.1 Allocating Transfer Information and DTC Vector Table 361
18.4 Operation 362
18.4.1 Transfer Information Read Skip Function 364
18.4.2 Transfer Information Write-Back Skip Function 365
18.4.3 Normal Transfer Mode 365
18.4.4 Repeat Transfer Mode 366
18.4.5 Block Transfer Mode 367
18.4.6 Chain Transfer 368
18.4.7 Operation Timing 369

18.4.8 Execution Cycles of DTC ... 371
18.4.9 DTC Bus Mastership Release Timing ... 371
18.5 DTC Setting Procedure ... 372
18.6 Examples of DTC Usage ... 373
18.6.1 Normal Transfer ... 373
18.6.2 Chain Transfer ... 373
18.6.3 Chain Transfer When Counter = 0 ... 375
18.7 Interrupt Source ... 376
18.8 Event Link ... 376
18.9 Snooze Control Interface ... 376
18.10 Module-Stop Function ... 377
18.11 Usage Notes ... 377
18.11.1 Transfer Information Start Address ... 377
19. Event Link Controller (ELC) ... 378
19.1 Overview ... 378
19.2 Register Descriptions ... 379
19.2.1 Event Link Controller Register (ELCR) ... 379
19.2.2 Event Link Software Event Generation Register n (ELSEGRn) (n = 0, 1) ... 379
19.2.3 Event Link Setting Register n (ELSRn) (n = 0 to 9, 12 to 18) ... 380
19.3 Operation ... 385
19.3.1 Relation between Interrupt Handling and Event Linking ... 385
19.3.2 Event Linkage ... 385
19.3.3 Example of Procedure for Linking Events ... 386
19.4 Usage Notes ... 386
19.4.1 Linking DMAC/DTC Transfer End Signals as Events ... 386
19.4.2 Setting Clocks ... 386
19.4.3 Module Stop Function Setting ... 386
19.4.4 ELC Delay Time ... 386
20. I/O Ports ... 387
20.1 Overview ... 387
20.2 Register Descriptions ... 389
20.2.1 Port Control Register 1 (PCNTR1/PODR/PDR) ... 389
20.2.2 Port Control Register 2 (PCNTR2/EIDR/PIDR) ... 390
20.2.3 Port Control Register 3 (PCNTR3/PORR/POSR) ... 391
20.2.4 Port Control Register 4 (PCNTR4/EORR/EOSR) ... 392
20.2.5 Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) (m = 0 to 9; n = 00 to 15) ... 393
20.2.6 Write-Protect Register (PWPR) ... 395
20.3 Operation ... 395
20.3.1 General I/O Ports ... 395
20.3.2 Port Function Select ... 396
20.3.3 Port Group Function for ELC ... 396

20.4 Handling of Unused Pins 398
20.5 Usage Notes 399
20.5.1 Procedure for Specifying the Pin Functions 399
20.5.2 Procedure for Using Port Group Input 399
20.5.3 Port Output Data Register (PODR) Summary 399
20.5.4 Notes on Using of Analog Functions 399
20.5.5 I/O Buffer Specification 399
20.6 Peripheral Select Settings for each product 400
21. Key Interrupt Function (KINT) 412
21.1 Overview 412
21.2 Register Descriptions 414
21.2.1 Key Return Control Register (KRCTL) 414
21.2.2 Key Return Flag Register (KRF) 414
21.2.3 Key Return Mode Register (KRM) 415
21.3 Operation 415
21.3.1 When Not Using Key Interrupt Flag (KRMD = 0) 415
21.3.2 When Using Key Interrupt Flag (KRMD = 1) 416
21.4 Usage Note 417
22. Port Output Enable for GPT (POEG) 418
22.1 Overview 418
22.2 Register Descriptions 420
22.2.1 POEG Group n Setting Register (POEGGn) (n = A to D) 420
22.3 Output-Disable Control Operation 421
22.3.1 Pin Input Level Detection Operation 421
22.3.2 Output-Disable Request from GPT 422
22.3.3 Comparator Interrupt Detection 422
22.3.4 Output Disable Control on Detection of Stopped Oscillation 422
22.3.5 Output Disable Control Using Registers 422
22.3.6 Release from Output Disable 422
22.4 Interrupt Sources 423
22.5 External Trigger Output to GPT 424
22.6 Usage Notes 424
22.6.1 Transition to Software Standby Mode 424
22.6.2 Specifying Pins Corresponding to GPT 424
23. General PWM Timer (GPT) 425
23.1 Overview 425
23.2 Register Descriptions 429
23.2.1 General PWM Timer Write-Protection Register (GTWP) 430
23.2.2 General PWM Timer Software Start Register (GTSTR) 430
23.2.3 General PWM Timer Software Stop Register (GTSTP) 431
23.2.4 General PWM Timer Software Clear Register (GTCLR) 431

23.2.5 General PWM Timer Start Source Select Register (GTSSR) ... 432
23.2.6 General PWM Timer Stop Source Select Register (GTPSR) ... 435
23.2.7 General PWM Timer Clear Source Select Register (GTCSR) ... 438
23.2.8 General PWM Timer Up Count Source Select Register (GTUPSR) ... 441
23.2.9 General PWM Timer Down Count Source Select Register (GTDNSR) ... 444
23.2.10 General PWM Timer Input Capture Source Select Register A(GTICASR) ... 447
23.2.11 General PWM Timer Input Capture Source Select Register B(GTICBSR) ... 450
23.2.12 General PWM Timer Control Register (GTCR) ... 453
23.2.13 General PWM Timer Count Direction and Duty Setting Register (GTUDDTYC) ... 454
23.2.14 General PWM Timer I/O Control Register (GTIOR) ... 456
23.2.15 General PWM Timer Interrupt Output Setting Register (GTINTAD) ... 460
23.2.16 General PWM Timer Status Register (GTST) ... 461
23.2.17 General PWM Timer Buffer Enable Register (GTBER) ... 464
23.2.18 General PWM Timer Counter (GTCNT) ... 466
23.2.19 General PWM Timer Compare Capture Register n (GTCCRn) (n = A to F) ... 466
23.2.20 General PWM Timer Cycle Setting Register (GTPR) ... 466
23.2.21 General PWM Timer Cycle Setting Buffer Register (GTPBR) ... 467
23.2.22 General PWM Timer Dead Time Control Register (GTDTCR) ... 467
23.2.23 General PWM Timer Dead Time Value Register U (GTDVU) ... 468
23.2.24 Output Phase Switching Control Register (OPSCR) ... 468
23.3 Operation ... 471
23.3.1 Basic Operation ... 471
23.3.2 Buffer Operation ... 481
23.3.3 PWM Output Operating Mode ... 489
23.3.4 Automatic Dead Time Setting Function ... 499
23.3.5 Count Direction Changing Function ... 504
23.3.6 Function of Output Duty 0% and 100% ... 505
23.3.7 Hardware Count Start/Count Stop and Clear Operation ... 507
23.3.8 Synchronized Operation ... 512
23.3.9 PWM Output Operation Examples ... 516
23.3.10 Phase Counting Function ... 522
23.3.11 Output Phase Switching (GPT_OPS) ... 529
23.4 Interrupt Sources ... 537
23.4.1 Interrupt Sources ... 537
23.4.2 DMAC/DTC Activation ... 540
23.5 Operations Linked by ELC ... 541
23.5.1 Event Signal Output to ELC ... 541
23.5.2 Event Signal Inputs from ELC ... 541
23.6 Noise Filter Function ... 541
23.7 Protection Function ... 542
23.7.1 Write-Protection for Registers ... 542

23.7.2 Disabling of Buffer Operation 542
23.7.3 GTIOC Pin Output Negate Control 543
23.8 Initialization Method of Output Pins 544
23.8.1 Pin Settings after Reset 544
23.8.2 Pin Initialization Due to Error during Operation 544
23.9 Usage Notes 545
23.9.1 Module Stop Function Setting 545
23.9.2 Settings of GTCCRn during Compare Match Operation (n = A to F) 545
23.9.3 Setting the Range of GTCNT Counter 545
23.9.4 GTCNT Counter Start/Stop 545
23.9.5 Priority Order of each Event 546
24. Asynchronous General Purpose Timer (AGT) 547
24.1 Overview 547
24.2 Register Descriptions 549
24.2.1 AGT Counter Register (AGT) 549
24.2.2 AGT Compare Match A Register (AGTCMA) 549
24.2.3 AGT Compare Match B Register (AGTCMB) 550
24.2.4 AGT Control Register (AGTCR) 550
24.2.5 AGT Mode Register 1 (AGTMR1) 552
24.2.6 AGT Mode Register 2 (AGTMR2) 553
24.2.7 AGT I/O Control Register (AGTIOC) 554
24.2.8 AGT Event Pin Select Register (AGTISR) 555
24.2.9 AGT Compare Match Function Select Register (AGTCMSR) 555
24.2.10 AGT Pin Select Register (AGTIOSEL) 556
24.3 Operation 557
24.3.1 Reload Register and Counter Rewrite Operation 557
24.3.2 Reload Register and Compare Register A/B Rewrite Operation 559
24.3.3 Timer Mode 560
24.3.4 Pulse Output Mode 561
24.3.5 Event Counter Mode 562
24.3.6 Pulse Width Measurement Mode 564
24.3.7 Pulse Period Measurement Mode 565
24.3.8 Compare Match function 566
24.3.9 Output Settings for each Mode 567
24.3.10 Standby Mode 568
24.3.11 Interrupt Sources 569
24.3.12 Event Signal Output to ELC 569
24.4 Usage Notes 569
24.4.1 Count Operation Start and Stop Control 569
24.4.2 Access to Counter Register 570
24.4.3 When Changing Mode 570

24.4.4 Digital Filter ... 570
24.4.5 How to Calculate Event Number, Pulse Width, and Pulse Period ... 570
24.4.6 When Count is Forcibly Stopped by TSTOP bit ... 570
24.4.7 When Selecting AGT0 Underflow as the Count Source ... 571
24.4.8 Reset of I/O Register ... 571
24.4.9 When Selecting PCLKB/8 as the Count Source ... 571
24.4.10 When Selecting AGTLCLK or AGTSCLK as the Count Source ... 571
24.4.11 When Count Source Clock Frequency is over 32 kHz ... 571
25. Realtime Clock (RTC) ... 572
25.1 Overview ... 572
25.2 Register Descriptions ... 574
25.2.1 64-Hz Counter ... 574
25.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0) ... 574
25.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1) ... 575
25.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2) ... 576
25.2.5 Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3) ... 577
25.2.6 Day Counter (RDAYCNT) ... 577
25.2.7 Month Counter (RMONCNT) ... 578
25.2.8 Year Counter (RYRCNT) ... 578
25.2.9 Second Alarm Register (RSECAR)/Binary Counter 0 Alarm Register (BCNT0AR) ... 579
25.2.10 Minute Alarm Register (RMINAR)/Binary Counter 1 Alarm Register (BCNT1AR) ... 580
25.2.11 Hour Alarm Register (RHRAR)/Binary Counter 2 Alarm Register (BCNT2AR) ... 581
25.2.12 Day-of-Week Alarm Register (RWKAR)/Binary Counter 3 Alarm Register (BCNT3AR) ... 582
25.2.13 Date Alarm Register (RDAYAR)/Binary Counter 0 Alarm Enable Register (BCNT0AER) ... 583
25.2.14 Month Alarm Register (RMONAR)/Binary Counter 1 Alarm Enable Register (BCNT1AER) ... 584
25.2.15 Year Alarm Register (RYRAR)/Binary Counter 2 Alarm Enable Register (BCNT2AER) ... 585
25.2.16 Year Alarm Enable Register (RYRAREN)/Binary Counter 3 Alarm Enable Register (BCNT3AER) ... 586
25.2.17 RTC Control Register 1 (RCR1) ... 587
25.2.18 RTC Control Register 2 (RCR2) ... 588
25.2.19 RTC Control Register 4 (RCR4) ... 591
25.2.20 Frequency Register (RFRH/RFRL) ... 592
25.2.21 Time Error Adjustment Register (RADJ) ... 593
25.2.22 Time Capture Control Register y (RTCCRy) (y = 0 to 2) ... 593
25.2.23 Second Capture Register y (RSECCPy) (y = 0 to 2)/BCNT0 Capture Register y (BCNT0CPy) (y = 0 to 2) ... 595
25.2.24 Minute Capture Register y (RMINCPy) (y = 0 to 2)/BCNT1 Capture Register y (BCNT1CPy) (y = 0 to 2) ... 595

25.2.25 Hour Capture Register y (RHRCPy) (y = 0 to 2)/BCNT2 Capture Register y (BCNT2CPy) (y = 0 to 2) ... 596
25.2.26 Date Capture Register y (RDAYCPy) (y = 0 to 2)/BCNT3 Capture Register y (BCNT3CPy) (y = 0 to 2) ... 597
25.2.27 Month Capture Register y (RMONCPy) (y = 0 to 2) ... 598
25.3 Operation ... 599
25.3.1 Outline of Initial Settings of Registers after Power On ... 599
25.3.2 Clock and Count Mode Setting Procedure ... 599
25.3.3 Setting the Time ... 600
25.3.4 30-Second Adjustment ... 601
25.3.5 Reading 64-Hz Counter and Time ... 602
25.3.6 Alarm Function ... 603
25.3.7 Procedure for Disabling Alarm Interrupt ... 604
25.3.8 Time Error Adjustment Function ... 604
25.4 Interrupt Sources ... 607
25.5 Event Link Output ... 608
25.5.1 Interrupt Handling and Event Linking ... 608
25.6 Usage Notes ... 609
25.6.1 Register Writing during Counting ... 609
25.6.2 Use of Periodic Interrupts ... 609
25.6.3 RTCOUT (1-Hz/64-Hz) Clock Output ... 610
25.6.4 Transitions to Low Power Consumption Modes after Setting Registers ... 610
25.6.5 Notes When Writing to and Reading from Registers ... 610
25.6.6 Changing the Count Mode ... 610
25.6.7 Initialization Procedure when the Realtime Clock is not to be Used ... 611
26. Watchdog Timer (WDT) ... 612
26.1 Overview ... 612
26.2 Register Descriptions ... 613
26.2.1 WDT Refresh Register (WDTRR) ... 613
26.2.2 WDT Control Register (WDTCR) ... 613
26.2.3 WDT Status Register (WDTSR) ... 616
26.2.4 WDT Reset Control Register (WDTRCR) ... 617
26.2.5 WDT Count Stop Control Register (WDTCSTPR) ... 617
26.2.6 Option Function Select Register 0 (OFS0) ... 617
26.3 Operation ... 618
26.3.1 Count Operation in Each Start Mode ... 618
26.3.2 Control over Writing to the WDTCR, WDTRCR, and WDTCSTPR Registers ... 621
26.3.3 Refresh Operation ... 622
26.3.4 Reset Output ... 623
26.3.5 Interrupt Sources ... 623
26.3.6 Reading Down-Counter Value ... 623
26.3.7 Correspondence between Option Function Select Register 0 (OFS0)

and WDT Registers 624
26.4 Link Operation by ELC 624
26.5 Usage Notes 624
26.5.1 ICU Event Link Setting Register n (IELSRn) setting 624
27. Independent Watchdog Timer (IWDT) 625
27.1 Overview 625
27.2 Register Descriptions 626
27.2.1 IWDT Refresh Register (IWDTRR) 626
27.2.2 IWDT Status Register (IWDTSR) 627
27.2.3 Option Function Select Register 0 (OFS0) 628
27.3 Operation 630
27.3.1 Auto-Start Mode 630
27.3.2 Refresh Operation 631
27.3.3 Status Flags 632
27.3.4 Reset Output 632
27.3.5 Interrupt Sources 633
27.3.6 Reading the Down-counter Value 633
27.4 Link Operation by ELC 633
27.5 Usage Notes 633
27.5.1 Refresh Operations 633
27.5.2 Clock Division Ratio Setting 633
28. USB 2.0 Full-Speed Module (USBFS) 634
28.1 Overview 634
28.2 Register Descriptions 636
28.2.1 System Configuration Control Register (SYSCFG) 636
28.2.2 System Configuration Status Register 0 (SYSSTS0) 637
28.2.3 Device State Control Register 0 (DVSTCTR0) 638
28.2.4 CFIFO Port Register (CFIFO/CFIFOL)
D0FIFO Port Register (D0FIFO/D0FIFOL)
D1FIFO Port Register (D1FIFO/D1FIFOL) 640
28.2.5 CFIFO Port Select Register (CFIFOSEL)
D0FIFO Port Select Register (D0FIFOSEL)
D1FIFO Port Select Register (D1FIFOSEL) 642
28.2.6 CFIFO Port Control Register (CFIFOCTR)
D0FIFO Port Control Register (D0FIFOCTR)
D1FIFO Port Control Register (D1FIFOCTR) 645
28.2.7 Interrupt Enable Register 0 (INTENB0) 646
28.2.8 Interrupt Enable Register 1 (INTENB1) 647
28.2.9 BRDY Interrupt Enable Register (BRDYENB) 648
28.2.10 NRDY Interrupt Enable Register (NRDYENB) 649
28.2.11 BEMP Interrupt Enable Register (BEMPENB) 650
28.2.12 SOF Output Configuration Register (SOFCFG) 651
28.2.13 Interrupt Status Register 0 (INTSTS0) 651

28.2.14 Interrupt Status Register 1 (INTSTS1) ... 654
28.2.15 BRDY Interrupt Status Register (BRDYSTS) ... 656
28.2.16 NRDY Interrupt Status Register (NRDYSTS) ... 657
28.2.17 BEMP Interrupt Status Register (BEMPSTS) ... 657
28.2.18 Frame Number Register (FRMNUM) ... 658
28.2.19 USB Request Type Register (USBREQ) ... 659
28.2.20 USB Request Value Register (USBVAL) ... 660
28.2.21 USB Request Index Register (USBINDX) ... 660
28.2.22 USB Request Length Register (USBLENG) ... 661
28.2.23 DCP Configuration Register (DCPCFG) ... 662
28.2.24 DCP Maximum Packet Size Register (DCPMAXP) ... 663
28.2.25 DCP Control Register (DCPCTR) ... 664
28.2.26 Pipe Window Select Register (PIPESEL) ... 667
28.2.27 Pipe Configuration Register (PIPECFG) ... 668
28.2.28 Pipe Maximum Packet Size Register (PIPEMAXP) ... 670
28.2.29 Pipe Cycle Control Register (PIPEPERI) ... 671
28.2.30 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9) ... 672
28.2.31 PIPEn Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5) ... 678
28.2.32 PIPEn Transaction Counter Register (PIPEnTRN) (n = 1 to 5) ... 679
28.2.33 Device Address n Configuration Register (DEVADDn) (n = 0 to 5) ... 680
28.2.34 USB Module Control Register (USBMC) ... 680
28.2.35 BC Control Register 0 (USBBCCTRL0) ... 681
28.3 Operation ... 682
28.3.1 System Control ... 682
28.3.2 Interrupts ... 688
28.3.3 Interrupt Descriptions ... 691
28.3.4 Pipe Control ... 700
28.3.5 FIFO Buffer Memory ... 704
28.3.6 FIFO Buffer Clearing ... 705
28.3.7 FIFO Port Functions ... 706
28.3.8 DMA Transfers (D0FIFO and D1FIFO Ports) ... 707
28.3.9 Control Transfers Using DCP ... 707
28.3.10 Bulk Transfers (Pipes 1 to 5) ... 709
28.3.11 Interrupt Transfers (Pipes 6 to 9) ... 709
28.3.12 Isochronous Transfers (Pipes 1 and 2) ... 709
28.3.13 SOF Interpolation Function ... 716
28.3.14 Pipe Schedule ... 716
28.3.15 Battery Charging Detection Processing ... 717
28.4 Usage Notes ... 720
28.4.1 Settings for the Module-Stop State ... 720
28.4.2 Clearing the Interrupt Status Register on Exiting Software Standby Mode ... 720

28.4.3 Clearing the Interrupt Status Register after Setting Up the Port Function 720
29. Serial Communications Interface (SCI) 721
29.1 Overview 721
29.2 Register Descriptions 725
29.2.1 Receive Shift Register (RSR) 725
29.2.2 Receive Data Register (RDR) 725
29.2.3 Receive 9-bit Data Register (RDRHL) 725
29.2.4 Receive FIFO Data Register H, L, HL (FRDRH, FRDRL, FRDRHL) 726
29.2.5 Transmit Data Register (TDR) 727
29.2.6 Transmit 9-Bit Data Register (TDRHL) 727
29.2.7 Transmit FIFO Data Register H, L, HL (FTDRH, FTDRL, FTDRHL) 728
29.2.8 Transmit Shift Register (TSR) 728
29.2.9 Serial Mode Register (SMR) for Non-Smart Card Interface Mode (SCMR.SMIF = 0) 729
29.2.10 Serial Mode Register for Smart Card Interface Mode (SMR_SMCI) (SCMR.SMIF = 1) 730
29.2.11 Serial Control Register (SCR) for Non-Smart Card Interface Mode (SCMR.SMIF = 0) 732
29.2.12 Serial Control Register for Smart Card Interface Mode (SCR_SMCI) (SCMR.SMIF = 1) 734
29.2.13 Serial Status Register (SSR) for Non-Smart Card Interface and Non-FIFO Mode (SCMR.SMIF = 0 and FCR.FM = 0) 736
29.2.14 Serial Status Register for Non-Smart Card Interface and FIFO Mode (SSR_FIFO) (SCMR.SMIF = 0 and FCR.FM = 1) 738
29.2.15 Serial Status Register for Smart Card Interface Mode (SSR_SMCI) (SCMR.SMIF = 1) 741
29.2.16 Smart Card Mode Register (SCMR) 743
29.2.17 Bit Rate Register (BRR) 744
29.2.18 Modulation Duty Register (MDDR) 752
29.2.19 Serial Extended Mode Register (SEMR) 754
29.2.20 Noise Filter Setting Register (SNFR) 755
29.2.21 I²C Mode Register 1 (SIMR1) 756
29.2.22 I²C Mode Register 2 (SIMR2) 757
29.2.23 I²C Mode Register 3 (SIMR3) 758
29.2.24 I²C Status Register (SISR) 759
29.2.25 SPI Mode Register (SPMR) 760
29.2.26 FIFO Control Register (FCR) 761
29.2.27 FIFO Data Count Register (FDR) 762
29.2.28 Line Status Register (LSR) 763
29.2.29 Compare Match Data Register (CDR) 764
29.2.30 Data Compare Match Control Register (DCCR) 764
29.2.31 Serial Port Register (SPTR) 766
29.3 Operation in Asynchronous Mode 766

29.3.1 Serial Data Transfer Format 767
29.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode 769
29.3.3 Clock 770
29.3.4 Double-Speed Operation and Frequency of 6 Times the Bit Rate 770
29.3.5 CTS and RTS Functions 770
29.3.6 Address Match (Receive Data Match Detection) Function 771
29.3.7 SCI Initialization in Asynchronous Mode 774
29.3.8 Serial Data Transmission in Asynchronous Mode 776
29.3.9 Serial Data Reception in Asynchronous Mode 781
29.4 Multi-Processor Communication Function 788
29.4.1 Multi-Processor Serial Data Transmission 789
29.4.2 Multi-Processor Serial Data Reception 791
29.5 Operation in Clock Synchronous Mode 796
29.5.1 Clock 796
29.5.2 CTS and RTS Functions 797
29.5.3 SCI Initialization in Clock Synchronous Mode 798
29.5.4 Serial Data Transmission in Clock Synchronous Mode 800
29.5.5 Serial Data Reception in Clock Synchronous Mode 804
29.5.6 Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode 807
29.6 Operation in Smart Card Interface Mode 809
29.6.1 Example Connection 809
29.6.2 Data Format (Except in Block Transfer Mode) 809
29.6.3 Block Transfer Mode 810
29.6.4 Receive Data Sampling Timing and Reception Margin 810
29.6.5 Initialization of the SCI 812
29.6.6 Serial Data Transmission (Except in Block Transfer Mode) 813
29.6.7 Serial Data Reception (Except in Block Transfer Mode) 816
29.6.8 Clock Output Control 818
29.7 Operation in Simple IIC Mode 819
29.7.1 Generation of Start, Restart, and Stop Conditions 820
29.7.2 Clock Synchronization 821
29.7.3 SDA Output Delay 822
29.7.4 SCI Initialization in Simple IIC Mode 823
29.7.5 Operation in Master Transmission (Simple IIC Mode) 824
29.7.6 Master Reception in Simple IIC Mode 826
29.8 Operation in Simple SPI Mode 828
29.8.1 States of Pins in Master and Slave Modes 829
29.8.2 SS Function in Master Mode 829
29.8.3 SS Function in Slave Mode 829
29.8.4 Relationship between Clock and Transmit/Receive Data 829

29.8.5 SCI Initialization (Simple SPI Mode) ... 830
29.8.6 Transmission and Reception of Serial Data in Simple SPI Mode ... 830
29.9 Bit Rate Modulation Function ... 831
29.10 Interrupt Sources ... 832
29.10.1 Buffer Operations for SCIn_TXI and SCIn_RXI Interrupts (non-FIFO selected) ... 832
29.10.2 Buffer Operations for SCIn_TXI and SCIn_RXI Interrupts (FIFO selected) ... 832
29.10.3 Interrupts in Asynchronous, Clock Synchronous, and Simple SPI Modes ... 832
29.10.4 Interrupts in Smart Card Interface Mode ... 833
29.10.5 Interrupts in Simple IIC Mode ... 834
29.11 Event Linking ... 835
29.12 Address non-match event output (SCI0_DCUF) ... 836
29.13 Noise Cancellation Function ... 836
29.14 Usage Notes ... 837
29.14.1 Settings for the Module-Stop State ... 837
29.14.2 SCI Operations during Low Power State ... 837
29.14.3 Break Detection and Processing ... 841
29.14.4 Mark State and Production of Breaks ... 842
29.14.5 Receive Error Flags and Transmit Operations in Clock Synchronous Mode and Simple SPI Mode ... 842
29.14.6 Restrictions on Clock Synchronous Transmission in Clock Synchronous and Simple SPI Modes ... 842
29.14.7 Restrictions on Using DMAC or DTC ... 843
29.14.8 Notes on Starting Transfer ... 844
29.14.9 External Clock Input in Clock Synchronous and Simple SPI Modes ... 844
29.14.10 Limitations on Simple SPI Mode ... 844
30. IrDA Interface ... 845
30.1 Overview ... 845
30.2 Register Descriptions ... 846
30.2.1 IrDA Control Register (IRCR) ... 846
30.3 Operation ... 846
30.3.1 IrDA Interface Setup Procedure ... 846
30.3.2 Transmission ... 846
30.3.3 Reception ... 847
30.4 Usage Notes ... 847
30.4.1 Module Stop Function Setting ... 847
30.4.2 Asynchronous Reference Clock for SCI1 ... 847
31. I2C Bus Interface (IIC) ... 848
31.1 Overview ... 848
31.2 Register Descriptions ... 851
31.2.1 I2C Bus Control Register 1 (ICCR1) ... 851
31.2.2 I2C Bus Control Register 2 (ICCR2) ... 853
31.2.3 I2C Bus Mode Register 1 (ICMR1) ... 856

31.2.4 I2C Bus Mode Register 2 (ICMR2) 857
31.2.5 I2C Bus Mode Register 3 (ICMR3) 858
31.2.6 I2C Bus Function Enable Register (ICFER) 860
31.2.7 I2C Bus Status Enable Register (ICSER) 862
31.2.8 I2C Bus Interrupt Enable Register (ICIER) 863
31.2.9 I2C Bus Status Register 1 (ICSR1) 864
31.2.10 I2C Bus Status Register 2 (ICSR2) 866
31.2.11 I2C Bus Wakeup Unit Register (ICWUR) 869
31.2.12 Reserved (ICWUR2) 869
31.2.13 Slave Address Register Ly (SARLy) (y = 0 to 2) 870
31.2.14 Slave Address Register Uy (SARUy) (y = 0 to 2) 870
31.2.15 I2C Bus Bit Rate Low-Level Register (ICBRL) 871
31.2.16 I2C Bus Bit Rate High-Level Register (ICBRH) 871
31.2.17 I2C Bus Transmit Data Register (ICDRT) 873
31.2.18 I2C Bus Receive Data Register (ICDRR) 873
31.2.19 I2C bus Shift Register (ICDRS) 873
31.3 Operation 874
31.3.1 Communication Data Format 874
31.3.2 Initial Settings 875
31.3.3 Master Transmit Operation 876
31.3.4 Master Receive Operation 879
31.3.5 Slave Transmit Operation 884
31.3.6 Slave Receive Operation 887
31.4 SCL Synchronization Circuit 889
31.5 SDA Output Delay Function 890
31.6 Digital Noise Filter Circuits 891
31.7 Address Match Detection 892
31.7.1 Slave Address Match Detection 892
31.7.2 Detection of General Call Address 894
31.7.3 Device ID Address Detection 894
31.7.4 Host Address Detection 896
31.8 Wakeup Function 897
31.8.1 Normal Wakeup Mode 1 897
31.8.2 Normal Wakeup Mode 2 900
31.8.3 Command Return Mode/ EEP Response Mode (Special Wakeup Mode) 902
31.8.4 Precautions for WFI Instruction Execution 905
31.9 Automatic Low-Hold Function for SCL 906
31.9.1 Function to Prevent Wrong Transmission of Transmit Data 906
31.9.2 NACK Reception Transfer Suspension Function 907
31.9.3 Function to Prevent Failure to Receive Data 907
31.10 Arbitration-Lost Detection Functions 909

31.10.1 Master Arbitration-Lost Detection (MALE Bit) ... 909
31.10.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit) ... 911
31.10.3 Slave Arbitration-Lost Detection (SALE Bit) ... 912
31.11 Start, Restart, and Stop Condition Issuing Function ... 913
31.11.1 Issuing a Start Condition ... 913
31.11.2 Issuing a Restart Condition ... 913
31.11.3 Issuing a Stop Condition ... 915
31.12 Bus Hanging ... 916
31.12.1 Timeout Function ... 916
31.12.2 Extra SCL Clock Cycle Output Function ... 917
31.12.3 IIC Reset and Internal Reset ... 918
31.13 SMBus Operation ... 919
31.13.1 SMBus Timeout Measurement ... 919
31.13.2 Packet Error Code (PEC) ... 920
31.13.3 SMBus Host Notification Protocol (Notify ARP Master Command) ... 920
31.14 Interrupt Sources ... 921
31.14.1 Buffer Operation for IICn_TXI and IICn_RXI Interrupts ... 921
31.15 Register States when Issuing each Condition ... 922
31.16 Event Link Output ... 923
31.16.1 Interrupt Handling and Event Linking ... 923
31.17 Usage Notes ... 923
31.17.1 Settings for the Module-Stop State ... 923
31.17.2 Notes on Starting Transfer ... 923
32. Controller Area Network (CAN) Module ... 924
32.1 Overview ... 924
32.2 Register Descriptions ... 926
32.2.1 Control Register (CTLR) ... 926
32.2.2 Bit Configuration Register (BCR) ... 929
32.2.3 Mask Register k (MKRk) (k = 0 to 7) ... 931
32.2.4 FIFO Received ID Compare Registers 0 and 1 (FIDCR0 and FIDCR1) ... 932
32.2.5 Mask Invalid Register (MKIVLR) ... 933
32.2.6 Mailbox Register j (MBj_ID, MBj_DL, MBj_Dm, MBj_TS) (j = 0 to 31, m = 0 to 7) ... 933
32.2.7 Mailbox Interrupt Enable Register (MIER) ... 937
32.2.8 Mailbox Interrupt Enable Register for FIFO Mailbox Mode (MIER_FIFO) ... 938
32.2.9 Message Control Register for Transmit (MCTL_TXj) (j = 0 to 31) ... 939
32.2.10 Message Control Register for Receive (MCTL_RXj) (j = 0 to 31) ... 941
32.2.11 Receive FIFO Control Register (RFCR) ... 943
32.2.12 Receive FIFO Pointer Control Register (RFPCR) ... 945
32.2.13 Transmit FIFO Control Register (TFCR) ... 945
32.2.14 Transmit FIFO Pointer Control Register (TFPCR) ... 947
32.2.15 Status Register (STR) ... 947

32.2.16 Mailbox Search Mode Register (MSMR) 949
32.2.17 Mailbox Search Status Register (MSSR) 950
32.2.18 Channel Search Support Register (CSSR) 951
32.2.19 Acceptance Filter Support Register (AFSR) 952
32.2.20 Error Interrupt Enable Register (EIER) 953
32.2.21 Error Interrupt Factor Judge Register (EIFR) 954
32.2.22 Receive Error Count Register (RECR) 956
32.2.23 Transmit Error Count Register (TECR) 956
32.2.24 Error Code Store Register (ECSR) 956
32.2.25 Time Stamp Register (TSR) 958
32.2.26 Test Control Register (TCR) 958
32.3 Modes of Operation 960
32.3.1 CAN Reset Mode 961
32.3.2 CAN Halt Mode 962
32.3.3 CAN Sleep Mode 963
32.3.4 CAN Operation Mode (Excluding Bus-Off State) 963
32.3.5 CAN Operation Mode (Bus-Off State) 964
32.4 Data Transfer Rate Configuration 965
32.4.1 Clock Setting 965
32.4.2 Bit Time Setting 965
32.4.3 Data Transfer Rate 966
32.5 Mailbox and Mask Register Structure 967
32.6 Acceptance Filtering and Masking Functions 968
32.7 Reception and Transmission 970
32.7.1 Reception 971
32.7.2 Transmission 973
32.8 Interrupt 974
32.9 Usage Notes 975
32.9.1 Settings for Module-Stop State 975
32.9.2 Settings for the Operating Clock 975
33. Serial Peripheral Interface (SPI) 976
33.1 Overview 976
33.2 Register Descriptions 979
33.2.1 SPI Control Register (SPCR) 979
33.2.2 SPI Slave Select Polarity Register (SSLP) 980
33.2.3 SPI Pin Control Register (SPPCR) 981
33.2.4 SPI Status Register (SPSR) 982
33.2.5 SPI Data Register (SPDR/SPDR_HA) 984
33.2.6 SPI Sequence Control Register (SPSCR) 987
33.2.7 SPI Sequence Status Register (SPSSR) 988
33.2.8 SPI Bit Rate Register (SPBR) 989

33.2.9 SPI Data Control Register (SPDCR) 990
33.2.10 SPI Clock Delay Register (SPCKD) 991
33.2.11 SPI Slave Select Negation Delay Register (SSLND) 992
33.2.12 SPI Next-Access Delay Register (SPND) 992
33.2.13 SPI Control Register 2 (SPCR2) 993
33.2.14 SPI Command Registers 0 to 7 (SPCMD0 to SPCMD7) 994
33.3 Operation 996
33.3.1 Overview of SPI Operations 996
33.3.2 Controlling SPI Pins 997
33.3.3 SPI System Configuration Examples 998
33.3.4 Data Format 1003
33.3.5 Transfer Formats 1012
33.3.6 Data Transfer Modes 1014
33.3.7 Transmit Buffer Empty and Receive Buffer Full Interrupts 1016
33.3.8 Error Detection 1018
33.3.9 Initializing the SPI 1022
33.3.10 SPI Operation 1023
33.3.11 Clock Synchronous Operation 1036
33.3.12 Loopback Mode 1041
33.3.13 Self-Diagnosis of Parity Bit Function 1042
33.3.14 Interrupt Sources 1043
33.4 Event Link Operation 1044
33.4.1 Receive Buffer Full Event Output 1044
33.4.2 Transmit Buffer Empty Event Output 1044
33.4.3 Mode Fault, Underrun, Overrun, or Parity Error Event Output 1044
33.4.4 SPI Idle Event Output 1044
33.4.5 Transmission-Complete Event Output 1045
33.5 Usage Notes 1045
33.5.1 Settings for the Module-Stop State 1045
33.5.2 Constraint on Low Power Function 1045
33.5.3 Constraint on Starting Transfer 1045
33.5.4 Constraint on Mode Fault, Underrun, Overrun or Parity Error Event Output 1045
33.5.5 Constraint on the SPRF and SPTEF Flags 1045
34. Quad Serial Peripheral Interface (QSPI) 1046
34.1 Overview 1046
34.2 Register Descriptions 1047
34.2.1 Transfer Mode Control Register (SFMSMD) 1047
34.2.2 Chip Selection Control Register (SFMSSC) 1048
34.2.3 Clock Control Register (SFMSKC) 1049
34.2.4 Status Register (SFMSST) 1050
34.2.5 Communication Port Register (SFMCOM) 1051

34.2.6 Communication Mode Control Register (SFMCMD) 1051
34.2.7 Communication Status Register (SFMCST) 1052
34.2.8 Instruction Code Register (SFMSIC) 1052
34.2.9 Address Mode Control Register (SFMSAC) 1053
34.2.10 Dummy Cycle Control Register (SFMSDC) 1054
34.2.11 SPI Protocol Control Register (SFMSPC) 1055
34.2.12 Port Control Register (SFMPMD) 1055
34.2.13 External QSPI Address Register (SFMCNT1) 1056
34.3 Memory Map 1057
34.3.1 Internal Bus Space 1057
34.3.2 Address Width of the SPI Space and SPI Bus 1058
34.4 SPI Bus 1059
34.4.1 SPI Protocol 1059
34.4.2 SPI Mode 1061
34.5 SPI Bus Timing Adjustment 1062
34.5.1 SPI Bus Reference Cycles 1062
34.5.2 QSPCLK Signal Duty Ratio 1063
34.5.3 Minimum High-Level Width of QSSL Signal 1063
34.5.4 QSSL Signal Setup Time 1063
34.5.5 QSSL Signal Hold Time 1064
34.5.6 Hold Time of the Serial Data Output Enable 1064
34.5.7 Setup Time of Serial Data Output 1065
34.5.8 Hold Time of Serial Data Output 1065
34.5.9 Serial Data Receiving Latency 1066
34.6 SPI Instruction Set Used for Flash Access 1067
34.6.1 Types of SPI Instructions Automatically Generated 1067
34.6.2 Standard Read Instruction 1068
34.6.3 Fast Read Instruction 1069
34.6.4 Fast Read Dual Output Instruction 1070
34.6.5 Fast Read Dual I/O Instruction 1071
34.6.6 Fast Read Quad Output Instruction 1072
34.6.7 Fast Read Quad I/O Instruction 1073
34.6.8 Enter 4-Byte Mode Instruction 1074
34.6.9 Exit 4-byte Mode Instruction 1074
34.6.10 Write Enable Instruction 1074
34.7 SPI Bus Cycle Arrangement 1075
34.7.1 Flash Read Based on Individual Conversion 1075
34.7.2 Flash Read Using Prefetch Function 1075
34.7.3 Halt of Prefetching 1076
34.7.4 Direct Specification of Prefetch Destination 1076
34.7.5 Prefetch State Polling 1076

34.7.6 Flash Read Using SPI Bus Cycle Extension Function ... 1076
34.8 XIP Control ... 1077
34.8.1 Setting XIP Mode ... 1077
34.8.2 Releasing XIP Mode ... 1078
34.9 QIO2 and QIO3 Pin States ... 1078
34.10 Direct Communication Mode ... 1078
34.10.1 About Direct Communication ... 1078
34.10.2 Direct Communication Mode ... 1078
34.10.3 SPI Bus Cycle Generation in Direct Communication ... 1079
34.11 Operation ... 1080
34.11.1 Procedure for Modifying Settings of Multiple Control Registers ... 1080
34.12 Interrupt ... 1080
34.13 Usage Note ... 1080
34.13.1 Setting for the Module-Stop State ... 1080
35. Cyclic Redundancy Check (CRC) Calculator ... 1081
35.1 Overview ... 1081
35.2 Register Descriptions ... 1082
35.2.1 CRC Control Register 0 (CRCCR0) ... 1082
35.2.2 CRC Control Register 1 (CRCCR1) ... 1082
35.2.3 CRC Data Input Register (CRCDIR/CRCDIR_BY) ... 1083
35.2.4 CRC Data Output Register (CRCDOR/CRCDOR_HA/CRCDOR_BY) ... 1083
35.2.5 Snoop Address Register (CRCSAR) ... 1084
35.3 Operation ... 1084
35.3.1 Basic Operation ... 1084
35.3.2 CRC Snoop ... 1087
35.4 Usage Notes ... 1088
35.4.1 Settings for the Module-Stop State ... 1088
35.4.2 Note on Transmission ... 1088
36. Serial Sound Interface (SSI) ... 1089
36.1 Overview ... 1089
36.2 Register Description ... 1091
36.2.1 Control Register (SSICR) ... 1091
36.2.2 Status Register (SSISR) ... 1095
36.2.3 FIFO Control Register (SSIFCR) ... 1097
36.2.4 FIFO Status Register (SSIFSR) ... 1098
36.2.5 Transmit FIFO Data Register (SSIFTDR) ... 1100
36.2.6 Receive FIFO Data Register (SSIFRDR) ... 1100
36.2.7 TDM Mode Register (SSITDMR) ... 1101
36.3 Operation ... 1102
36.3.1 Bus Format ... 1102
36.3.2 Non-compression Mode ... 1102

36.3.3 WS Continue Mode 1108
36.3.4 Operating States 1108
36.3.5 Transmit Operation 1109
36.3.6 Receive Operation 1112
36.3.7 Serial Bit Clock Control 1113
36.4 Interrupt Sources 1114
36.5 Usage Notes 1114
36.5.1 Setting for the Module-Stop State 1114
36.5.2 Notes on Changing Transfer Modes 1114
36.5.3 Constraints on the WS Continue Mode 1114
37. SD/MMC Host Interface (SDHI) 1115
37.1 Overview 1115
37.2 Register Descriptions 1117
37.2.1 Command Type Register (SD_CMD) 1117
37.2.2 SD Command Argument Register (SD_ARG) 1118
37.2.3 SD Command Argument Register 1 (SD_ARG1) 1118
37.2.4 Data Stop Register (SD_STOP) 1119
37.2.5 Block Count Register (SD_SECCNT) 1120
37.2.6 SD Card Response Register 10 (SD_RSP10), SD Card Response Register 32 (SD_RSP32), SD Card Response Register 54 (SD_RSP54) 1120
37.2.7 SD Card Response Register 1 (SD_RSP1), SD Card Response Register 3 (SD_RSP3), SD Card Response Register 5 (SD_RSP5) 1120
37.2.8 SD Card Response Register 76 (SD_RSP76) 1121
37.2.9 SD Card Response Register 7 (SD_RSP7) 1121
37.2.10 SD Card Interrupt Flag Register 1 (SD_INFO1) 1122
37.2.11 SD Card Interrupt Flag Register 2 (SD_INFO2) 1124
37.2.12 SD INFO1 Interrupt Mask Register (SD_INFO1_MASK) 1128
37.2.13 SD INFO2 Interrupt Mask Register (SD_INFO2_MASK) 1129
37.2.14 SD Clock Control Register (SD_CLK_CTRL) 1130
37.2.15 Transfer Data Length Register (SD_SIZE) 1131
37.2.16 SD Card Access Control Option Register (SD_OPTION) 1132
37.2.17 SD Error Status Register 1 (SD_ERR_STS1) 1133
37.2.18 SD Error Status Register 2 (SD_ERR_STS2) 1134
37.2.19 SD Buffer Register (SD_BUF0) 1135
37.2.20 SDIO Mode Control Register (SDIO_MODE) 1135
37.2.21 SDIO Interrupt Flag Register (SDIO_INFO1) 1137
37.2.22 SDIO INFO1 Interrupt Mask Register (SDIO_INFO1_MASK) 1138
37.2.23 DMA Mode Enable Register (SD_DMAEN) 1138
37.2.24 Software Reset Register (SOFT_RST) 1139
37.2.25 SD Interface Mode Setting Register (SDIF_MODE) 1140

37.2.26 Swap Control Register (EXT_SWAP) ... 1140
37.3 Operation ... 1141
37.3.1 SD/MMC Interface ... 1141
37.3.2 Card Detect/Write Protect ... 1143
37.3.3 Interrupt Request and DMA Transfer Request ... 1144
37.3.4 Communication Errors and Timeouts ... 1145
37.3.5 Command without Data Transfer [SD/MMC] ... 1147
37.3.6 Single Block Read [SD/MMC] ... 1149
37.3.7 Single Block Write [SD/MMC] ... 1151
37.3.8 Multiple Block Read [SD/MMC] ... 1153
37.3.9 Multiple Block Write (SD/MMC using internal timer) ... 1155
37.3.10 Multiple Block Write (MMC using external timer) ... 1157
37.3.11 IO_RW_DIRECT Command (SD: CMD52) ... 1159
37.3.12 IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) ... 1160
37.3.13 IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write) ... 1162
37.3.14 DMA Transfer [SD/MMC] ... 1164
37.3.15 Example of SD_CMD Register Setting ... 1165
37.4 Usage Notes ... 1167
37.4.1 SD_BUF Illegal Write Access [SD/MMC] ... 1167
37.4.2 Block Number Constraint for Multiple Block Read [SD] ... 1167
37.4.3 Automatic Control of SD/MMC Clock Output [SD/MMC] ... 1168
37.4.4 Control of the C52PUB Setting for Multiple Block Write [SD] ... 1168
37.4.5 Notes on SD_CLK_CTRL Register Settings [SD/MMC] ... 1168
37.4.6 Specification Limitations ... 1169
37.4.7 STP Bit Setting during Multiple Block Read [SD/MMC] ... 1169
37.4.8 Register Setting Notes ... 1169
38. Boundary Scan ... 1170
38.1 Overview ... 1170
38.2 Register Descriptions ... 1171
38.2.1 Instruction Register (JTIR) ... 1171
38.2.2 ID Code Register (JTIDR) ... 1172
38.2.3 Bypass Register (JTBPR) ... 1172
38.2.4 Boundary Scan Register (JTBSR) ... 1172
38.3 Operations ... 1173
38.3.1 TAP Controller ... 1173
38.3.2 List of Commands ... 1173
38.4 Usage Note ... 1175
39. 14-Bit A/D Converter (ADC14) ... 1176
39.1 Overview ... 1176
39.2 Register Descriptions ... 1180
39.2.1 A/D Data Registers y (ADDRy), A/D Data Duplexing Register (ADDBLDR),

A/D Data Duplexing Register A (ADDBLDRA), A/D Data Duplexing Register B (ADDBLDRB), A/D Temperature Sensor Data Register (ADTSDR), A/D Internal Reference Voltage Data Register (ADOCDR) 1180
39.2.2 A/D Self-Diagnosis Data Register (ADRD) 1183
39.2.3 A/D Control Register (ADCSR) 1185
39.2.4 A/D Channel Select Register A0 (ADANSA0) 1188
39.2.5 A/D Channel Select Register A1 (ADANSA1) 1189
39.2.6 A/D Channel Select Register B0 (ADANSB0) 1189
39.2.7 A/D Channel Select Register B1 (ADANSB1) 1190
39.2.8 A/D-Converted Value Addition/Average Channel Select Register 0 (ADADS0) 1190
39.2.9 A/D-Converted Value Addition/Average Channel Select Register 1 (ADADS1) 1191
39.2.10 A/D-Converted Value Addition/Average Count Select Register (ADADC) 1192
39.2.11 A/D Control Extended Register (ADCER) 1193
39.2.12 A/D Conversion Start Trigger Select Register (ADSTRGR) 1194
39.2.13 A/D Conversion Extended Input Control Register (ADEXICR) 1196
39.2.14 A/D Sampling State Register n (ADSSTRn) (n = 00 to 15, L, T, O) 1197
39.2.15 A/D Disconnection Detection Control Register (ADDISCR) 1198
39.2.16 A/D Group Scan Priority Control Register (ADGSPCR) 1199
39.2.17 A/D Compare Function Control Register (ADCMPCR) 1200
39.2.18 A/D Compare Function Window A Channel Select Register 0 (ADCMPANSR0) 1201
39.2.19 A/D Compare Function Window A Channel Select Register 1 (ADCMPANSR1) 1202
39.2.20 A/D Compare Function Window A Extended Input Select Register (ADCMPANSER) 1202
39.2.21 A/D Compare Function Window A Comparison Condition Setting Register 0 (ADCMPLR0) 1203
39.2.22 A/D Compare Function Window A Comparison Condition Setting Register 1 (ADCMPLR1) 1204
39.2.23 A/D Compare Function Window A Extended Input Comparison Condition Setting Register (ADCMPLER) 1205
39.2.24 A/D Compare Function Window A Lower-Side Level Setting Register (ADCMPDR0), A/D Compare Function Window A Upper-Side Level Setting Register (ADCMPDR1), A/D Compare Function Window B Lower-Side Level Setting Register (ADWINLLB), A/D Compare Function Window B Upper-Side Level Setting Register (ADWINULB) 1206
39.2.25 A/D Compare Function Window A Channel Status Register 0 (ADCMPSR0) 1207
39.2.26 A/D Compare Function Window A Channel Status Register1 (ADCMPSR1) 1208
39.2.27 A/D Compare Function Window A Extended Input Channel Status Register (ADCMPSER) 1209
39.2.28 A/D Compare Function Window B Channel Select Register (ADCMPBNSR) 1210
39.2.29 A/D Compare Function Window B Status Register (ADCMPBSR) 1212
39.2.30 A/D Compare Function Window A/B Status Monitor Register (ADWINMON) 1213
39.2.31 A/D High-Potential/Low-Potential Reference Voltage Control Register (ADHVREFCNT) 1214
39.3 Operation 1215
39.3.1 Scanning Operation 1215

39.3.2 Single-Scan Mode ... 1216
39.3.3 Continuous-Scan Mode ... 1220
39.3.4 Group-Scan Mode ... 1222
39.3.5 Compare Function (Window A, Window B) ... 1231
39.3.6 Analog Input Sampling and Scan Conversion Time ... 1234
39.3.7 Usage Example of A/D Data Register Automatic Clearing Function ... 1236
39.3.8 A/D-Converted Value Addition/Average Mode ... 1236
39.3.9 Disconnection Detection Assist Function ... 1237
39.3.10 Starting A/D Conversion with Asynchronous Trigger ... 1238
39.3.11 Starting A/D Conversion with Synchronous Trigger from Peripheral Module ... 1238
39.4 Interrupt Sources and DTC/DMAC Transfer Requests ... 1239
39.4.1 Interrupt Requests ... 1239
39.5 Event Link Function ... 1240
39.5.1 Event output to the ELC ... 1240
39.5.2 14-bit ADC operation by an event from the ELC ... 1240
39.6 Selecting Reference Voltage ... 1240
39.7 A/D Conversion Procedure when Selecting Internal Reference Voltage as High-Potential Reference Voltage ... 1240
39.8 Usage Notes ... 1241
39.8.1 Notes on Reading Data Registers ... 1241
39.8.2 Notes on Stopping A/D Conversion ... 1241
39.8.3 A/D Conversion Restarting Timing and Termination Timing ... 1242
39.8.4 Notes on Scan End Interrupt Handling ... 1242
39.8.5 Module Stop Function Setting ... 1242
39.8.6 Notes on Entering Low Power Consumption States ... 1242
39.8.7 Error in Absolute Accuracy when Disconnection Detection Assistance is in Use ... 1242
39.8.8 ADHSC Bit Rewriting Procedure ... 1242
39.8.9 Notes on Operating Modes and Status Bits ... 1242
39.8.10 Notes on Board Design ... 1243
39.8.11 Notes on Noise Prevention ... 1243
39.8.12 Port Setting when Using the 14-bit A/D Converter Input ... 1243
39.8.13 Relationship between A/D converter, OPAMP, ACMPHS, and ACMPLP ... 1244
39.8.14 Notes on Canceling Software Standby Mode ... 1244
40. 12-Bit D/A Converter (DAC12) ... 1245
40.1 Overview ... 1245
40.2 Register Descriptions ... 1246
40.2.1 D/A Data Register m (DADRm) (m = 0, 1) ... 1246
40.2.2 D/A Control Register (DACR) ... 1246
40.2.3 DADR0 Format Select Register (DADPR) ... 1247
40.2.4 D/A A/D Synchronous Start Control Register (DAADSCR) ... 1247
40.2.5 D/A VREF Control Register (DAVREFCR) ... 1248

40.3 Operation 1249
40.3.1 Minimizing Interference between D/A and A/D Conversion 1249
40.3.2 Notes on Using the Internal Reference Voltage as the Reference Voltage 1251
40.4 Event Link Operation Setting Procedure 1251
40.4.1 DA0 event link operation setting procedure 1251
40.4.2 DA1 event link operation setting procedure 1252
40.5 Usage Notes on Event Link Operation 1252
40.6 Usage Notes 1252
40.6.1 Settings for the Module-Stop Function 1252
40.6.2 DAC12 Operation in Module-Stop State 1252
40.6.3 DAC12 Operation in Software Standby Mode 1252
40.6.4 Note on Usage when Interference Reduction between D/A and A/D Conversion is Enabled 1252
41. Temperature Sensor (TSN) 1253
41.1 Overview 1253
41.2 Register Descriptions 1254
41.2.1 Temperature Sensor Calibration Data Register H (TSCDRH) 1254
41.2.2 Temperature Sensor Calibration Data Register L (TSCDRL) 1254
41.3 Using the Temperature Sensor 1255
41.3.1 Preparation for Using Temperature Sensor 1255
41.3.2 Procedure for Using the Temperature Sensor 1256
42. Operational Amplifier (OPAMP) 1257
42.1 Overview 1257
42.2 Register Descriptions 1258
42.2.1 Operational Amplifier Mode Control Register (AMPMC) 1258
42.2.2 Operational Amplifier Trigger Mode Control Register (AMPTRM) 1259
42.2.3 Operational Amplifier Activation Trigger Select Register (AMPTRS) 1260
42.2.4 Operational Amplifier Control Register (AMPC) 1260
42.2.5 Operational Amplifier Monitor Register (AMPMON) 1261
42.3 Operation 1262
42.3.1 State Transitions 1262
42.3.2 Operational Amplifier Control Operation 1263
42.4 Software Trigger Mode 1267
42.5 Activation Trigger Mode 1268
42.6 Activation and A/D Trigger Mode 1269
42.7 Usage Notes 1269
43. High-Speed Analog Comparator (ACMPHS) 1270
43.1 Overview 1270
43.2 Register Descriptions 1272
43.2.1 Comparator Control Register (CMPCTL) 1272
43.2.2 Comparator Input Select Register (CMPSEL0) 1273
43.2.3 Comparator Reference Voltage Select Register (CMPSEL1) 1273

43.2.4 Comparator Output Monitor Register (CMPMON) ... 1274
43.2.5 Comparator Output Control Register (CPIOC) ... 1274
43.3 Operation ... 1275
43.4 Noise Filter ... 1276
43.5 ACMPHS Interrupts ... 1277
43.6 ACMPHS Output to the Event Link Controller (ELC) ... 1277
43.7 ACMPHS Pin Output ... 1277
43.8 Usage Notes ... 1277
43.8.1 Settings for the Module-Stop Function ... 1277
43.8.2 Relationship with 14-bit A/D converter ... 1277
44. Low Power Analog Comparator (ACMPLP) ... 1278
44.1 Overview ... 1278
44.2 Register Descriptions ... 1281
44.2.1 ACMPLP Mode Setting Register (COMPMDR) ... 1281
44.2.2 ACMPLP Filter Control Register (COMPFIR) ... 1282
44.2.3 ACMPLP Output Control Register (COMPOCR) ... 1282
44.3 Operation ... 1283
44.4 Noise Filter ... 1286
44.5 ACMPLP Interrupts ... 1287
44.6 ELC Event Output ... 1287
44.7 Interrupt Handling and ELC Linking ... 1287
44.8 Comparator Pin Output ... 1287
44.9 Usage Notes ... 1287
44.9.1 Settings for the Module-Stop Function ... 1287
44.9.2 Relationship with A/D converter ... 1287
45. Capacitive Touch Sensing Unit (CTSU) ... 1288
45.1 Overview ... 1288
45.2 Register Descriptions ... 1290
45.2.1 CTSU Control Register 0 (CTSUCR0) ... 1290
45.2.2 CTSU Control Register 1 (CTSUCR1) ... 1291
45.2.3 CTSU Synchronous Noise Reduction Setting Register (CTSUSDPRS) ... 1293
45.2.4 CTSU Sensor Stabilization Wait Control Register (CTSUSST) ... 1294
45.2.5 CTSU Measurement Channel Register 0 (CTSUMCH0) ... 1295
45.2.6 CTSU Measurement Channel Register 1 (CTSUMCH1) ... 1297
45.2.7 CTSU Channel Enable Control Register 0 (CTSUCHAC0) ... 1298
45.2.8 CTSU Channel Enable Control Register 1 (CTSUCHAC1) ... 1298
45.2.9 CTSU Channel Enable Control Register 2 (CTSUCHAC2) ... 1299
45.2.10 CTSU Channel Enable Control Register 3 (CTSUCHAC3) ... 1299
45.2.11 CTSU Channel Enable Control Register 4 (CTSUCHAC4) ... 1300
45.2.12 CTSU Channel Transmit/Receive Control Register 0 (CTSUCHTRC0) ... 1300
45.2.13 CTSU Channel Transmit/Receive Control Register 1 (CTSUCHTRC1) ... 1301

45.2.14 CTSU Channel Transmit/Receive Control Register 2 (CTSUCHTRC2) ... 1301
45.2.15 CTSU Channel Transmit/Receive Control Register 3 (CTSUCHTRC3) ... 1302
45.2.16 CTSU Channel Transmit/Receive Control Register 4 (CTSUCHTRC4) ... 1302
45.2.17 CTSU High-Pass Noise Reduction Control Register (CTSUDCLKC) ... 1303
45.2.18 CTSU Status Register (CTSUST) ... 1303
45.2.19 CTSU High-Pass Noise Reduction Spectrum Diffusion Control Register (CTSUSSC) ... 1305
45.2.20 CTSU Sensor Offset Register 0 (CTSUSO0) ... 1306
45.2.21 CTSU Sensor Offset Register 1 (CTSUSO1) ... 1306
45.2.22 CTSU Sensor Counter (CTSUSC) ... 1308
45.2.23 CTSU Reference Counter (CTSURC) ... 1308
45.2.24 CTSU Error Status Register (CTSUERRS) ... 1309
45.3 Operation ... 1309
45.3.1 Principles of Measurement Operation ... 1309
45.3.2 Measurement Modes ... 1311
45.3.3 Functions Common to Multiple Modes ... 1320
45.4 Usage Notes ... 1322
45.4.1 Measurement Result Data (CTSUSC and CTSURC Counters) ... 1322
45.4.2 Software Trigger ... 1322
45.4.3 External Trigger ... 1323
45.4.4 Notes on Forcing Operation Stop ... 1323
45.4.5 TSCAP Pin ... 1323
45.4.6 Notes on Measurement Operation (CTSUCR0.CTSUSTRT Bit = 1) ... 1323
46. Data Operation Circuit (DOC) ... 1324
46.1 Overview ... 1324
46.2 Register Descriptions ... 1325
46.2.1 DOC Control Register (DOCR) ... 1325
46.2.2 DOC Data Input Register (DODIR) ... 1326
46.2.3 DOC Data Setting Register (DODSR) ... 1326
46.3 Operation ... 1326
46.3.1 Data Comparison Mode ... 1326
46.3.2 Data Addition Mode ... 1327
46.3.3 Data Subtraction Mode ... 1327
46.4 Interrupt Request and Output to the Event Link Controller (ELC) ... 1328
46.5 Usage Notes ... 1328
46.5.1 Settings for the Module-Stop State ... 1328
47. SRAM ... 1329
47.1 Overview ... 1329
47.2 Register Descriptions ... 1329
47.2.1 SRAM Parity Error Operation After Detection Register (PARIOAD) ... 1329
47.2.2 SRAM Protection Register (SRAMPRCR) ... 1330

47.2.3 ECC Operating Mode Control Register (ECCMODE) ... 1330
47.2.4 ECC 2-Bit Error Status Register (ECC2STS) ... 1331
47.2.5 ECC 1-Bit Error Information Update Enable Register (ECC1STSEN) ... 1331
47.2.6 ECC 1-Bit Error Status Register (ECC1STS) ... 1332
47.2.7 ECC Protection Register (ECCPRCR) ... 1332
47.2.8 ECC Protection Register 2 (ECCPRCR2) ... 1333
47.2.9 ECC Test Control Register (ECCETST) ... 1333
47.2.10 SRAM ECC Error Operation After Detection Register (ECCOAD) ... 1334
47.3 Operation ... 1335
47.3.1 Low Power Consumption Function ... 1335
47.3.2 ECC Function ... 1335
47.3.3 ECC Error Generation ... 1335
47.3.4 ECC Decoder Testing ... 1336
47.3.5 Parity Calculation Function ... 1337
47.3.6 SRAM Error Sources ... 1338
47.3.7 Access Cycle ... 1339
47.4 Usage Notes ... 1339
47.4.1 Instruction Fetch from the SRAM area ... 1339
47.4.2 Store Buffer of SRAM ... 1339
48. Flash Memory ... 1340
48.1 Overview ... 1340
48.2 Memory Structure ... 1341
48.3 Flash Cache ... 1343
48.3.1 Overview ... 1343
48.3.2 Register Descriptions ... 1344
48.4 Operation ... 1345
48.4.1 Notice to use Flash Cache ... 1345
48.5 Operating Modes Associated with the Flash Memory ... 1345
48.5.1 ID Code Protection ... 1346
48.6 Overview of Functions ... 1346
48.6.1 Configuration Area Bit Map ... 1348
48.6.2 Startup Area Select ... 1349
48.6.3 Protection by Access Window ... 1349
48.7 Programming Commands ... 1350
48.8 Suspend Operation ... 1350
48.9 Protection ... 1350
48.10 Serial Programming Mode ... 1350
48.10.1 SCI Boot Mode ... 1350
48.10.2 USB Boot Mode ... 1351
48.11 Using a Serial Programmer ... 1352
48.11.1 Serial Programming ... 1352

48.12 Self-Programming ... 1352
48.12.1 Overview ... 1352
48.12.2 Background Operation ... 1353
48.13 Reading the Flash Memory ... 1353
48.13.1 Reading the Code Flash Memory ... 1353
48.13.2 Reading the Data Flash Memory ... 1353
48.14 Usage Notes ... 1353
48.14.1 Erase Suspended Area ... 1353
48.14.2 Suspension with Erase Suspend Commands ... 1353
48.14.3 Constraints on Additional Writes ... 1353
48.14.4 Reset during Programming and Erasure ... 1353
48.14.5 Non-maskable Interrupt Disabled during Programming and Erasure ... 1353
48.14.6 Location of Interrupt Vectors during Programming and Erasure ... 1353
48.14.7 Programming and Erasure in Low-Speed Operating Mode ... 1354
48.14.8 Abnormal Termination during Programming and Erasure ... 1354
48.14.9 Actions Prohibited during Programming and Erasure ... 1354
49. Segment LCD Controller/Driver (SLCDC) ... 1355
49.1 Overview ... 1355
49.2 Register Descriptions ... 1359
49.2.1 LCD Mode Register 0 (LCDM0) ... 1359
49.2.2 LCD Mode Register 1 (LCDM1) ... 1360
49.2.3 LCD Clock Control Register 0 (LCDC0) ... 1361
49.2.4 LCD Boost Level Control Register (VLCD) ... 1362
49.3 LCD Display Data Registers ... 1363
49.4 Selection of LCD Display Data Register ... 1366
49.4.1 A-Pattern Area and B-pattern Area Data Display ... 1366
49.4.2 Blinking Display (Alternately Displaying A-Pattern and B-Pattern Area Data) ... 1366
49.5 Setting LCD Controller/Driver ... 1368
49.6 Operation Stop Procedure ... 1371
49.7 Supplying LCD Drive Voltages VL1, VL2, VL3, and VL4 ... 1372
49.7.1 External Resistance Division Method ... 1372
49.7.2 Internal Voltage Boosting Method ... 1374
49.7.3 Capacitor Split Method ... 1375
49.8 Common and Segment Signals ... 1376
49.9 Display Modes ... 1383
49.9.1 Static Display Example ... 1383
49.9.2 Two-Time-Slice Display Example ... 1386
49.9.3 Three-Time-Slice Display Example ... 1389
49.9.4 Four-Time-Slice Display Example ... 1393
49.9.5 Eight-Time-Slice Display Example ... 1397

50. Secure Cryptographic Engine (SCE5) 1401
50.1 Overview 1401
50.2 Operation 1403
50.2.1 Encryption Engine 1403
50.2.2 Encryption and Decryption 1404
50.3 Usage Notes 1404
50.3.1 Software Standby Mode 1404
50.3.2 Settings for the Module-Stop Function 1404
51. Internal Voltage Regulator 1405
51.1 Overview 1405
51.2 Operation 1405
52. Electrical Characteristics 1406
52.1 Absolute Maximum Ratings 1407
52.2 DC Characteristics 1409
52.2.1 Tj/Ta Definition 1409
52.2.2 I/O VIH, V_{IL} 1409
52.2.3 I/O I_{OH}, I_{OL} 1411
52.2.4 I/O V_{OH}, V_{OL}, and Other Characteristics 1412
52.2.5 I/O Pin Output Characteristics of Low Drive Capacity 1414
52.2.6 I/O Pin Output Characteristics of Middle Drive Capacity 1416
52.2.7 P408, P409 I/O Pin Output Characteristics of Middle Drive Capacity 1419
52.2.8 IIC I/O Pin Output Characteristics 1421
52.2.9 Operating and Standby Current 1422
52.2.10 VCC Rise and Fall Gradient and Ripple Frequency 1430
52.3 AC Characteristics 1431
52.3.1 Frequency 1431
52.3.2 Clock Timing 1434
52.3.3 Reset Timing 1438
52.3.4 Wakeup Time 1439
52.3.5 NMI and IRQ Noise Filter 1442
52.3.6 Bus Timing 1443
52.3.7 I/O Ports, POEG, GPT, AGT, KINT, and ADC14 Trigger Timing 1449
52.3.8 CAC Timing 1450
52.3.9 SCI Timing 1451
52.3.10 SPI Timing 1458
52.3.11 QSPI Timing 1463
52.3.12 IIC Timing 1465
52.3.13 SSI Timing 1467
52.3.14 SD/MMC Host Interface Timing 1469
52.3.15 CLKOUT Timing 1469
52.4 USB Characteristics 1470

52.4.1 USBFS Timing 1470
52.4.2 USB External Supply 1471
52.5 ADC14 Characteristics 1472
52.6 DAC12 Characteristics 1482
52.7 TSN Characteristics 1484
52.8 OSC Stop Detect Characteristics 1484
52.9 POR and LVD Characteristics 1485
52.10 Battery Backup Function Characteristics 1489
52.11 CTSU Characteristics 1491
52.12 Segment LCD Controller/Driver Characteristics 1491
52.12.1 Resistance Division Method 1491
52.12.2 Internal Voltage Boosting Method 1492
52.12.3 Capacitor Split Method 1494
52.13 Comparator Characteristics 1495
52.14 OPAMP Characteristics 1496
52.15 Flash Memory Characteristics 1497
52.15.1 Code Flash Memory Characteristics 1497
52.15.2 Data Flash Memory Characteristics 1498
52.16 Boundary Scan 1499
52.17 Joint Test Action Group (JTAG) 1500
52.17.1 Serial Wire Debug (SWD) 1502
Appendix 1. Port States in Each Processing Mode 1504
Appendix 2. Package Dimensions 1509
Appendix 3. I/O Registers 1516
3.1 Peripheral Base Addresses 1516
3.2 Access Cycles 1518
3.3 Register Descriptions 1520
Revision History 1547

High efficiency 48-MHz Arm® Cortex®-M4 microcontroller, up to 1-MB code flash memory, 192-KB SRAM, Segment LCD Controller, Capacitive Touch Sensing Unit, USB 2.0 Full-Speed Module, 14-bit A/D Converter, 12-bit D/A Converter, security and safety features

Features

■ Arm Cortex-M4 Core with Floating Point Unit (FPU)
- Armv7E-M architecture with DSP instruction set
- Maximum operating frequency: 48 MHz
- Support for 4-GB address space
- Arm Memory Protection Unit (Arm MPU) with 8 regions
- Debug and Trace: ITM, DWT, FPB, TPIU, ETB
- CoreSight™ Debug Port: JTAG-DP and SW-DP

■ Memory
- Up to 1-MB code flash memory
- 16-KB data flash memory (100,000 program/erase (P/E) cycles)
- Up to 192-KB SRAM
- Flash Cache (FCACHE)
- Memory Protection Unit (MPU)
- Memory Mirror Function (MMF)
- 128-bit unique ID

■ Connectivity
- USB 2.0 Full-Speed Module (USBFS)
  - On-chip transceiver with voltage regulator
  - Compliant with USB Battery Charging Specification 1.2
- Serial Communications Interface (SCI) × 6
  - UART
  - Simple IIC
  - Simple SPI
- Serial Peripheral Interface (SPI) × 2
- I^2C bus interface (IIC) × 3
- Controller Area Network (CAN) module
- Serial Sound Interface (SSI) × 2
- SD/MMC Host Interface (SDHI)
- Quad Serial Peripheral Interface (QSPI)
- IrDA interface
- External address space
  - 8- or 16-bit bus space selectable per area

■ Analog
- 14-Bit A/D Converter (ADC14)
- 12-Bit D/A Converter (DAC12) × 2
- High-Speed Analog Comparator (ACMPHS) × 2
- Low-Power Analog Comparator (ACMPLP) × 2
- Operational Amplifier (OPAMP) × 4
- Temperature Sensor (TSN)

■ Timers
- General PWM Timer 32-bit (GPT32) × 10
- Asynchronous General-Purpose Timer (AGT) × 2
  - VBATT support
- Watchdog Timer (WDT)

■ Safety
- Error Correction Code (ECC) in SRAM
- 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

■ System and Power Management
- Low power modes
- Realtime Clock (RTC) with calendar and Battery Backup support
- Event Link Controller (ELC)
- DMA Controller (DMAC) × 4
- Data Transfer Controller (DTC)
- Key Interrupt Function (KINT)
- Power-on reset
- Low Voltage Detection (LVD) with voltage settings

■ Security and Encryption
- AES128/256
- GHASH
- True Random Number Generator (TRNG)

■ Human Machine Interface (HMI)
- Segment LCD Controller (SLCDC)
  - Up to 52 segments × 4 commons
  - Up to 48 segments × 8 commons
- Capacitive Touch Sensing Unit (CTSU)

■ Multiple Clock Sources
- Main clock oscillator (MOSC)
  (1 to 20 MHz when VCC = 2.4 to 5.5 V)
  (1 to 8 MHz when VCC = 1.8 to 2.4 V)
  (1 to 4 MHz when VCC = 1.6 to 1.8 V)
- Sub-clock oscillator (SOSC) (32.768 kHz)
- High-speed on-chip oscillator (HOCO)
  (24, 32, 48, 64 MHz when VCC = 2.4 to 5.5 V)
  (24, 32, 48 MHz when VCC = 1.8 to 5.5 V)
  (24, 32 MHz when VCC = 1.6 to 5.5 V)
- Middle-speed on-chip oscillator (MOCO) (8 MHz)
- Low-speed on-chip oscillator (LOCO) (32.768 kHz)
- IWDT-dedicated on-chip oscillator (15 kHz)
- Clock trim function for HOCO/MOCO/LOCO
- Clock out support

■ General Purpose I/O Ports
- Up to 124 input/output pins
  - Up to 3 CMOS input
  - Up to 121 CMOS input/output
    - Up to 10 input/output 5-V tolerant
    - Up to 2 high current (20 mA)

■ Operating Voltage
- VCC: 1.6 to 5.5 V

■ Operating Temperature and Packages
- Ta = -40°C to +85°C
  - 145-pin LGA(7 mm × 7 mm, 0.5 mm pitch)
  - 121-pin BGA (8 mm × 8 mm, 0.65 mm pitch)
  - 100-pin LGA (7 mm × 7 mm, 0.65 mm pitch)
- Ta = -40°C to +105°C
  - 144-pin LQFP (20 mm × 20 mm, 0.5 mm pitch)
  - 100-pin LQFP (14 mm × 14 mm, 0.5 mm pitch)
  - 64-pin LQFP (10 mm × 10 mm, 0.5 mm pitch)
  - 64-pin QFN (8 mm × 8 mm, 0.4 mm pitch)

1. Overview

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

The MCU provides an optimal combination of low power, high performance Arm Cortex®-M4 core running up to 48 MHz with the following features:

- Up to 1-MB code flash memory
- 192-KB SRAM
- Segment LCD Controller (SLCDC)
- Capacitive Touch Sensing Unit (CTSU)
- USB 2.0 Full-Speed Module (USBFS)
- 14-bit A/D Converter (ADC14)
- 12-bit D/A Converter (DAC12)
- Security features.

1.1 Function Outline

Table 1.1 Arm core

Feature	Functional description
Arm Cortex-M4	• Maximum operating frequency: up to 48 MHz • Arm Cortex-M4 - Revision: r0p1-01rel0 - Armv7E-M architecture profile - Single precision floating-point unit compliant with the ANSI/IEEE Std 754-2008. • Arm Memory Protection Unit (Arm MPU) - Armv7 Protected Memory System Architecture - 8 protected regions. • SysTick timer - Driven by SYSTICCLK (LOCO) or ICLK.

Table 1.2 Memory

Feature	Functional description
Code flash memory	Maximum 1-MB code flash memory. See section 48, Flash Memory.
Data flash memory	16-KB data flash memory. See section 48, Flash Memory.
Option-setting memory	The option-setting memory determines the state of the MCU after a reset. See section 7, Option-Setting Memory.
Memory Mirror Function (MMF)	The Memory Mirror Function (MMF) can be configured to mirror the desired application image load address in code flash memory to the application image link address in the 23-bit unused memory space (memory mirror space addresses). Your application code is developed and linked to run from this MMF destination address. The application code does not need to know the load location where it is stored in code flash memory. See section 5, Memory Mirror Function (MMF).
SRAM	On-chip high-speed SRAM with either parity-bit or Error Correction Code (ECC). An area in SRAM0 provides error correction capability using ECC. See section 47, SRAM.

Table 1.3 System (1 of 2)

Feature	Functional description
Operating modes	Two operating modes: • Single-chip mode • SCI/USB boot mode. See section 3, Operating Modes.
Resets	14 resets: • RES pin reset • Power-on reset • VBATT-selected voltage power-on reset • Independent watchdog timer reset • Watchdog timer reset • Voltage monitor 0 reset • Voltage monitor 1 reset • Voltage monitor 2 reset • SRAM parity error reset • SRAM ECC error reset • Bus master MPU error reset • Bus slave MPU error reset • CPU stack pointer error reset • Software reset. See section 6, Resets.
Low Voltage Detection (LVD)	The Low Voltage Detection (LVD) function monitors the voltage level input to the VCC pin, and the detection level can be selected using a software program. See section 8, Low Voltage Detection (LVD).
Clocks	• Main clock oscillator (MOSC) • Sub-clock oscillator (SOSC) • High-speed on-chip oscillator (HOCO) • Middle-speed on-chip oscillator (MOCO) • Low-speed on-chip oscillator (LOCO) • PLL frequency synthesizer • IWDT-dedicated on-chip oscillator • Clock out support. See section 9, 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 to be used as a 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 10, Clock Frequency Accuracy Measurement Circuit (CAC).
Interrupt Controller Unit (ICU)	The Interrupt Controller Unit (ICU) controls which event signals are linked to the NVIC/DTC module and DMAC module. The ICU also controls NMI interrupts. See section 14, Interrupt Controller Unit (ICU).
Key Interrupt Function (KINT)	A key interrupt can be generated by setting the Key Return Mode Register (KRM) and inputting a rising or falling edge to the key interrupt input pins. See section 21, Key Interrupt Function (KINT).
Low power modes	Power consumption can be reduced in multiple ways, such as by setting clock dividers, controlling EBCLK output, stopping modules, selecting power control mode in normal operation, and transitioning to low power modes. See section 11, Low Power Modes.
Battery backup function	A battery backup function is provided for partial powering by a battery. The battery-powered area includes RTC, AGT, SOSC, LOCO, Wakeup Control, Backup Memory, VBATT_R Low Voltage Detection, and switches between VCC and VBATT. During normal operation, the battery powered area is powered by the main power supply, the VCC pin. When a VCC voltage drop is detected, the power source is switched to the dedicated battery backup power pin, the VBATT pin. When the voltage rises again, the power source is switched from the VBATT pin to the VCC pin. See section 12, Battery Backup Function.
Register write protection	The register write protection function protects important registers from being overwritten due to software errors. See section 13, Register Write Protection.

Table 1.3 System (2 of 2)

Feature	Functional description
Memory Protection Unit (MPU)	Four MPUs and a CPU stack pointer monitor function are provided for memory protection. See section 16, Memory Protection Unit (MPU).
Watchdog Timer (WDT)	The WDT is a 14-bit down-counter. It 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, a non-maskable interrupt or interrupt can be generated by an underflow. A refresh-permitted period can be set to refresh the counter and be used as the condition for detecting when the system runs out of control. See section 26, Watchdog Timer (WDT).
Independent Watchdog Timer (IWDT)	The 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/interrupt for a timer underflow. 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 on a reset, underflow, or refresh error, or by a refresh of the count value in the registers. See section 27, Independent Watchdog Timer (IWDT).

Table 1.4 Event link

Feature	Functional description
Event Link Controller (ELC)	The ELC uses the interrupt requests generated by various peripheral modules as event signals to connect them to different modules, enabling direct interaction between the modules without CPU intervention. See section 19, Event Link Controller (ELC).

Table 1.5 Direct memory access

Feature	Functional description
Data Transfer Controller (DTC)	A DTC module is provided for transferring data when activated by an interrupt request. See section 18, Data Transfer Controller (DTC).
DMA Controller (DMAC)	A 4-channel DMAC module is provided for transferring data without the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address. See section 17, DMA Controller (DMAC).

Table 1.6 External bus interface

Feature	Functional description
External bus	• CS area: Connected to the external devices (external memory interface) • QSPI area: Connected to the QSPI (external device interface).

Table 1.7 Timers

Feature	Functional description
General PWM Timer (GPT)	The GPT is a 32-bit timer with 10 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 23, General PWM Timer (GPT).
Port Output Enable for GPT (POEG)	Use the POEG function to place the General PWM Timer (GPT) output pins in the output disable state. See section 22, Port Output Enable for GPT (POEG).
Asynchronous General Purpose Timer (AGT)	The AGT is a 16-bit timer that can be used for pulse output, external pulse width or period measurement, and counting of external events. This 16-bit timer consists of a reload register and a down-counter. The reload register and the down-counter are allocated to the same address, and they can be accessed with the AGT register. See section 24, Asynchronous General Purpose Timer (AGT).
Realtime Clock (RTC)	The RTC has two counting modes, calendar count mode and binary count mode, that are controlled by the register settings. For calendar count mode, the RTC has a 100-year calendar from 2000 to 2099 and automatically adjusts dates for leap years. For binary count mode, the RTC counts seconds and retains the information as a serial value. Binary count mode can be used for calendars other than the Gregorian (Western) calendar. See section 25, Realtime Clock (RTC).

Table 1.8 Communication interfaces (1 of 2)

Feature	Functional description
Serial Communications Interface (SCI)	The SCI is configurable to five 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. Each SCI has FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured individually using an on-chip baud rate generator. See section 29, Serial Communications Interface (SCI).
IrDA Interface (IrDA)	The IrDA interface sends and receives IrDA data communication waveforms in cooperation with the SCI1 based on the IrDA (Infrared Data Association) standard 1.0. See section 30, IrDA Interface.
I^2C Bus Interface (IIC)	The 3-channel IIC module conforms with and provides a subset of the NXP I^2C bus (Inter-Integrated Circuit bus) interface functions. See section 31, I^2C Bus Interface (IIC).
Serial Peripheral Interface (SPI)	Two independent SPI channels are capable of high-speed, full-duplex synchronous serial communications with multiple processors and peripheral devices. See section 33, Serial Peripheral Interface (SPI).
Serial Sound Interface (SSI)	The SSI peripheral provides functionality to interface with digital audio devices for transmitting PCM audio data over a serial bus with the MCU. The SSI supports an audio clock frequency of up to 50 MHz, and can be operated as a slave or master receiver, transmitter, or transceiver to suit various applications. The SSI includes 8-stage FIFO buffers in the receiver and transmitter, and supports interrupts and DMA-driven data reception and transmission. See section 36, Serial Sound Interface (SSI).
Quad Serial Peripheral Interface (QSPI)	The QSPI is a memory controller for connecting a serial ROM (nonvolatile memory such as a serial flash memory, serial EEPROM, or serial FeRAM) that has an SPI-compatible interface. See section 34, Quad Serial Peripheral Interface (QSPI).
Controller Area Network (CAN) Module	The CAN module provides functionality to receive and transmit data using a message-based protocol between multiple slaves and masters in electromagnetically noisy applications. The CAN module complies with the ISO 11898-1 (CAN 2.0A/CAN 2.0B) standard and supports up to 32 mailboxes, which can be configured for transmission or reception in normal mailbox and FIFO modes. Both standard (11-bit) and extended (29-bit) messaging formats are supported. See section 32, Controller Area Network (CAN) Module.

Table 1.8 Communication interfaces (2 of 2)

Feature	Functional description
USB 2.0 Full-Speed Module (USBFS)	The full-speed USB controller can operate as a host controller or device controller. The module supports full-speed and low-speed (host controller only) transfer as defined in the Universal Serial Bus Specification 2.0. The module has an internal USB transceiver and supports all of the transfer types defined in the Universal Serial Bus Specification 2.0. The USB has buffer memory for data transfer, providing a maximum of 10 pipes. Pipes 1 to 9 can be assigned any endpoint number based on the peripheral devices used for communication or based on the user system. The MCU supports revision 1.2 of the battery charging specification. Because the MCU can be powered at 5 V, the USB LDO regulator provides the internal USB transceiver power supply 3.3 V. See section 28, USB 2.0 Full-Speed Module (USBFS).
SD/MMC Host Interface (SDHI)	The SDHI provides the functionality needed to connect a variety of external memory cards to the MCU. The SDHI supports both 1-bit and 4-bit buses for connecting memory cards that support SD, SDHC, and SDXC formats. When developing host devices that are compliant with the SD specifications, you must comply with the SD Host/Ancillary Product License Agreement (SD HALA). The MMC interface supports 1-bit, 4-bit, and 8-bit MMC buses that provide eMMC 4.51 (JEDEC Standard JESD 84-B451) device access. This interface also provides backward compatibility and supports high-speed SDR transfer modes. See section 37, SD/MMC Host Interface (SDHI).

Table 1.9 Analog

Feature	Functional description
14-bit A/D Converter (ADC14)	A 14-bit successive approximation A/D converter is provided. Up to 28 analog input channels are selectable. Temperature sensor output and internal reference voltage are selectable for conversion. The A/D conversion accuracy is selectable from 12-bit and 14-bit conversion making it possible to optimize the tradeoff between speed and resolution in generating a digital value. See section 39, 14-Bit A/D Converter (ADC14).
12-bit D/A Converter (DAC12)	The 12-bit D/A converts data and includes an output amplifier. See section 40, 12-Bit D/A Converter (DAC12).
Temperature Sensor (TSN)	The on-chip temperature sensor 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 linear. The output voltage is provided to the ADC14 for conversion and can be further used by the end application. See section 41, Temperature Sensor (TSN).
High-Speed Analog Comparator (ACMPHS)	ACMPHS compares the test voltage with a reference voltage and to provide a digital output based on the result of conversion. Both the test and reference voltages can be provided to the comparator from internal sources such as the DAC12 output and internal reference voltage, and an external source. Such flexibility is useful in applications that require go/no-go comparisons to be performed between analog signals without necessarily requiring A/D conversion. See section 43, High-Speed Analog Comparator (ACMPHS).
Low-Power Analog Comparator (ACMPLP)	ACMPLP compares a reference input voltage and analog input voltage. The comparison result can be read by software and also be output externally. The reference input voltage can be selected from either an input to the CMPREFi (i = 0, 1) pin or from the internal reference voltage (Vref) generated internally in the MCU. The ACMPLP response speed can be set before starting an operation. Setting the high-speed mode decreases the response delay time, but increases current consumption. Setting the low-speed mode increases the response delay time, but decreases current consumption. See section 44, Low Power Analog Comparator (ACMPLP).
Operational Amplifier (OPAMP)	OPAMP amplifies small analog input voltages and outputs the amplified voltages. A total of four differential operational amplifier units with two input pins and one output pin are provided. See section 42, Operational Amplifier (OPAMP).

Table 1.10 Human machine interfaces

Feature	Functional description
Segment LCD Controller (SLCDC)	The SLCDC provides the following functions: • Waveform A or B selectable • The LCD driver voltage generator can switch between an internal voltage boosting method, a capacitor split method, and an external resistance division method • Automatic output of segment and common signals based on automatic display data register read • The reference voltage generated when operating the voltage boost circuit can be selected in 16 steps (contrast adjustment) • The LCD can be made to blink. See section 49, Segment LCD Controller/Driver (SLCDC).
Capacitive Touch Sensing Unit (CTSU)	The Capacitive Touch Sensing Unit (CTSU) measures the electrostatic capacitance of the touch sensor. Changes in the electrostatic capacitance are determined by software, which enables the CTSU to detect whether a finger is in contact with the touch sensor. The electrode surface of the touch sensor is usually enclosed with an electrical insulator so that a finger does not come into direct contact with the electrodes. See section 45, Capacitive Touch Sensing Unit (CTSU).

Table 1.11 Data processing

Feature	Functional description
Cyclic Redundancy Check (CRC) Calculator	The CRC calculator 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 monitoring reads from and writes 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 35, Cyclic Redundancy Check (CRC) Calculator.
Data Operation Circuit (DOC)	The Data Operation Circuit (DOC) compares, adds, and subtracts 16-bit data. See section 46, Data Operation Circuit (DOC).

Table 1.12 Security

Feature	Functional description
Secure Crypto Engine 5 (SCE5)	• Security algorithm: - Symmetric algorithm: AES. • Other support features: - TRNG (True Random Number Generator) - Hash-value generation: GHASH.

1.2 Block Diagram

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

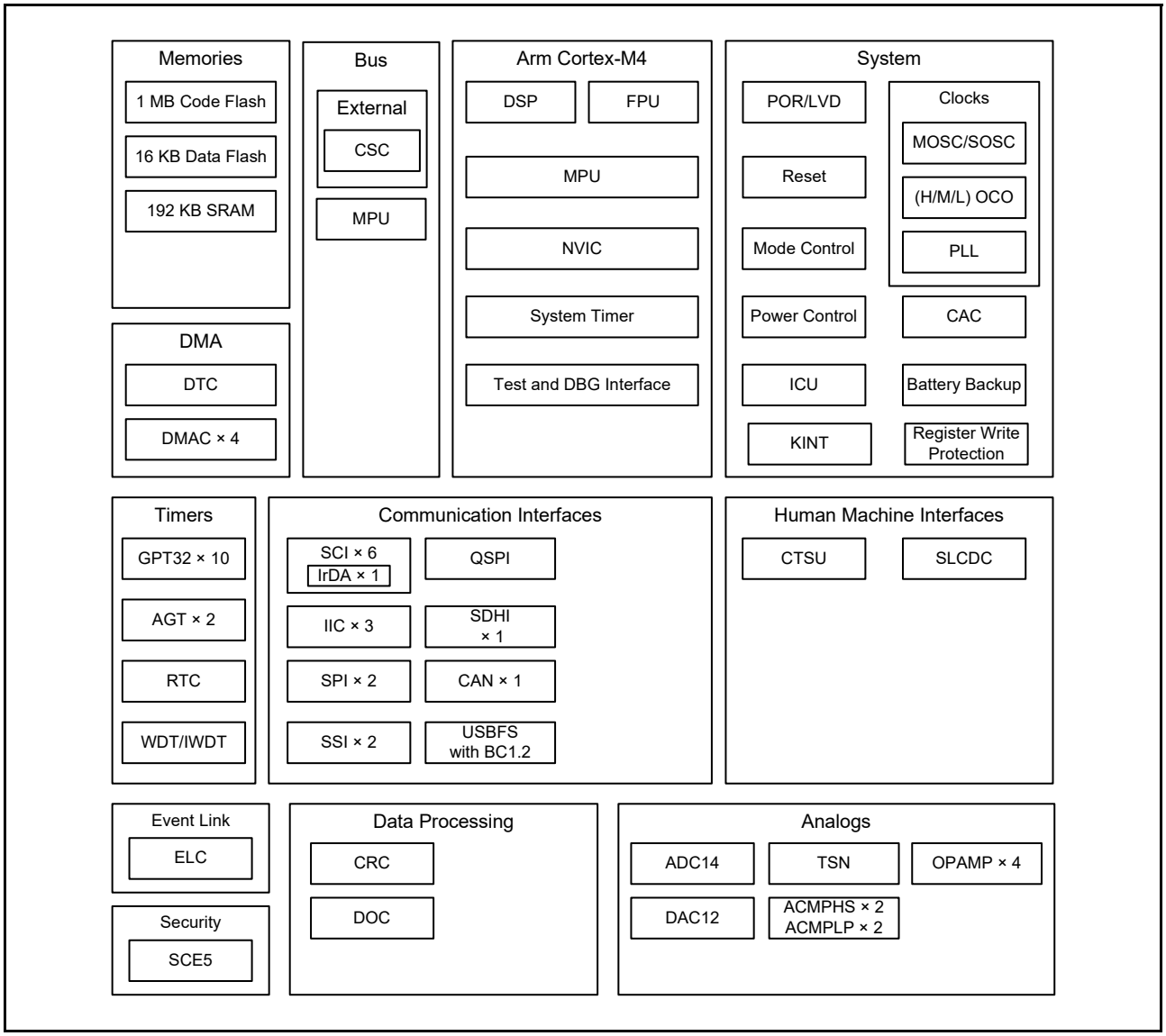


Figure 1.1 Block diagram

1.3 Part Numbering

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

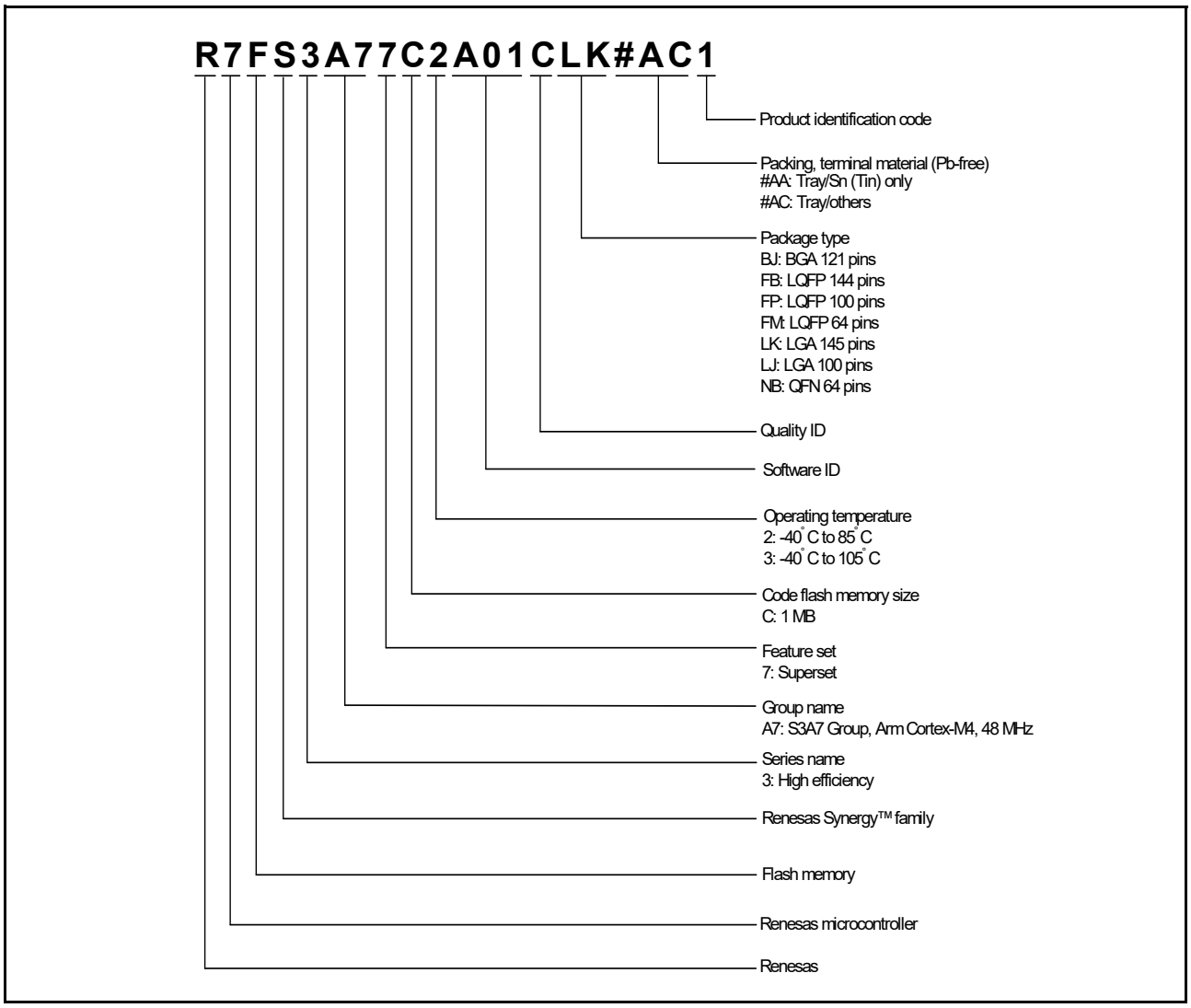


Figure 1.2 Part numbering scheme

Table 1.13 Product list

Part number	Ordering part number	Package	Code flash	Data flash	SRAM	Operating temperature
R7FS3A77C2A01CLK	R7FS3A77C2A01CLK#AC1	PTLG0145KA-A	1 MB	16 KB	192 KB	-40 to +85°C
R7FS3A77C3A01CFB	R7FS3A77C3A01CFB#AA1	PLQP0144KA-B	1 MB	16 KB	192 KB	-40 to +105°C
R7FS3A77C2A01CBJ	R7FS3A77C2A01CBJ#AC1	PLBG0121JA-A	1 MB	16 KB	192 KB	-40 to +85°C
R7FS3A77C3A01CFP	R7FS3A77C3A01CFP#AA1	PLQP0100KB-B	1 MB	16 KB	192 KB	-40 to +105°C
R7FS3A77C2A01CLJ	R7FS3A77C2A01CLJ#AC1	PTLG0100JA-A	1 MB	16 KB	192 KB	-40 to +85°C
R7FS3A77C3A01CFM	R7FS3A77C3A01CFM#AA1	PLQP0064KB-C	1 MB	16 KB	192 KB	-40 to +105°C
R7FS3A77C3A01CNB	R7FS3A77C3A01CNB#AC1	PWQN0064LA-A	1 MB	16 KB	192 KB	-40 to +105°C

1.4 Function Comparison

Table 1.14 Function comparison

Parts number		R7FS3A77C2A01CLK	R7FS3A77C3A01CFB	R7FS3A77C2A01CBJ	R7FS3A77C3A01CFP	R7FS3A77C2A01CLJ	R7FS3A77C3A01CFM R7FS3A77C3A01CNB
Pin count		145	144	121	100	100	64
Package		LGA	LQFP	BGA	LQFP	LGA	LQFP/QFN
Code flash memory		1 MB					
Data flash memory		16 KB					
SRAM		192 KB					
	Parity	176 KB					
	ECC	16 KB					
System	CPU clock	48 MHz					
	Backup registers	512 bytes					
	ICU	Yes					
	KINT	8					
Event control	ELC	Yes					
DMA	DTC	Yes					
	DMAC	4					
BUS	External bus	16-bit bus		8-bit bus			No
Timers	GPT32	10	10	10	10	10	9
	AGT	2	2	2	2	2	2
	RTC	Yes					
	WDT/IWDT	Yes					
Communication	SCI	6					
	IIC	3			2		
	SPI	2					
	SSI	2					1
	QSPI	1					No
	SDHI	1					No
	CAN	1					
	USBFS	Yes					
Analog	ADC14	28		26	25	25	18
	DAC12	2					
	ACMPHS	2					
	ACMPLP	2					
	OPAMP	4	4	4	4	4	3
	TSN	Yes					
HMI	SLCDC	4 com × 52 seg or 8 com x 48 seg		4 com × 38 seg or 8 com x 34 seg	4 com × 26 seg or 8 com x 22 seg	4 com × 26 seg or 8 com x 22 seg	No
	CTSU	31			26		14
Data processing	CRC	Yes					
	DOC	Yes					
Security		SCE5					

1.5 Pin Functions

Function	Signal	I/O	Description
Power supply	VCC	Input	Power supply pin. Connect to the system power supply. Connect this pin to VSS through a 0.1-μF capacitor. Place the capacitor close to the pin.
	VCL	Input	Connect this pin to the VSS pin through 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).
	VBATT	Input	Backup power supply pin
Clock	XTAL	Output	Pins for a crystal resonator. An external clock signal can be input through the EXTAL pin.
	EXTAL	Input	
	XCIN	Input	Input/output pins for the sub-clock oscillator. Connect a crystal resonator between XCOUT and XCIN.
	XCOUT	Output	
	EBCLK	Output	Outputs the external bus clock for external devices
	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
Interrupt	NMI	Input	Non-maskable interrupt request pin
	IRQ0 to IRQ15	Input	Maskable interrupt request pins
KINT	KR00 to KR07	Input	A key interrupt (KINT) can be generated by inputting a falling edge to the key interrupt input pins
On-chip debug	TMS	I/O	On-chip emulator or boundary scan pins
	TDI	Input	
	TCK	Input	
	TDO	Output	
	SWDIO	I/O	Serial wire debug data Input/output pin
	SWCLK	Input	Serial wire clock pin
	SWO	Output	Serial wire trace output pin
External bus interface	RD	Output	Strobe signal indicating that reading from the external bus interface space is in progress, active-low
	WR	Output	Strobe signal indicating that writing to the external bus interface space is in progress, in 1-write strobe mode, active-low
	WR0, WR1	Output	Strobe signals indicating that either group of data bus pins (D07 to D00, D15 to D08) is valid in writing to the external bus interface space, in byte strobe mode, active-low
	BC0, BC1	Output	Strobe signals indicating that either group of data bus pins (D07 to D00, D15 to D08) is valid in access to the external bus interface space, in 1-write strobe mode, active-low
	WAIT	Input	Input pin for wait request signals in access to the external space, active-low
	CS0 to CS3	Output	Select signals for CS areas, active-low
	A00 to A16	Output	Address bus
	D00 to D15	I/O	Data bus
Battery Backup	VBATWIO0 to VBATWIO2	I/O	Output wakeup signal for the VBATT wakeup control function. External event input for the VBATT wakeup control function.

Function	Signal	I/O	Description
GPT	GTETRGA, GTETRGB, GTETRGC, GTETRGD	Input	External trigger input pins
	GTIOC0A to GTIOC9A, GTIOC0B to GTIOC9B	I/O	Input capture, output capture, or PWM output pins
	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)
AGT	AGTEE0, AGTEE1	Input	External event input enable signals
	AGTIO0, AGTIO1	I/O	External event input and pulse output pins
	AGTO0, AGTO1	Output	Pulse output pins
	AGTOA0, AGTOA1	Output	Output compare match A output pins
	AGTOB0, AGTOB1	Output	Output compare match B output pins
RTC	RTCOUT	Output	Output pin for 1-Hz/64-Hz clock
	RTCIC0 to RTCIC2	Input	Time capture event input pins
SCI	SCK0 to SCK4, SCK9	I/O	Input/output pins for the clock (clock synchronous mode)
	RXD0 to RXD4, RXD9	Input	Input pins for received data (asynchronous mode/clock synchronous mode)
	TXD0 to TXD4, TXD9	Output	Output pins for transmitted data (asynchronous mode/clock synchronous mode)
	CTS0_RTS0 to CTS4_RTS4, CTS9_RTS9	I/O	Input/output pins for controlling the start of transmission and reception (asynchronous mode/clock synchronous mode), active-low
	SCL0 to SCL4, SCL9	I/O	Input/output pins for the I^2C clock (simple IIC)
	SDA0 to SDA4, SDA9	I/O	Input/output pins for the I^2C data (simple IIC)
	SCK0 to SCK4, SCK9	I/O	Input/output pins for the clock (simple SPI)
	MISO0 to MISO4, MISO9	I/O	Input/output pins for slave transmission of data (simple SPI)
	MOSI0 to MOSI4, MOSI9	I/O	Input/output pins for master transmission of data (simple SPI)
	SS0 to SS4, SS9	Input	Chip-select input pins (simple SPI), active-low
IIC	SCL0 to SCL2	I/O	Input/output pins for the clock
	SDA0 to SDA2	I/O	Input/output pins for the data
SSI	SSISCK0	I/O	SSI serial bit clock pins
	SSISCK1		
	SSIWS0	I/O	Word select pins
	SSIWS1		
	SSITXD0	Output	Serial data output pin
	SSIRXD0	Input	Serial data input pin
	SSIDATA1	I/O	Serial data input/output pin
	AUDIO_CLK	Input	External clock pin for audio (input oversampling clock)

Function	Signal	I/O	Description
SPI	RSPCKA, RSPCKB	I/O	Clock input/output pin
	MOSIA, MOSIB	I/O	Input or output pins for data output from the master
	MISOA, MISOB	I/O	Input or output pins for data output from the slave
	SSLA0, SSLB0	I/O	Input or output pins for slave selection
	SSLA1, SSLA2, SSLA3, SSLB1, SSLB2, SSLB3	Output	Output pins for slave selection
QSPI	QSPCLK	Output	QSPI clock output pin
	QSSL	Output	QSPI slave output pin
	QIO0	I/O	Master transmit data/Data 0
	QIO1	I/O	Master input data/Data 1
	QIO2, QIO3	I/O	Data 2, Data 3
CAN	CRX0	Input	Receive data
	CTX0	Output	Transmit data
USBFS	VSS_USB	Input	Ground pins
	VCC_USB_LDO	Input	Power supply pin for USB LDO regulator
	VCC_USB	I/O	Input: Power supply pin for USB transceiver. Output: USB LDO regulator output pin. This pin should be connected to an external capacitor.
	USB_DP	I/O	D+ I/O pin of the USB on-chip transceiver. Connect this pin to the D+ pin of the USB bus.
	USB_DM	I/O	D- I/O pin of the USB on-chip transceiver. Connect this pin to the D- pin of the USB bus.
	USB_VBUS	Input	USB cable connection monitor pin. Connect this pin to VBUS of the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a device controller.
	USB_EXICEN	Output	Low power control signal for external power supply (OTG) chip
	USB_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip
	USB_OVRCURA, USB_OVRCURB	Input	Connect the external overcurrent detection signals to these pins. Connect the VBUS comparator signals to these pins when the OTG power supply chip is connected.
	USB_ID	Input	Connect the MicroAB connector ID input signal to this pin during operation in OTG mode
SDHI	SD0CLK	Output	SD clock output pin
	SD0CMD	I/O	Command output pin and response input signal pin
	SD0DAT0 to SD0DAT7	I/O	SD and MMC data bus pins
	SD0WP	Input	SD write-protect signal
Analog power supply	AVCC0	Input	Analog voltage supply pin
	AVSS0	Input	Analog voltage supply ground pin
	VREFH0	Input	Analog reference voltage supply pin
	VREFL0	Input	Reference power supply ground pin
	VREFH	Input	Analog reference voltage supply pin for DAC12
	VREFL	Input	Analog reference ground pin for DAC12
ADC14	AN000 to AN027	Input	Input pins for the analog signals to be processed by the ADC14
	ADTRG0	Input	Input pins for the external trigger signals that start the A/D conversion, active-low
DAC12	DA0, DA1	Output	Output pins for the analog signals to be processed by the D/A converter
Comparator output	VCOUT	Output	Comparator output pin
ACMPHS	IVREF0 to IVREF5	Input	Reference voltage input pin
	IVCMP0 to IVCMP5	Input	Analog voltage input pins
ACMPLP	CMPREF0, CMPREF1	Input	Reference voltage input pins
	CMPIN0, CMPIN1	Input	Analog voltage input pins
OPAMP	AMP0+ to AMP3+	Input	Analog voltage input pins
	AMP0- to AMP3-	Input	Analog voltage input pins
	AMP0O to AMP3O	Output	Analog voltage output pins

Function	Signal	I/O	Description
CTSU	TS00, TS01, TS03 to TS22, TS26 to TS27, TS29 to TS35	Input	Capacitive touch detection pins (touch pins)
	TSCAP	-	Secondary power supply pin for the touch driver
I/O ports	P000 to P015	I/O	General-purpose input/output pins
	P100 to P115	I/O	General-purpose input/output pins
	P200	Input	General-purpose input pin
	P201 to P206, P212, P213	I/O	General-purpose input/output pins
	P214, P215	Input	General-purpose input pins
	P300 to P315	I/O	General-purpose input/output pins
	P400 to P415	I/O	General-purpose input/output pins
	P500 to P507, P511, P512	I/O	General-purpose input/output pins
	P600 to P606, P608 to P614	I/O	General-purpose input/output pins
	P700 to P705, P708 to P713	I/O	General-purpose input/output pins
	P800 to P809	I/O	General-purpose input/output pins
	P900 to P902	I/O	General-purpose input/output pins
SLCDC	VL1, VL2, VL3, VL4	I/O	Voltage pin for driving the LCD
	CAPH, CAPL	I/O	Capacitor connection pin for the LCD controller/driver
	COM0 to COM7	Output	Common signal output pins for the LCD controller/driver
	SEG00 to SEG51	Output	Segment signal output pins for the LCD controller/driver

1.6 Pin Assignments

Figure 1.3 to Figure 1.9 show the pin assignments.

R7FS3A77C2A01CLK

	A	B	C	D	E	F	G	H	J	K	L	M	N	
13	P407	P409	P412	P708	P711	VCC	P212 /EXTAL	P215 /XCIN	VCL	P702	P405	P402	P400	13
12	USB_DM	USB_DP	P410	P414	P710	VSS	P213 /XTAL	P214 /XCOUT	VBATT	P701	P404	P511	VCC	12
11	VCC_ USB	VSS_ USB	VCC_ USB_LDO	P411	P415	P712	P705	P704	P703	P403	P401	P512	VSS	11
10	P205	P206	P204	P408	P413	P709	P713	P700	P406	P003	P000	P002	P001	10
9	P203	P313	P202	P314						P004	P006	P009	P008	9
8	P900	P901	P200	P315						P005	AVSS0	P011 /VREFL0	P010 /VREFH0	8
7	VSS	P902	RES	P310						P007	AVCC0	P013 /VREFL	P012 /VREFH	7
6	VCC	P201/MD	P312	P305						P505	P506	P015	P014	6
5	P309	P311	P308	P303	NC					P503	P504	VSS	VCC	5
4	P307	P306	P304	P109/TDO /SWO	P114	P608	P604	P600	P105	P500	P502	P501	P507	4
3	P808	P809	P301	P112	P115	P610	P614	P603	P107	P106	P104	P803	P802	3
2	P302	P300/TCK /SWCLK	P111	P806	P609	P612	VSS	P605	P601	P805	P800	P101	P801	2
1	P108/TMS /SWDIO	P110/TDI	P113	P807	P611	P613	VCC	P606	P602	P804	P103	P102	P100	1
	A	B	C	D	E	F	G	H	J	K	L	M	N	

Figure 1.3 Pin assignment for 145-pin LGA (top view)

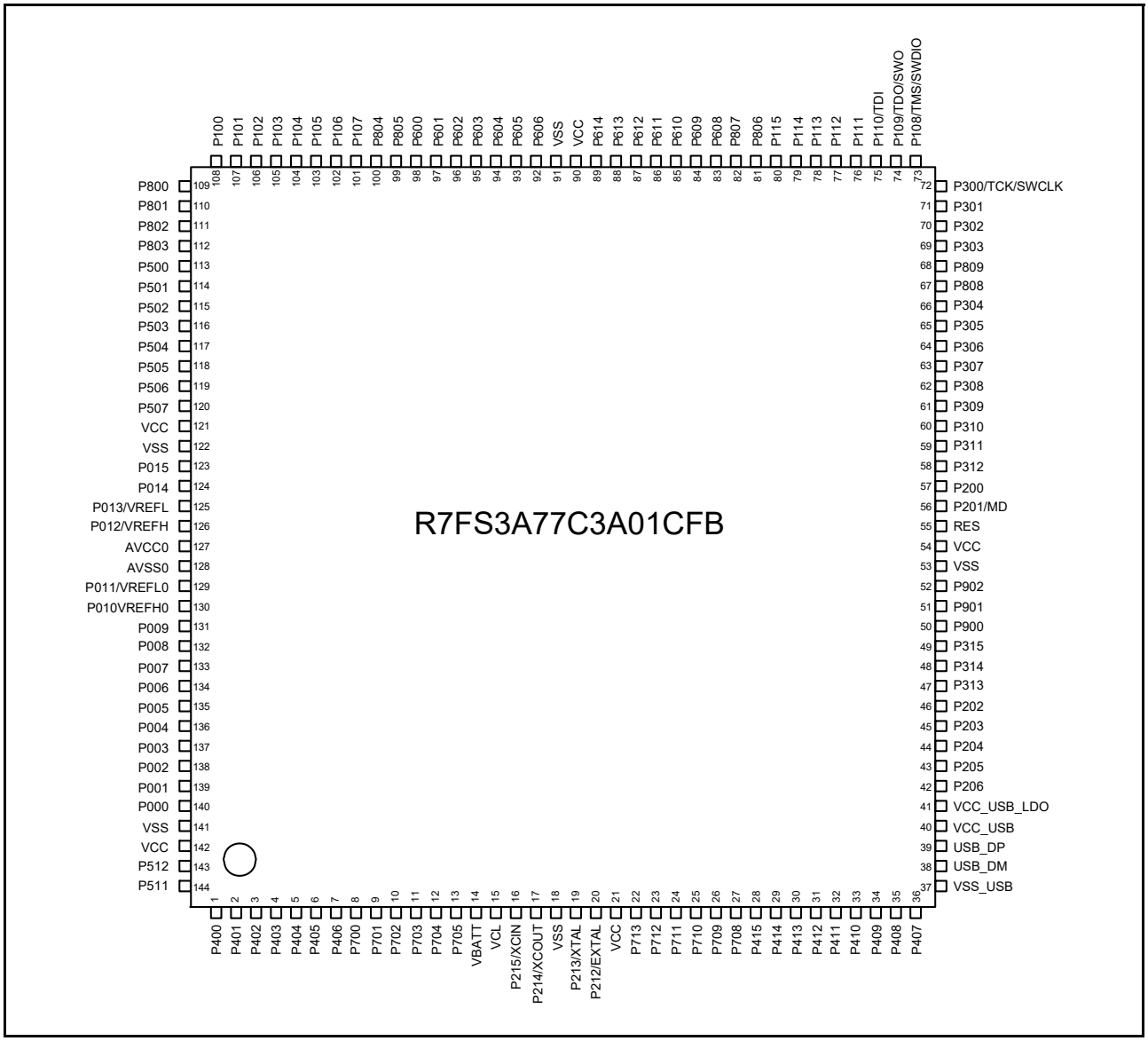


Figure 1.4 Pin assignment for 144-pin LQFP (top view)

R7FS3A77C2A01CBJ

	A	B	C	D	E	F	G	H	J	K	L	
11	P407	P408	P411	P414	P212/ EXTAL	P215/ XCIN	VCL	P406	P403	P401	P400	11
10	USB_DM	USB_DP	P410	P415	P213/ XTAL	P214/ XCOUT	VBATT	P405	P402	P511	P512	10
9	VCC_ USB	VSS_ USB	P409	P412	P708	VCC	VSS	P404	P002	P001	P000	9
8	P205	VCC_ USB_ LDO	P206	P204	P413	P710	P702	P006	P004	P003	P005	8
7	P203	P202	P313	P314	P315	P709	P701	P007	AVSS0	P011/ VREFL0	P010/ VREFH0	7
6	VSS	VCC	RES	P201/MD	P200	NC	P700	P008	AVCC0	P013/ VREFL	P012/ VREFH	6
5	P308	P309	P307	P302	P304	P612	P601	P506	P505	P015	P014	5
4	P305	P306	P808	P114	P611	P603	P600	P504	P503	VSS	VCC	4
3	P809	P303	P110/TDI	P111	P609	P604	P106	P104	P502	P500	P501	3
2	P301	P108/ TMS/ SWDIO	P113	P608	P613	P605	P602	P105	P102	P801	P800	2
1	P300/ TCK/ SWCLK	P109/ TDO/ SWO	P112	P115	P610	VCC	VSS	P107	P103	P101	P100	1
	A	B	C	D	E	F	G	H	J	K	L	

Figure 1.5 Pin assignment for 121-pin BGA (top view)

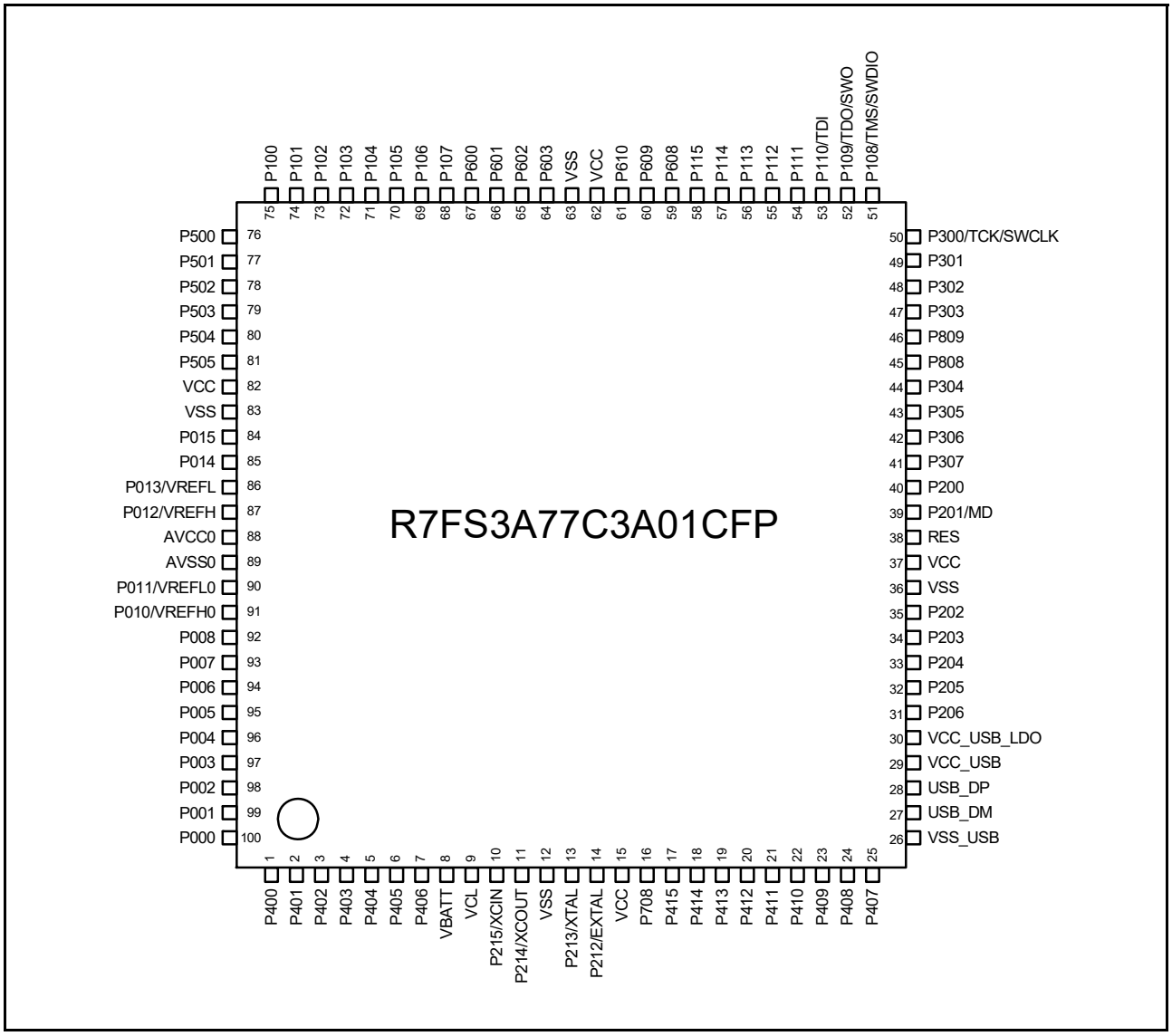


Figure 1.6 Pin assignment for 100-pin LQFP (top view)

R7FS3A77C2A01CLJ

	A	B	C	D	E	F	G	H	J	K	
10	P407	P409	P412	VCC	P212/ EXTAL	P215/ XCIN	VCL	P403	P400	P000	10
9	USB_DM	USB_DP	P413	VSS	P213/ XTAL	P214/ XCOUT	VBATT	P405	P401	P001	9
8	VCC_ USB	VSS_ USB	VCC_US B_LDO	P411	P415	P708	P404	P003	P004	P002	8
7	P205	P204	P206	P408	P414	P406	P006	P007	P008	P005	7
6	VSS	VCC	P202	P203	P410	P402	P505	AVSS0	P011/ VREFL0	P010/ VREFH0	6
5	P200	P201/MD	P307	RES	P113	P600	P504	AVCC0	P013/ VREFL	P012/ VREFH	5
4	P305	P304	P808	P306	P115	P601	P503	P100	P015	P014	4
3	P809	P303	P110/TDI	P111	P609	P602	P107	P103	VSS	VCC	3
2	P300/ TCK/ SWCLK	P302	P301	P114	P610	P603	P106	P101	P501	P502	2
1	P108/ TMS/ SWDIO	P109/ TDO/ SWO	P112	P608	VCC	VSS	P105	P104	P102	P500	1
	A	B	C	D	E	F	G	H	J	K	

Figure 1.7 Pin assignment for 100-pin LGA (top view)

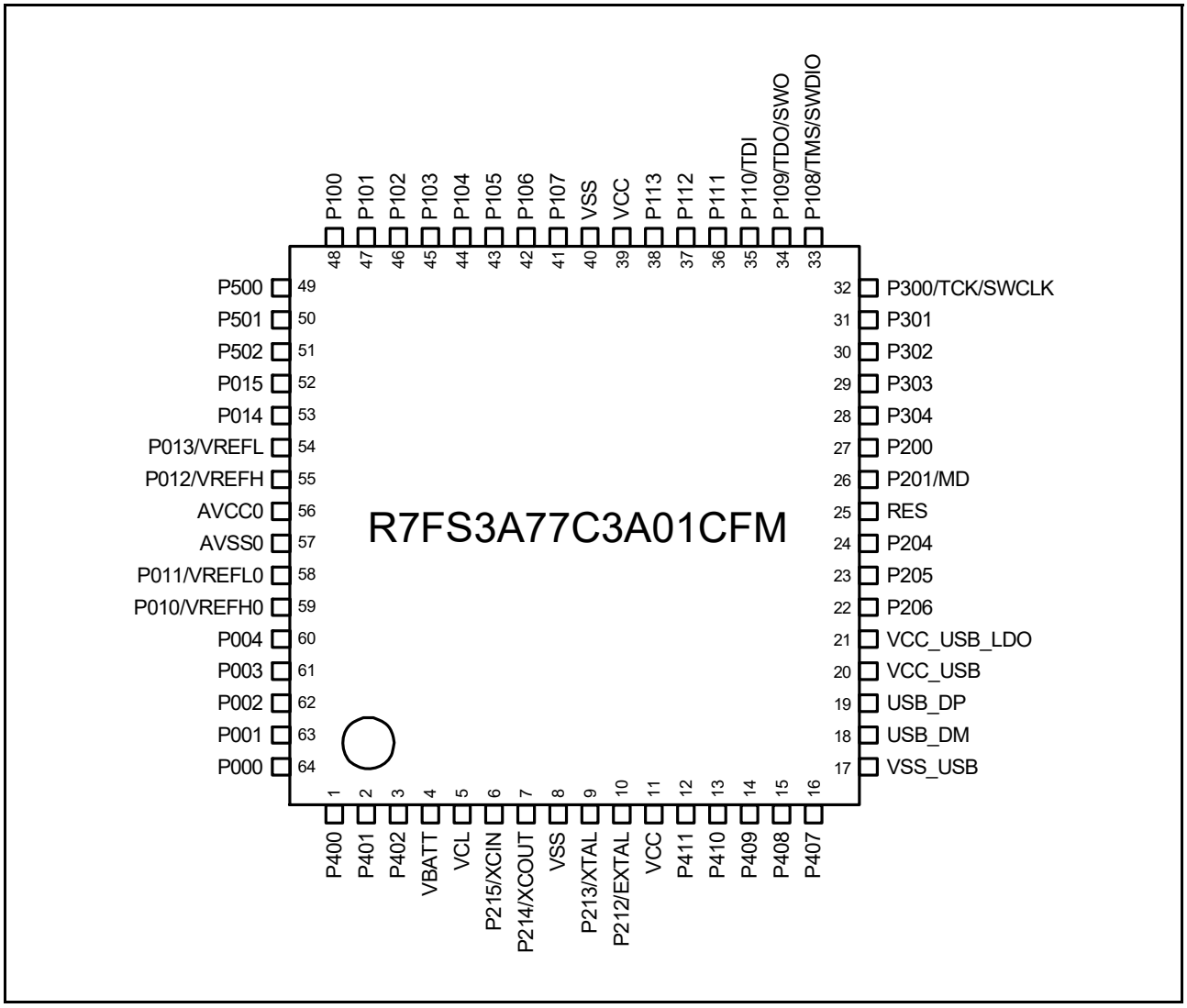


Figure 1.8 Pin assignment for 64-pin LQFP (top view)

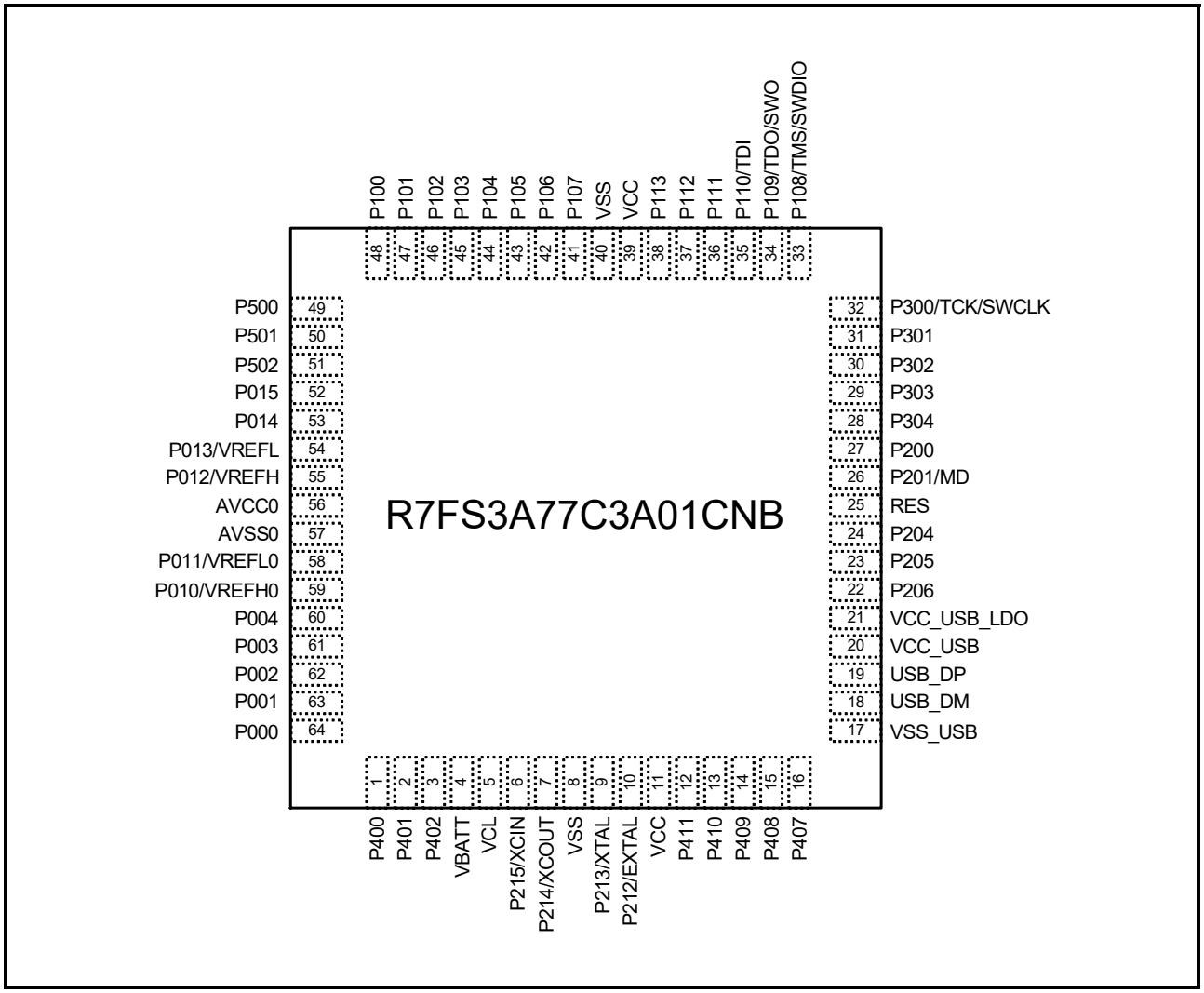


Figure 1.9 Pin assignment for 64-pin QFN (top view)

1.7 Pin Lists

Pin number											Timers				Communication interfaces						Analogs			HMI	
LGA145	LQFP144	BGA121	LQFP100	LGA100	LQFP64	QFN64	Power, System, Clock, Debug, CAC, VBATT	Interrupt	I/O ports	External bus	AGT	GPT_OPS, POEG	GPT	RTC	USBFS,CAN	SCI	IIC	SPI/QSPI	SSI	SDHI	ADC14	DAC12, OPAMP	ACMPHS, ACMPLP	SLCDC	CTSU
N13	1	L11	1	J10	1	1		IRQ0	P400				GTIOC6A_A			SCK4_B	SCL0_A		AUDIO_CLK						TS20
L11	2	K11	2	J9	2	2		IRQ5	P401			GTETRGA_B	GTIOC6B_A		CTX0_B	CTS4_RTS4_B/SS4_B	SDA0_A								TS19
M13	3	J10	3	F6	3	3	VBATWIO0	IRQ4	P402		AGTIO0_B/AGTIO1_B			RTCIC0	CRX0_B										TS18
K11	4	J11	4	H10			VBATWIO1		P403		AGTIO0_C/AGTIO1_C		GTIOC3A_B	RTCIC1					SSISCK0_A						TS17
L12	5	H9	5	G8			VBATWIO2		P404				GTIOC3B_B	RTCIC2					SSIWS0_A						TS16
L13	6	H10	6	H9					P405				GTIOC1A_B						SSITXD0_A						TS15
J10	7	H11	7	F7					P406				GTIOC1B_B						SSIRXD0_A						TS14
H10	8	G6							P700				GTIOC5A_B												TS32
K12	9	G7							P701				GTIOC5B_B												TS33
K13	10	G8							P702				GTIOC6A_B												TS34
J11	11								P703				GTIOC6B_B												
H11	12								P704																
G11	13								P705																
J12	14	G10	8	G9	4	4	VBATT																		
J13	15	G11	9	G10	5	5	VCL																		
H13	16	F11	10	F10	6	6	XCIN		P215																
H12	17	F10	11	F9	7	7	XCOUT		P214																
F12	18	G9	12	D9	8	8	VSS																		
G12	19	E10	13	E9	9	9	XTAL	IRQ2	P213			GTETRGC_A				TXD1_A/MOSI1_A/SDA1_A									
G13	20	E11	14	E10	10	10	EXTAL	IRQ3	P212		AGTEE1	GTETRGD_A				RXD1_A/MISO1_A/SCL1_A									
F13	21	F9	15	D10	11	11	VCC																		
G10	22								P713				GTIOC2A_B												
F11	23								P712				GTIOC2B_B												
E13	24								P711							CTS1_RTS1_B/SS1_B									
E12	25	F8							P710							SCK1_B									TS35
F10	26	F7						IRQ10	P709							TXD1_B/MOSI1_B/SDA1_B									TS13
D13	27	E9	16	F8			CACREF_B	IRQ11	P708							RXD1_B/MISO1_B/SCL1_B		SSLA3_B							TS12
E11	28	D10	17	E8					P415									SSLA2_B							TS11
D12	29	D11	18	E7					P414									SSLA1_B		SD0WP					TS10
E10	30	E8	19	C9					P413			GTOUUP_B				CTS0_RTS0_B/SS0_B		SSLA0_B		SD0CLK					TS09
C13	31	D9	20	C10					P412			GTOULO_B				SCK0_B		RSPCKA_B		SD0CMD					TS08

Pin number							Power, System, Clock, Debug, CAC, VBATT	Interrupt	I/O ports	External bus	Timers				Communication interfaces						Analogs			HMI	
LGA145	LQFP144	BGA121	LQFP100	LGA100	LQFP64	QFN64					AGT	GPT_OPS, POEG	GPT	RTC	USBFS,CAN	SCI	IIC	SPI/QSPI	SSI	SDHI	ADC14	DAC12, OPAMP	ACMPHS, ACMPLP	SLCDC	CTSU
D11	32	C11	21	D8	12	12		IRQ4	P411		AGTOA1	GTOVUP_B	GTIOC9A_A			TXD0_B/MOSI0_B/SDA0_B/CTS3_RTS3_A/SS3_A		MOSIA_B		SD0DAT0					TS07
C12	33	C10	22	E6	13	13		IRQ5	P410		AGTOB1	GTOVLO_B	GTIOC9B_A			RXD0_B/MISO0_B/SCL0_B/SCK3_A		MISOA_B		SD0DAT1					TS06
B13	34	C9	23	B10	14	14		IRQ6	P409			GTOWUP_B			USB_EXICEN_A	TXD3_A/MOSI3_A/SDA3_A									TS05
D10	35	B11	24	D7	15	15		IRQ7	P408			GTOWLO_B			USB_ID_A	RXD3_A/MISO3_A/SCL3_A									TS04
A13	36	A11	25	A10	16	16			P407					RTCOUT	USB_VBUS	CTS4_RTS4_A/SS4_A	SDA0_B	SSLB3_A			ADTRG0_B				TS03
B11	37	B9	26	B8	17	17	VSS_USB																		
A12	38	A10	27	A9	18	18									USB_DM										
B12	39	B10	28	B9	19	19									USB_DP										
A11	40	A9	29	A8	20	20	VCC_USB																		
C11	41	B8	30	C8	21	21	VCC_USB_LDO																		
B10	42	C8	31	C7	22	22		IRQ0	P206	WAIT		GTIU_A			USB_VBUSEN_A	RXD4_A/MISO4_A/SCL4_A	SDA1_A	SSLB1_A	SSIDATA1_A	SD0DAT2					TS01
A10	43	A8	32	A7	23	23	CLKOUT_A	IRQ1	P205	A16	AGTO1	GTIV_A	GTIOC4A_B		USB_OVRCURA	TXD4_A/MOSI4_A/SDA4_A/CTS9_RTS9_A/SS9_A	SCL1_A	SSLB0_A	SSIWS1_A	SD0DAT3					TSCAP_A
C10	44	D8	33	B7	24	24	CACREF_A		P204		AGTIO1_A	GTIW_A	GTIOC4B_B		USB_OVRCURB	SCK4_A/SCK9_A	SCL0_B	RSPCKB_A	SSISCK1_A	SD0DAT4				SEG23	TS00
A9	45	A7	34	D6				IRQ2	P203				GTIOC5A_A		CTX0_A	CTS2_RTS2_A/SS2_A/TXD9_A/MOSI9_A/SDA9_A		MOSIB_A		SD0DAT5				SEG22	TSCAP_B
C9	46	B7	35	C6				IRQ3	P202	WR1/BC1			GTIOC5B_A		CRX0_A	SCK2_A/RXD9_A/MISO9_A/SCL9_A		MISOB_A		SD0DAT6				SEG21	
B9	47	C7							P313											SD0DAT7				SEG20	
D9	48	D7							P314															SEG4	
D8	49	E7							P315															SEG5	
A8	50								P900															SEG6	
B8	51								P901															SEG7	
B7	52								P902															SEG8	
A7	53	A6	36	A6			VSS																		
A6	54	B6	37	B6			VCC																		
C7	55	C6	38	D5	25	25	RES																		
B6	56	D6	39	B5	26	26	MD		P201																
C8	57	E6	40	A5	27	27		NMI	P200																
C6	58								P312	CS3														SEG9	

Pin number											Timers				Communication interfaces						Analogs			HMI	
LGA145	LQFP144	BGA121	LQFP100	LGA100	LQFP64	QFN64	Power, System, Clock, Debug, CAC, VBATT	Interrupt	I/O ports	External bus	AGT	GPT_OPS, POEG	GPT	RTC	USBFS,CAN	SCI	IIC	SPI/QSPI	SSI	SDHI	ADC14	DAC12, OPAMP	ACMPHS, ACMPLP	SLCDC	CTSU
B5	59								P311	CS2														SEG10	
D7	60								P310	A15														SEG11	
A5	61	B5							P309	A14														SEG12	
C5	62	A5							P308	A13														SEG13	
A4	63	C5	41	C5					P307	A12														SEG14	
B4	64	B4	42	D4					P306	A11														SEG15	
D6	65	A4	43	A4				IRQ8	P305	A10														SEG16	
C4	66	E5	44	B4	28	28		IRQ9	P304	A09			GTIOC 7A_A											SEG17	
A3	67	C4	45	C4					P808															SEG18	
B3	68	A3	46	A3					P809															SEG19	
D5	69	B3	47	B3	29	29			P303	A08			GTIOC 7B_A											SEG3/ COM7	
A2	70	D5	48	B2	30	30		IRQ5	P302	A07		GTOU UP_A	GTIOC 4A_A			TXD2_ A/ MOSI2 _A/ SDA2_ A		SSLB3 _B						SEG2/ COM6	
C3	71	A2	49	C2	31	31		IRQ6	P301	A06		GTOU LO_A	GTIOC 4B_A			RXD2_ A/ MISO2 _A/ SCL2_ A		SSLB2 _B						SEG1/ COM5	
B2	72	A1	50	A2	32	32	TCK/ SWCL K		P300				GTIOC 0A_A					SSLB1 _B							
A1	73	B2	51	A1	33	33	TMS/ SWDI O		P108				GTIOC 0B_A			CTS9_ RTS9_ B/ SS9_B		SSLB0 _B							
D4	74	B1	52	B1	34	34	TDO/ SWO/ CLKO UT_B		P109			GTOV UP_A	GTIOC 1A_A			TXD9_ B/ MOSI9 _B/ SDA9_ B		MOSIB _B							
B1	75	C3	53	C3	35	35	TDI	IRQ3	P110			GTOV LO_A	GTIOC 1B_A			CTS2_ RTS2_ B/ SS2_B / RXD9_ B/ MISO9 _B/ SCL9_ B		MISOB _B					VCOU T		
C2	76	D3	54	D3	36	36		IRQ4	P111	A05			GTIOC 3A_A			SCK2_ B/ SCK9_ B		RSPC KB_B						CAPH	
D3	77	C1	55	C1	37	37			P112	A04			GTIOC 3B_A			TXD2_ B/ MOSI2 _B/ SDA2_ B			SSISC K0_B					CAPL	
C1	78	C2	56	E5	38	38			P113	A03						RXD2_ B/ MISO2 _B/ SCL2_ B			SSIWS 0_B					SEG0/ COM4	
E4	79	D4	57	D2					P114	A02									SSIRX D0_B					SEG24	
E3	80	D1	58	E4					P115	A01									SSITX D0_B					SEG25	
D2	81								P806															SEG26	
D1	82								P807															SEG27	
F4	83	D2	59	D1					P608	A00/ BC0														SEG28	
E2	84	E3	60	E3					P609	CS1														SEG29	
F3	85	E1	61	E2					P610	CS0														SEG30	
E1	86	E4							P611															SEG31	
F2	87	F5							P612	D08														SEG32	
F1	88	E2							P613	D09														SEG33	
G3	89								P614	D10														SEG34	
G1	90	F1	62	E1	39	39	VCC																		
G2	91	G1	63	F1	40	40	VSS																		
H1	92								P606															SEG35	
H2	93	F2							P605	D11														SEG36	
G4	94	F3							P604	D12														SEG37	
H3	95	F4	64	F2					P603	D13														SEG38	
J1	96	G2	65	F3					P602	EBCLK														SEG39	
J2	97	G5	66	F4					P601	WR/ WR0														SEG40	
H4	98	G4	67	F5					P600	RD														SEG41	
K2	99								P805															SEG42	
K1	100								P804															SEG43	

Pin number							Power, System, Clock, Debug, CAC, VBATT	Interrupt	I/O ports	External bus	Timers				Communication interfaces						Analogs			HMI	
LGA145	LQFP144	BGA121	LQFP100	LGA100	LQFP64	QFN64					AGT	GPT_OPS, POEG	GPT	RTC	USBFS,CAN	SCI	IIC	SPI/QSPI	SSI	SDHI	ADC14	DAC12, OPAMP	ACMPHS, ACMPLP	SLCDC	CTSU
J3	101	H1	68	G3	41	41		KR07	P107	D07			GTIOC 8A_A											COM3	
K3	102	G3	69	G2	42	42		KR06	P106	D06			GTIOC 8B_A					SSLA3 _A						COM2	
J4	103	H2	70	G1	43	43		KR05/ IRQ0	P105	D05		GTET RGA_ C						SSLA2 _A						COM1	
L3	104	H3	71	H1	44	44		KR04/ IRQ1	P104	D04		GTET RGB_ B						SSLA1 _A						COM0	
L1	105	J1	72	H3	45	45		KR03	P103	D03		GTOW UP_A	GTIOC 2A_A			CTS0_ RTS0_ A/ SS0_A		SSLA0 _A			AN024		CMPR EF1	VL4	
M1	106	J2	73	J1	46	46		KR02	P102	D02	AGTO 0	GTOW LO_A	GTIOC 2B_A			SCK0_ A		RSPC KA_A			AN025 / ADTR G0_A		CMPIN 1	VL3	
M2	107	K1	74	H2	47	47		KR01/ IRQ1	P101	D01	AGTE E0	GTET RGB_ A				TXD0_ A/ MOSI0 _A/ SDA0_ A/ CTS1_ RTS1_ A/ SS1_A	SDA1_ B	MOSIA _A			AN026		CMPR EF0	VL2	
N1	108	L1	75	H4	48	48		KR00/ IRQ2	P100	D00	AGTIO 0_A	GTET RGA_ A				RXD0_ A/ MISO0 _A/ SCL0_ A/ SCK1_ A	SCL1_ B	MISOA _A			AN027		CMPIN 0	VL1	
L2	109	L2							P800	D14														SEG44	
N2	110	K2							P801	D15														SEG45	
N3	111								P802															SEG46	
M3	112								P803															SEG47	
K4	113	K3	76	K1	49	49			P500		AGTO A0	GTIU_ B			USB_V BUSE N_B			QSPC LK			AN016			SEG48	
M4	114	L3	77	J2	50	50		IRQ11	P501		AGTO B0	GTIV_ B			USB_ OVRC URA			QSSL			AN017			SEG49	
L4	115	J3	78	K2	51	51		IRQ12	P502			GTIW_ B			USB_ OVRC URB			QIO0			AN018			SEG50	
K5	116	J4	79	G4					P503			GTET RGC_ B			USB_E XICEN _B			QIO1			AN019			SEG51	
L5	117	H4	80	G5					P504			GTET RGD_ B			USB_I D_B			QIO2			AN020				
K6	118	J5	81	G6				IRQ14	P505									QIO3			AN021				
L6	119	H5						IRQ15	P506												AN022				
N4	120								P507												AN023				
N5	121	L4	82	K3			VCC																		
M5	122	K4	83	J3			VSS																		
M6	123	K5	84	J4	52	52		IRQ13	P015												AN015	DA1	IVCMP 5/ IVCMP 2		
N6	124	L5	85	K4	53	53			P014												AN014	DA0	IVREF 5/ IVREF 2		
M7	125	K6	86	J5	54	54	VREFL		P013												AN013	AMP1+			
N7	126	L6	87	K5	55	55	VREF H		P012												AN012	AMP1-			
L7	127	J6	88	H5	56	56	AVCC0																		
L8	128	J7	89	H6	57	57	AVSS0																		
M8	129	K7	90	J6	58	58	VREFL 0	IRQ15	P011												AN011	AMP2+			TS31
N8	130	L7	91	K6	59	59	VREF H0	IRQ14	P010												AN010	AMP2-			TS30
M9	131							IRQ13	P009												AN009				
N9	132	H6	92	J7				IRQ12	P008												AN008				TS29
K7	133	H7	93	H7					P007												AN007	AMP3 O	IVCMP 4/ IVCMP 1		
L9	134	H8	94	G7				IRQ11	P006												AN006	AMP3-	IVREF 4/ IVREF 1		TS27
K8	135	L8	95	K7				IRQ10	P005												AN005	AMP3+	IVREF 0		TS26
K9	136	J8	96	J8	60	60		IRQ9	P004												AN004	AMP2 O	IVCMP 0		

Pin number							Power, System, Clock, Debug, CAC, VBATT	Interrupt	I/O ports	External bus	Timers				Communication interfaces						Analogs			HMI	
LGA145	LQFP144	BGA121	LQFP100	LGA100	LQFP64	QFN64					AGT	GPT_OPS, POEG	GPT	RTC	USBFS,CAN	SCI	IIC	SPI/QSPI	SSI	SDHI	ADC14	DAC12, OPAMP	ACMPHS, ACMPLP	SLCDC	CTSU
K10	137	K8	97	H8	61	61			P003												AN003	AMP1O	IVREF3/ IVCMP3		
M10	138	J9	98	K8	62	62		IRQ8	P002												AN002	AMP0O	IVREF2/ IVCMP2		
N10	139	K9	99	K9	63	63		IRQ7	P001												AN001	AMP0-	IVREF1/ IVCMP1		TS22
L10	140	L9	100	K10	64	64		IRQ6	P000												AN000	AMP0+	IVREF0/ IVCMP0		TS21
N11	141						VSS																		
N12	142						VCC																		
M11	143	L10						IRQ14	P512				GTIOC0A_B			TXD4_B/ MOSI4_B/ SDA4_B	SCL2								
M12	144	K10						IRQ15	P511				GTIOC0B_B			RXD4_B/ MISO4_B/ SCL4_B	SDA2								
E5		F6					NC																		

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

2. CPU

The MCU is based on the Arm® Cortex®-M4 core.

2.1 Overview

2.1.1 CPU

- Arm Cortex-M4
  - Revision: r0p1-01rel0
  - Armv7E-M architecture profile
  - Single precision floating point unit compliant with the ANSI/IEEE Std 754-2008.
- Memory Protection Unit (MPU)
  - Armv7 Protected Memory System Architecture
  - 8 protected regions.
- SysTick timer
  - Driven by SYSTICCLK (LOCO) or ICLK.

See reference 1. and 2. for details.

2.1.2 Debug

- Arm CoreSight™ ETM™-M4
  - Revision: r0p1-00rel0
  - Arm ETM Architecture version 3.5.
- CoreSight Instrumentation Trace Macrocell (ITM)
- Data Watchpoint and Trace Unit (DWT)
  - 4 comparators for watchpoints and triggers.
- Flash Patch and Breakpoint Unit (FPB)
  - Flash Patch (Remap) function is unavailable, only Breakpoint function is available
  - 6 instruction comparators
  - 2 literal comparators.
- CoreSight Time Stamp Generator (TSG)
  - Time stamp for ETM and ITM
  - Driven by CPU clock.
- Debug Register Module (DBGREG)
  - Reset control
  - Halt control.
- CoreSight Debug Access Port (DAP)
  - JTAG Debug Port (JTAG-DP)
  - Serial Wire Debug Port (SW-DP).
- Cortex-M4 Trace Port Interface Unit (TPIU)
  - Serial Wire Output (SWO).
- CoreSight Embedded Trace Buffer (ETB)
  - CoreSight Trace Memory Controller with ETB configuration
  - Buffer size: 1 KB.

See reference 1. and 2. for details.

2.1.3 Operating Frequency

The operating frequencies for the MCU are as follows:

- CPU: maximum 48 MHz
- Serial Wire Output (SWO) trace interface: maximum 12.5 MHz
- Joint Test Action Group (JTAG) interface: maximum 12.5 MHz
- Serial Wire Debug (SWD) interface: maximum 12.5 MHz.

Figure 2.1 shows the block diagram of the Cortex-M4 CPU.

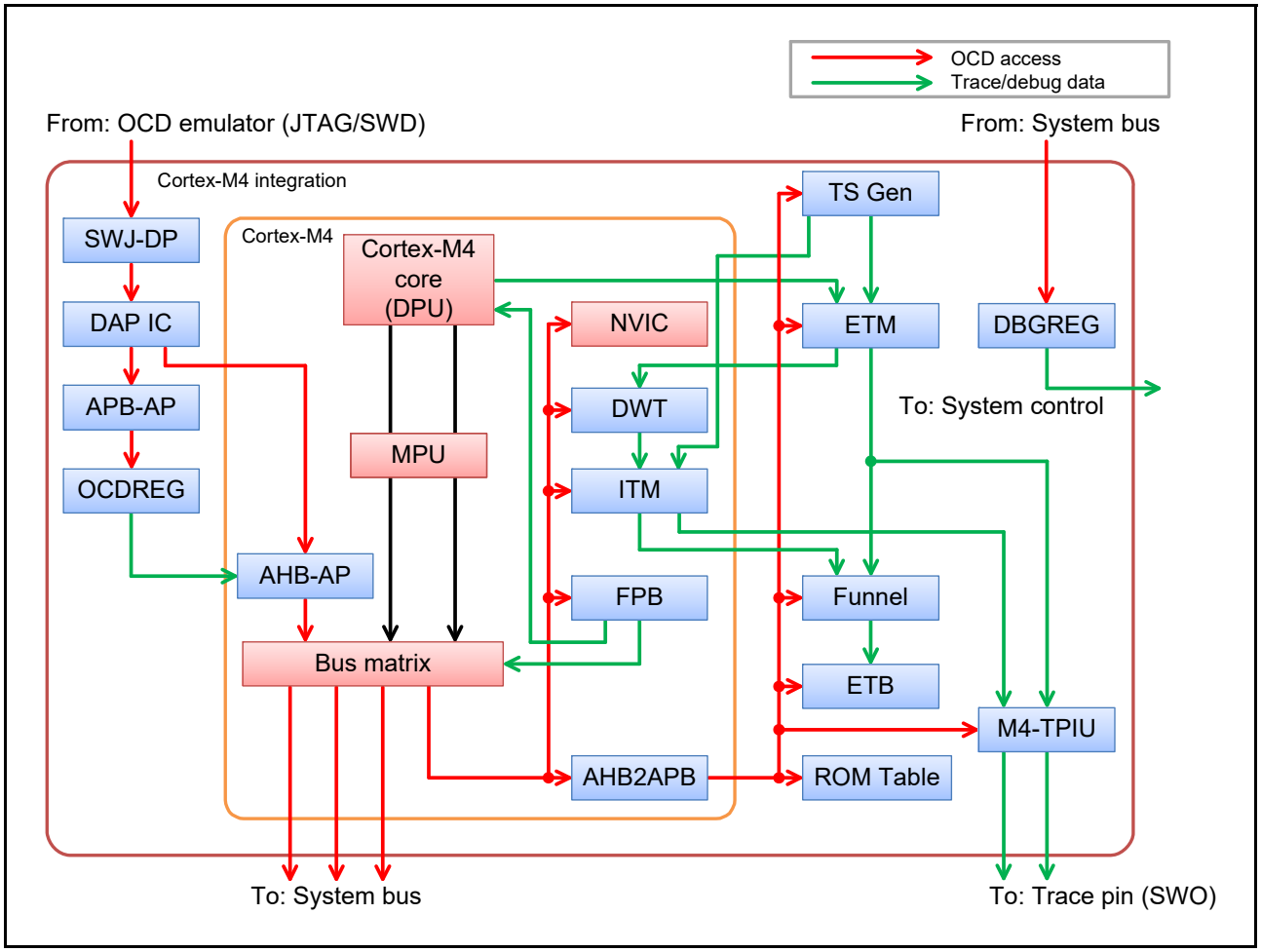


Figure 2.1 Cortex-M4 CPU block diagram

2.2 MCU Implementation Options

Table 2.1 shows the implementation options for the MCU and is based on the configurable options in reference 2.

Table 2.1 Implementation options

Component	Description
MPU	Included, 8 protected regions
FPB	Flash Patch (Remap) function is unavailable, only Breakpoint function is available
DWT	Included
ITM	Included
ETM	Included
AHB-AP	Included
HTM interface	Not included
TPIU	Included (only Serial Wire Output)
WIC	Not included The ICU can wake up the CPU instead of the Wakeup Interrupt Controller (WIC). For more details, see section 14, Interrupt Controller Unit (ICU).
Debug port	SWJ-DP
FPU	Included
Number of interrupts	64
Number of priority bits	4 bits (16 levels)
Endianness	Little-endian
Time Stamp Generator	Included
ETB	Included
Sleep mode power saving	Sleep mode and other low power modes are supported. For details, see section 11, Low Power Modes. SCB.SCR.SLEEPDEEP is ignored.
Memory features	Cacheable attribute is utilized in the MCU. See section 15, Buses for more details.
SysTick Timer SYST_CALIB register	Included SYST_CALIB = 4000 0147h Bit [31] = 0 Reference clock provided Bit [30] = 1 TERMS value is inexact Bits [29:24] = 00h Reserved Bits [23:0] = 000147h TERM: (32768 × 10 ms) - 1 / 32.768 kHz = 326.66 decimal = 327 with skew = 000147h
Event input/output	Not implemented
System reset request output	The SYSRESETREQ bit in the Application Interrupt and Reset Control Register causes a CPU reset
Auxiliary fault inputs, AUXFAULT	Not implemented

2.3 Trace Interface

A Serial Wire Output (SWO) provides trace output. Table 2.2 shows the MCU pin for the trace function. This pin is multiplexed with other functions.

Table 2.2 Trace function pin

Name	I/O	Width	Function	When not in use
TDO/SWO	Output	1 bit	Serial wire output multiplexed with JTAG TDO pin	Open

2.4 JTAG/SWD Interface

Table 2.3 shows the JTAG/Serial Wire Data (SWD) pins.

Table 2.3 JTAG/SWD pins

Name	I/O	P/N	Width	Function	When not in use
TCK/SWCLK	Input	Positive	1 bit	JTAG clock pin/SWD clock pin	Pull-up
TMS/SWDIO	I/O	Negative	1 bit	JTAG TMS pin/SWD I/O pin	Pull-up
TDI	Input	Positive	1 bit	JTAG TDI pin	Pull-up
TDO/SWO	Output	Negative	1 bit	JTAG TDO pin multiplexed with SWO pin	Open

2.5 Debug Mode

2.5.1 Debug Mode Definition

In single chip mode, the debugger state of the connection is defined as On-Chip Debugger (OCD) mode, the non-connected debugger state is defined as User mode. Table 2.4 shows the CPU debug modes and usage conditions.

Table 2.4 CPU Debug mode and conditions

Conditions		Mode	
OCD Connect	**JTAG/SWD Authentication**	**Debug Mode**	**Debug Authentication**
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 Armv7-M architecture. Enabled means that both invasive and non-invasive CPU debugging are permitted. Disabled means that both are not permitted.

2.5.2 Debug Mode Effects

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

2.5.2.1 Low Power Mode

All CoreSight debug components can store register settings even when the CPU enters Software Standby mode or Snooze mode. However, AHB-AP cannot respond to 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. See section 2.6.5.3, MCU Control Register (MCUCTRL) for details.

2.5.2.2 Reset

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

Table 2.5 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 or interrupt	Does not occur*1	Depends on DBGSTOPCR setting*2
Watchdog Timer reset or interrupt	Does not occur*1	Depends on DBGSTOPCR setting*2
Voltage monitor 0 reset	Depends on DBGSTOPCR setting*3	
Voltage monitor 1 reset or interrupt	Depends on DBGSTOPCR setting*3	
Voltage monitor 2 reset or interrupt	Depends on DBGSTOPCR setting*3	
SRAM parity error reset or interrupt	Depends on DBGSTOPCR setting*3	
SRAM ECC error reset or interrupt	Depends on DBGSTOPCR setting*3	
MPU bus master reset or interrupt	Same as user mode	
MPU bus slave reset or interrupt	Same as user mode	
Stack pointer error reset or 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.
Note 2. The IWDT and WDT operation depends on the DBGSTOPCR setting.
Note 3. Reset or interrupt masking depends on the DBGSTOPCR setting.

2.6 Programmers Model

2.6.1 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 OCD register.

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

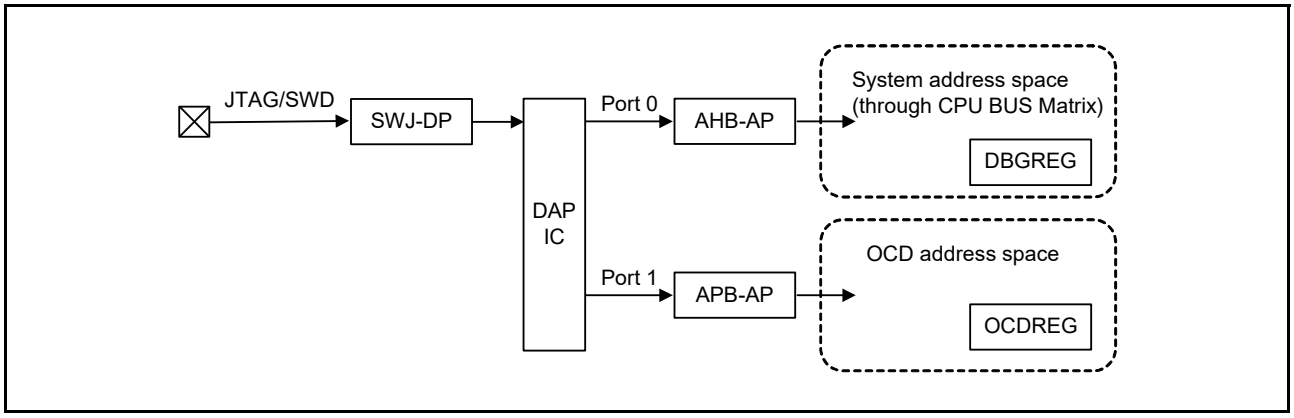


Figure 2.2 JTAG/SWD authentication block diagram

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 be accessed only from the OCD tool. The CPU and other bus masters cannot access the OCD registers.

2.6.2 Cortex-M4 Peripheral Address Map

In system address space, the Cortex-M4 core has a Private Peripheral Bus (PPB) that can only be accessed from the CPU and OCD emulator. The PPB is expanded from the Cortex-M4 original implementation for the MCU. Table 2.6 shows the address map of the MCU.

Table 2.6 Cortex-M4 peripheral address map

Component name	Start address	End address	Note
ITM	E000 0000h	E000 0FFFh	See reference 2.
DWT	E000 1000h	E000 1FFFh	See reference 2.
FPB	E000 2000h	E000 2FFFh	See reference 2.
SCS	E000 E000h	E000 EFFFh	See reference 2.
TPIU	E004 0000h	E004 0FFFh	See reference 2.
ETM	E004 1000h	E004 1FFFh	See reference 5.
ATB Funnel	E004 2000h	E004 2FFFh	See section 2.7 and reference 4.
ETB	E004 3000h	E004 3FFFh	See reference 6.
Time Stamp Generator	E004 4000h	E004 4FFFh	See section 2.10 and reference 4.
ROM Table	E00F F000h	E00F FFFFh	See section 2.6.3 and reference 7.

2.6.3 CoreSight ROM Table

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

2.6.3.1 ROM Entries

Table 2.7 shows the ROM entries in the CoreSight ROM table. The OCD emulator can use the ROM entries to determine which components are implemented in a system. See reference 7. for details.

Table 2.7 CoreSight ROM Table

#	Address	Access size	R/W	Value	Target module
0	E00F F000h	32 bits	R	FFF0F003	SCS
1	E00F F004h	32 bits	R	FFF02003	DWT
2	E00F F008h	32 bits	R	FFF03003	FPB
3	E00F F00Ch	32 bits	R	FFF01003	ITM
4	E00F F010h	32 bits	R	FFF41003	TPIU
5	E00F F014h	32 bits	R	FFF42003	ETM
6	E00F F018h	32 bits	R	FFF43003	Funnel
7	E00F F01Ch	32 bits	R	FFF44003	ETB
8	E00F F020h	32 bits	R	FFF45003	TSG
9	E00F F024h	32 bits	R	00000000	(End of entries)

2.6.3.2 CoreSight Component Registers

The CoreSight ROM Table lists the CoreSight component registers defined in the Arm CoreSight architecture. Table 2.8 lists these registers. See reference 7. for details on each register.

Table 2.8 CoreSight Component Registers in the CoreSight ROM table

Name	Address	Access size	R/W	Initial value
DEVTYPE	E00F FFCCh	32 bits	R	0000_0001h
PID4	E00F FFD0h	32 bits	R	0000_0004h
PID5	E00F FFD4h	32 bits	R	0000_0000h
PID6	E00F FFD8h	32 bits	R	0000_0000h
PID7	E00F FFDCh	32 bits	R	0000_0000h
PID0	E00F FFE0h	32 bits	R	0000_0002h
PID1	E00F FFE4h	32 bits	R	0000_0030h
PID2	E00F FFE8h	32 bits	R	0000_000Ah
PID3	E00F FFECh	32 bits	R	0000_0000h
CID0	E00F FFF0h	32 bits	R	0000_000Dh
CID1	E00F FFF4h	32 bits	R	0000_0010h
CID2	E00F FFF8h	32 bits	R	0000_0005h
CID3	E00F FFFCh	32 bits	R	0000_00B1h

2.6.4 DBGREG Module

The DBGREG module controls the debug functionalities and is implemented as a CoreSight-compliant component. Table 2.9 lists 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	4001 B000h	32 bits	R
Debug Stop Control Register	DBGSTOPCR	Port 0	4001 B010h	32 bits	R/W
Trace Control Register	TRACECTR	Port 0	4001 B020h	32 bits	R/W

2.6.4.1 Debug Status Register (DBGSTR)

Address(es): DBG.DBGSTR 4001 B000h

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

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

Bit	Symbol	Bit name	Description	R/W
b27 to b0	—	Reserved	These bits are read as 0.	R
b28	CDBGPWRUPREQ	Debug power-up request	0: OCD is not requesting debug power-up 1: OCD is requesting debug power-up.	R
b29	CDBGPWRUPACK	Debug power-up acknowledge	0: Debug power-up request is not acknowledged 1: Debug power-up request is acknowledged.	R
b31, b30	—	Reserved	These bits are read as 0	R

2.6.4.2 Debug Stop Control Register (DBGSTOPCR)

Address(es): DBG.DBGSTOPCR 4001 B010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	DBGSTOP_RECCR	DBGSTOP_RPER	—	—	—	—	—	DBGSTOP_LVD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	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	Bit name	Description	R/W
b0	DBGSTOP_IWDT	Mask bit for IWDT reset or interrupt	0: Enable IWDT reset or interrupt 1: Mask IWDT reset or interrupt and stop WDT count when CPU is in OCD break mode.	R/W
b1	DBGSTOP_WDT	Mask bit for WDT reset or interrupt	0: Enable WDT reset or interrupt 1: Mask WDT reset or interrupt and stop WDT count when CPU is in OCD break mode.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	DBGSTOP_LVD[2:0]	Mask bit for LVD0 reset	0: Enable LVD0 reset 1: Mask LVD0 reset.	R/W
b17		Mask bit for LVD1 reset or interrupt	0: Enable LVD1 reset or interrupt 1: Mask LVD1 reset or interrupt.	R/W
b18		Mask bit for LVD2 reset or interrupt	0: Enable LVD2 reset or interrupt 1: Mask LVD2 reset or interrupt.	R/W
b23 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	DBGSTOP_RPER	Mask bit for SRAM parity error reset or interrupt	0: Enable SRAM parity error reset or interrupt 1: Mask SRAM parity error reset or interrupt.	R/W
b25	DBGSTOP_RECCR	Mask bit for SRAM ECC error reset or interrupt	0: Enable SRAM ECC error reset or interrupt 1: Mask SRAM ECC error reset or interrupt.	R/W
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

2.6.4.3 Trace Control Register (TRACECTR)

Address(es): DBG.TRACECTR 4001 B020h

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

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

Bit	Symbol	Bit name	Description	R/W
b30 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b31	ENETBFULL	Enable bit for halt request on ETB full	0: ETB full does not cause a CPU halt 1: ETB full causes a CPU halt.	R/W

2.6.4.4 DBGREG CoreSight component registers

The DBGREG module provides the CoreSight component registers defined in the Arm CoreSight architecture. Table 2.10 lists these registers. See reference 7. for details on each register.

Table 2.10 DBGREG CoreSight component registers

Name	Address	Access size	R/W	Initial value
PID4	4001 BFD0h	32 bits	R	0000_0004h
PID5	4001 BFD4h	32 bits	R	0000_0000h
PID6	4001 BFD8h	32 bits	R	0000_0000h
PID7	4001 BFDCh	32 bits	R	0000_0000h
PID0	4001 BFE0h	32 bits	R	0000_0005h
PID1	4001 BFE4h	32 bits	R	0000_0030h
PID2	4001 BFE8h	32 bits	R	0000_001Ah
PID3	4001 BFECh	32 bits	R	0000_0000h
CID0	4001 BFF0h	32 bits	R	0000_000Dh
CID1	4001 BFF4h	32 bits	R	0000_00F0h
CID2	4001 BFF8h	32 bits	R	0000_0005h
CID3	4001 BFFCh	32 bits	R	0000_00B1h

2.6.5 OCDREG Module

The OCDREG register module controls the On-chip Debug (OCD) emulator functionalities and is implemented as a CoreSight-compliant component. Table 2.11 lists the OCDREG registers.

Table 2.11 OCDREG registers

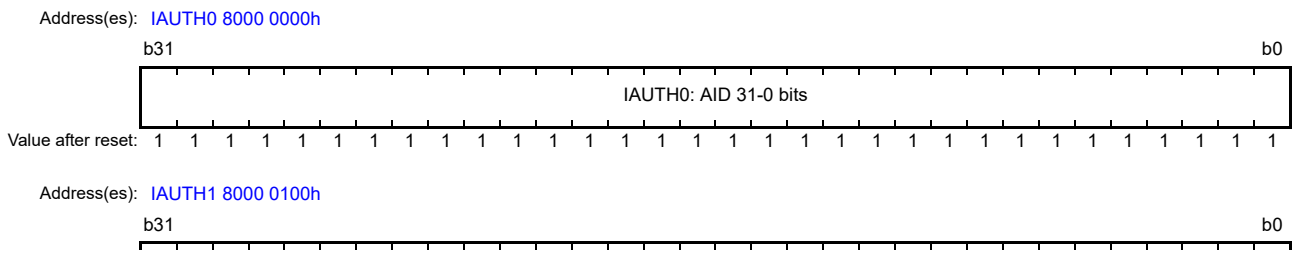
Name		DAP Port	Address	Access Size	R/W
ID Authentication Code Register 0	IAUTH0	Port 1	8000 0000h	32 bits	W
ID Authentication Code Register 1	IAUTH1	Port 1	8000 0100h	32 bits	W
ID Authentication Code Register 2	IAUTH2	Port 1	8000 0200h	32 bits	W
ID Authentication Code Register 3	IAUTH3	Port 1	8000 0300h	32 bits	W
MCU Status Register	MCUSTAT	Port 1	8000 0400h	32 bits	R
MCU Control Register	MCUCTRL	Port 1	8000 0410h	32 bits	R/W

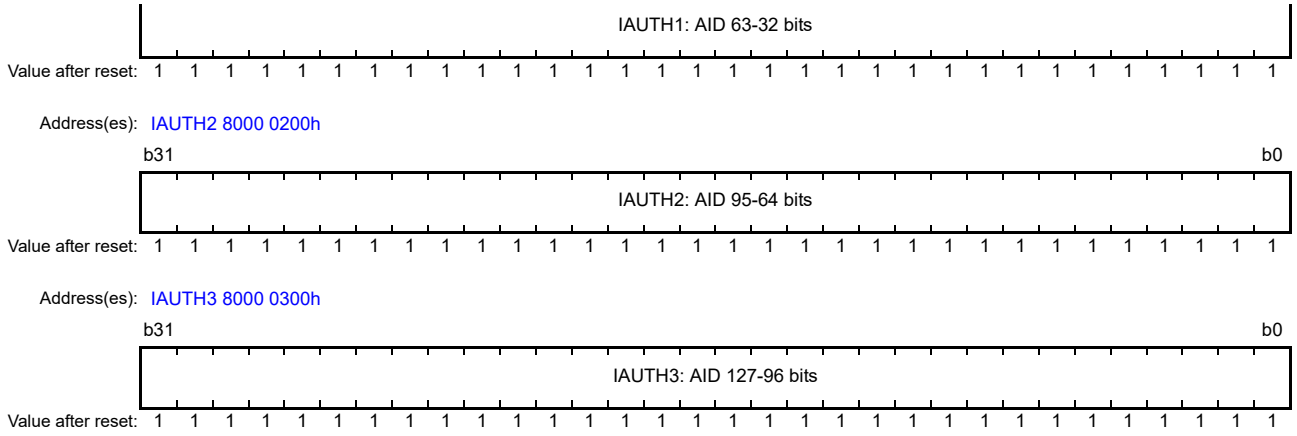
Note: OCDREG is located in the dedicated OCD address space. This address map is independent from the system address map.

2.6.5.1 ID Authentication Code Register (IAUTH0 to 3)

Four authentication registers are provided for writing the 128-bit key. These registers must be written in sequential order from IAUTH0 to IAUTH3. If the set of register writes is not compliant with this order, the result is unpredictable.

Only 32-bit writes are permitted. The initial value of the registers is all 1s. This means that JTAG/SWD access is initially permitted when the ID code in the OSIS register has the initial value. See section 2.11.2, Unlock ID Code.





2.6.5.2 MCU Status Register (MCUSTAT)

Address(es): MCUSTAT 8000 0400h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	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	Bit name	Description	R/W
b0	AUTH	Authentication Status	0:Authentication failed 1:Authentication succeeded.	R
b1	CPUSLEEP	-	0: CPU is not in Sleep mode 1: CPU is in Sleep mode.	R
b2	CPUSTOPCLK	-	0: CPU clock is not stopped, indicating that the MCU is in Normal mode or Sleep mode 1: CPU clock is stopped, indicating that the MCU is in Snooze mode or Software Standby mode.	R
b31 to b3	—	Reserved	These bits are read as 0.	R

Note 1. Depends on the chip status.

2.6.5.3 MCU Control Register (MCUCTRL)

Address(es): MCUCTRL 8000 0410h

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

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

Bit	Symbol	Bit name	Description	R/W
b0	EDBGRQ	External Debug Request	Writing 1 to the bit causes a CPU halt or debug monitor exception. 0: Debug event not requested 1: Debug event requested. When the EDBGRQ bit is set to 0 or the CPU is halted, the EDBCRQ bit is cleared.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	DBIRQ	Debug Interrupt Request	Writing 1 to the bit wakes up the MCU from low power mode. 0: Debug interrupt not requested 1: Debug interrupt requested. The condition can be cleared by writing 0 to the DBIRQ bit.	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set DBIRQ and EDBGRQ to the same value.

2.6.5.4 OCDREG CoreSight component registers

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

Table 2.12 lists these registers. See reference 7. for details on each register.

Table 2.12 OCDREG CoreSight Component registers

Name	Address	Access size	R/W	Initial value
PID4	8000 0FD0h	32 bits	Read only	0000_0004h
PID5	8000 0FD4h	32 bits	Read only	0000_0000h
PID6	8000 0FD8h	32 bits	Read only	0000_0000h
PID7	8000 0FDCh	32 bits	Read only	0000_0000h
PID0	8000 0FE0h	32 bits	Read only	0000_0004h
PID1	8000 0FE4h	32 bits	Read only	0000_0030h
PID2	8000 0FE8h	32 bits	Read only	0000_000Ah
PID3	8000 0FECh	32 bits	Read only	0000_0000h
CID0	8000 0FF0h	32 bits	Read only	0000_000Dh
CID1	8000 0FF4h	32 bits	Read only	0000_00F0h
CID2	8000 0FF8h	32 bits	Read only	0000_0005h
CID3	8000 0FFCh	32 bits	Read only	0000_00B1h

2.7 CoreSight ATB Funnel

There is one CoreSight ATB funnel in the chip. The funnel has two ATB slaves and one ATB master, and it selects the debug trace source from ETM and ITM to ETB. Figure 2.3 shows the CoreSight ATB connection in the chip.

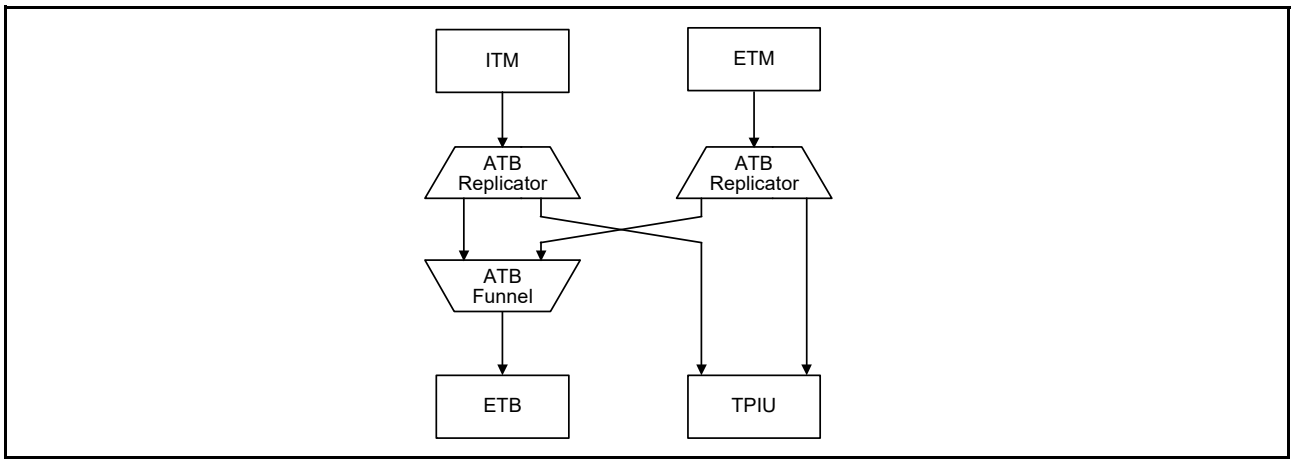


Figure 2.3 CoreSight ATB connection

Table 2.13 shows the ATB slave connection for the funnel.

Table 2.13 ATB slave connection

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

For details of ATB and funnel, see reference 4.

2.8 Flash Patch and Break Unit

The MCU has a Flash Patch and Break unit. Breakpoint function is available but flash patch (remap) function is unavailable. Therefore, do not set the REPLACE bits, [31:30], in the FP_COMPn register to 0. Bit [28] of the FP_REMAP register is always set to 1. When writing to this register, write 1 in bit [28]. When reading this register, bit [28] is always read as 1. See reference 1. for details.

2.9 SysTick System Timer

The SysTick system timer provides a simple 24-bit down counter. The reference clock for the timer can be selected as the CPU clock (ICLK) or clkly_stclk. See section 9, Clock Generation Circuit and reference 1.*1 for details.

Note 1. In reference 1., the IMPLEMENTATION DEFINED external reference clock is SYSTICCLK (LOCO), and the processor clock is ICLK.

2.10 CoreSight Time Stamp Generator

A CoreSight Time Stamp Generator provides a CPU clock-based timestamp to ITM and ETM. The 48 LSB bits of the 64-bit counter are used for the two components. See reference 4. for details.

2.11 OCD Emulator Connection

A JTAG/SWD authentication mechanism checks access permission for debug and chip resources. To obtain full debug functionality, a pass result of the authentication mechanism is required. Figure 2.4 shows the block diagram of the authentication mechanism.

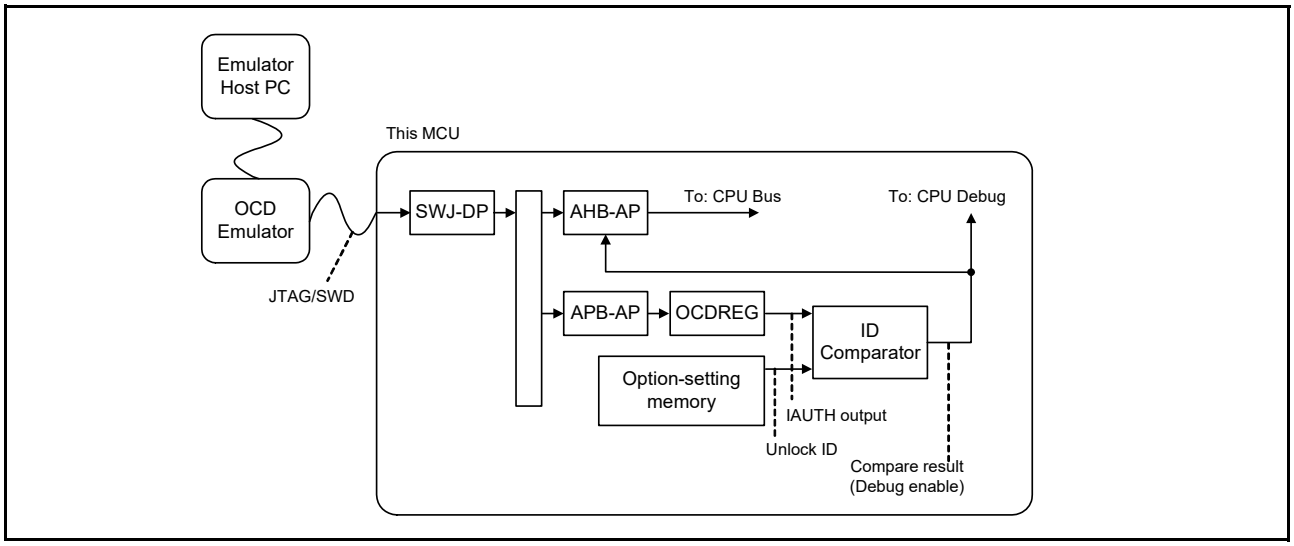


Figure 2.4 Authentication mechanism block diagram

An ID comparator is available in the MCU for authentication. The comparator compares the 128-bit IAUTH output from 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.

2.11.1 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. See section 11, Low Power Modes for details.

2.11.2 Unlock ID Code

The Unlock ID code is used for checking permissions for debug and access to on-chip resources. If the Unlock ID code matches the 128-bit data written in the ID Authentication Registers 0 to 3, the JTAG/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 (FFFFFFFF_FFFFFFFF_FFFFFFFF_FFFFFFFFh). See section 7, Option-Setting Memory for details.

2.11.3 Restrictions on Connecting an OCD Emulator

This section describes the restrictions on emulator access.

2.11.3.1 Starting connection while in a low power mode

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

2.11.3.2 Changing a low power mode while in OCD mode

When the chip is in OCD mode, the low power mode can be changed in the chip. 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 2.14 shows the constraints by mode.

Table 2.14 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, using the MCUCTRL.EDBGRQ bit in OCDREG, the OCD emulator can wake up the MCU without starting CPU execution by using a CPU break.

2.11.3.3 Modify the Unlock ID code in OSIS

After modifying the unlock ID code in 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 the reset.

2.11.3.4 Connecting Sequence and JTAG/SWD Authentication

Because the OCD emulator is protected by the JTAG/SWD authentication mechanism, the OCD might be required to input the ID code to the authentication registers. The OSIS value in the option-setting memory determines whether the code is required. After the negation of the reset, a 44 µs wait time is needed before comparing the OSIS at cold start.

(1) When MSB of OSIS is 0 (Bit [127] = 0)

The ID code is always non-matching, and connection to the on-chip debugger is prohibited.

(2) When OSIS is all 1s (default)

OCD authentication is not required and OCD can use the AHB-AP without the authentication.

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

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

OCD authentication is required and the OCD must write the unlock 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 JTAG interface or 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, and then wait until CSDBGPWRUPACK in the same register is asserted.
3. Set up the APB-AP to access OCDREG. APB-AP is connected to the DAP bus port 1.
4. Write the 128-bit ID code to IAUTH registers 0 to 3 in the OCDREG using the APB-AP.
5. If the 128-bit ID code matches the value of the OSIS, the AHB-AP is authorized to issue an AHB transaction. The authorization result can be confirmed in 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 a match with the OSIS value. AHB transfers are permitted.
   - When the DbgStatus bit is 0, the 128-bit ID code is not a match with the OSIS value. AHB transfers are not permitted.
6. Set up the AHB-AP to access 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] is 11b

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

When IATUH0-3 are "ALeRASE" in ASCII code (414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFFh), the content of Code flash, Data flash, and Configuration area are erased at once. See section 48., Flash Memory for details.

The ALeRASE sequence is as follows:

1. Connect the OCD debugger to the MCU through the JTAG or SWD interface.
2. Set up the 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 CSDBGPWRUPACK in the same register is asserted.
3. Set the APB-AP to access OCDREG. This APB-AP is connected to the DAP bus port 1.
4. Write the 128-bit ID code to IAUTH registers 0 to 3 in the OCDREG using the APB-AP.
5. If the 128-bit ID code is "ALeRASE" in ASCII code, the contents of Code flash, Data flash, and Configuration area are erased. Then, the MCU transitions to Sleep mode.

2.12 References

1. *ARM®v7-M Architecture Reference Manual* (ARM DDI 0403D)
2. *ARM® Cortex®-M4 Processor Technical Reference Manual* (ARM DDI 0439D)
3. *ARM® Cortex®-M4 Devices Generic User Guide* (ARM DUI 0553A)
4. *ARM® CoreSight™ SoC-400 Technical Reference Manual* (ARM DDI 0480F)
5. *ARM® CoreSight™ ETM-M4 Technical Reference Manual* (ARM DDI 0440C)
6. *ARM® CoreSight™ Trace Memory Controller Technical Reference Manual* (ARM DDI 0461B)
7. *ARM® CoreSight™ Architecture Specification* (ARM IHI 0029D)

3. Operating Modes

3.1 Overview

Table 3.1 shows the selection of operating modes by the mode-setting pin. For details, see section 3.2, Operating Mode Details. 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**	**On-chip flash memory**
1	Single-chip mode	Enable
0	SCI/USB boot mode	Enable

3.2 Operating Mode Details

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 chip 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 a dedicated area within the MCU, is used. The on-chip flash, including the code flash memory and data flash memory, can be modified from outside the MCU by using a Serial Communication Interface (SCI). For details, see section 48, Flash Memory. The MCU starts up in SCI boot mode if the MD pin is held low on release from the reset state.

3.2.3 USB Boot Mode

In this mode, the on-chip flash memory programming routine (USB boot program), stored in the boot area within the MCU, is used. The on-chip flash, including the code flash memory and data flash memory, can be modified from outside the MCU by using the USB. For details, see section 48, Flash Memory. The MCU starts in USB boot mode if the MD pin is held low on release from the reset state.

3.3 Operating Mode 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.

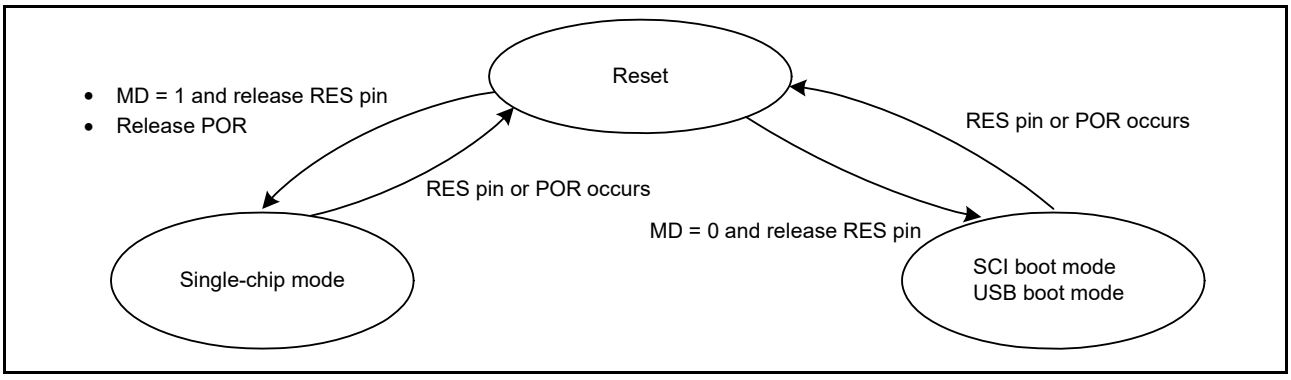
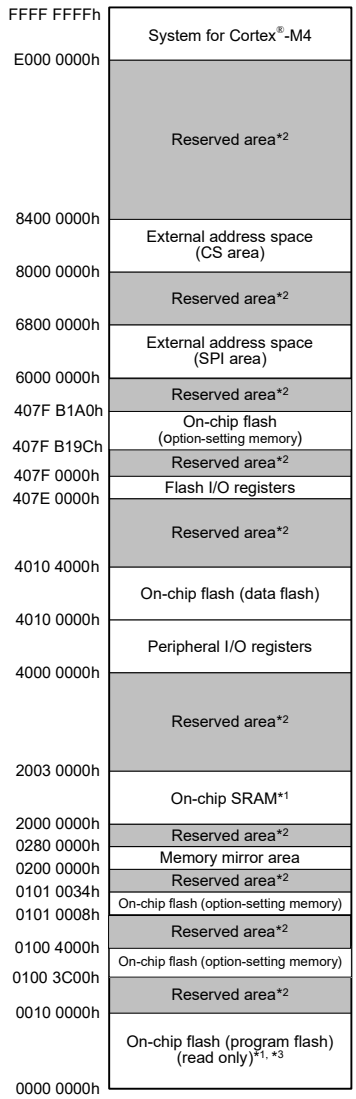


Figure 3.1 Mode-setting pin level and operating mode

4. Address Space

4.1 Address Space

The MCU supports a 4-GB linear address space ranging from 0000 0000h to FFFF FFFFh, that can contain both program and data. Figure 4.1 shows the memory map.

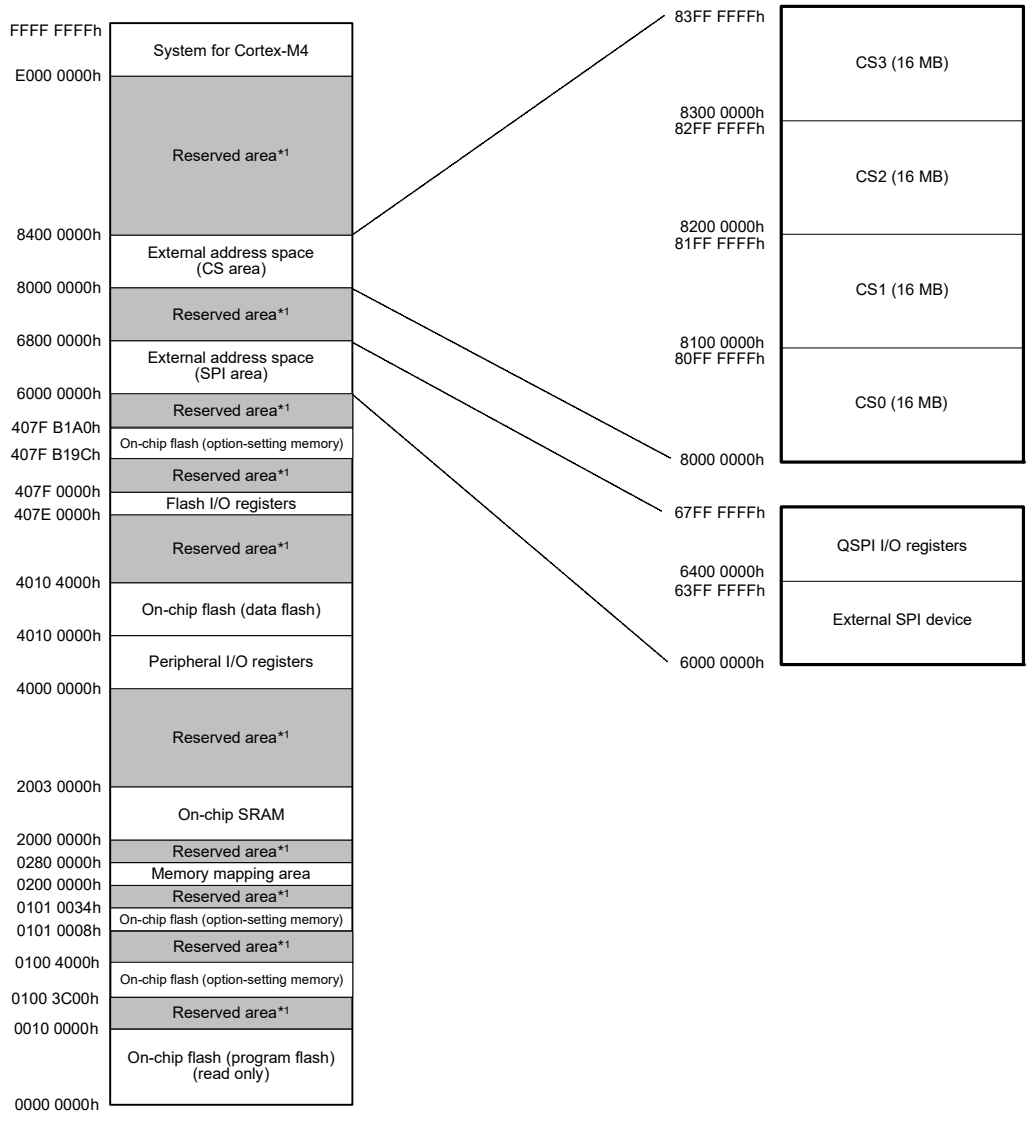


Note 1. The capacity of the flash or SRAM depends on the products.
Note 2. Do not access reserved areas.
Note 3. Some regions are reserved for the option-setting memory. For details, see section 7, Option-Setting Memory.

Figure 4.1 Memory map

4.2 External Address Space

The external address space is divided into CS areas (CS0 to CS3) and SPI area. The four CS areas (CS0 to CS3) each correspond to the CSn signal output from a CSn (n = 0 to 3) pin. The SPI area is divided into two areas, QSPI I/O registers, and external SPI device space. Figure 4.2 shows the address ranges associated with the individual CS areas (CS0 to CS3) and SPI area.



Note 1. Do not access reserved areas.

Figure 4.2 Association between external address spaces and CS areas

5. Memory Mirror Function (MMF)

5.1 Overview

The MCU provides a Memory Mirror Function (MMF). You can configure the MMF to map an application image load address in the code flash memory to the application image link address in the unused 23-bit memory mirror space addresses. Your application code must be developed and linked to run from this MMF destination address. The application code is not required to know the load location where it is stored in code flash memory.

Table 5.1 lists the MMF specifications.

Table 5.1 MMF specifications

Parameter	Description
Memory mirror space	8 MB (0200 0000h to 027F FFFFh)
Memory mirror boundary	128 bytes

5.2 Register Descriptions

5.2.1 MemMirror Special Function Register (MMSFR)

Address(es): MMF.MMSFR 4000 1000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	KEY[7:0]								—	MEMMIRADDR[15:9]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MEMMIRADDR[8:0]									—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Symbol	Bit name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b22 to b7	MEMMIRADDR[15:0]	Memory Mirror Address	0000h to FFFFh (8 MB)	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b31 to b24	KEY[7:0]	MMSFR Key Code	These bits enable or disable rewriting of the MEMMIRADDR bits	R/W

MEMMIRADDR[15:0] bits (Memory Mirror Address)

The MEMMIRADDR bits specify bits [22:7] of the memory mirror address. They define where the start address of the memory mirror space addresses (0200 0000h) is linked to. Writing to these bits is enabled only when this register is accessed in 32-bit units and the value DBh is written to the KEY[7:0] bits.

KEY[7:0] bits (MMSFR Key Code)

The KEY[7:0] bits enable or disable rewriting of the MEMMIRADDR[15:0] bits. Data written to the KEY[7:0] bits is not saved. These bits are read as 0. The KEY code and MEMMIRADDR[15:0] bits must be written to in the same cycle.

5.2.2 MemMirror Enable Register (MMEN)

Address(es): MMF.MMEN 4000 1004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	KEY[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bits	Symbol	Bit name	Description	R/W
b0	EN	Memory Mirror Function Enable	0: Disable MMF 1: Enable MMF.	R/W
b23 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b24	KEY[7:0]	MMEN Key Code	These bits enable or disable rewriting of the EN bit	R/W

EN bit (Memory Mirror Function Enable)

Writing to the EN bit is enabled only when the MemMirror Enable register is accessed in 32-bit units and the value DBh is written to the KEY[7:0] bits.

KEY[7:0] bits (MMEN Key Code)

The KEY[7:0] bits enable or disable rewriting of the EN bit. Data written to the KEY[7:0] bits is not saved. These bits are read as 0. The KEY code and the EN bit must be written in the same cycle.

5.3 Operation

5.3.1 MMF

The MMF links the memory mirror space (0200 0000h to 027F FFFFh) to the code flash area. If MMEN.EN = 1, the CPU can access code flash using both normal addresses (starting at 0000 0000h) and memory mirror space addresses (starting at 0200 0000h). Figure 5.1 shows an overview of the MMF. The MMSFR.MEMMIRADDR bits specify where the starting address of the memory mirror space addresses (0200 0000h) is linked to. Figure 5.2, Figure 5.3, and Figure 5.4 show the MMF operation. Figure 5.5 shows the setting procedure of the MMF.

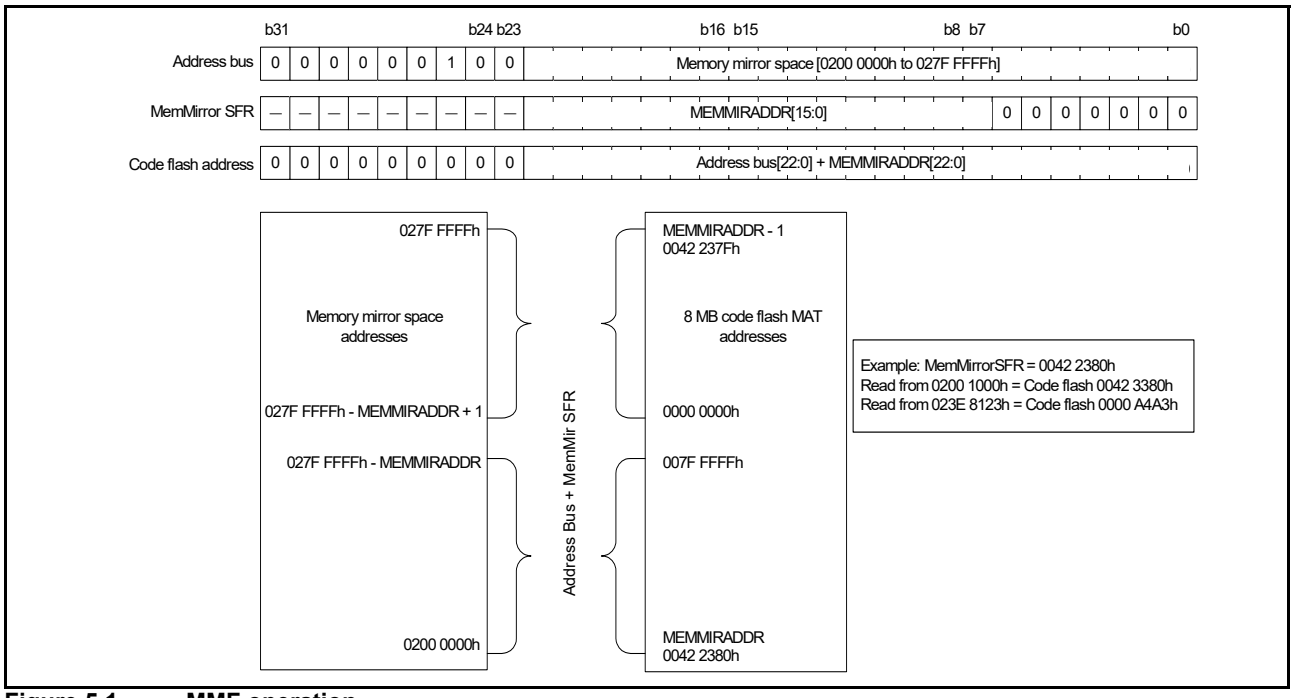
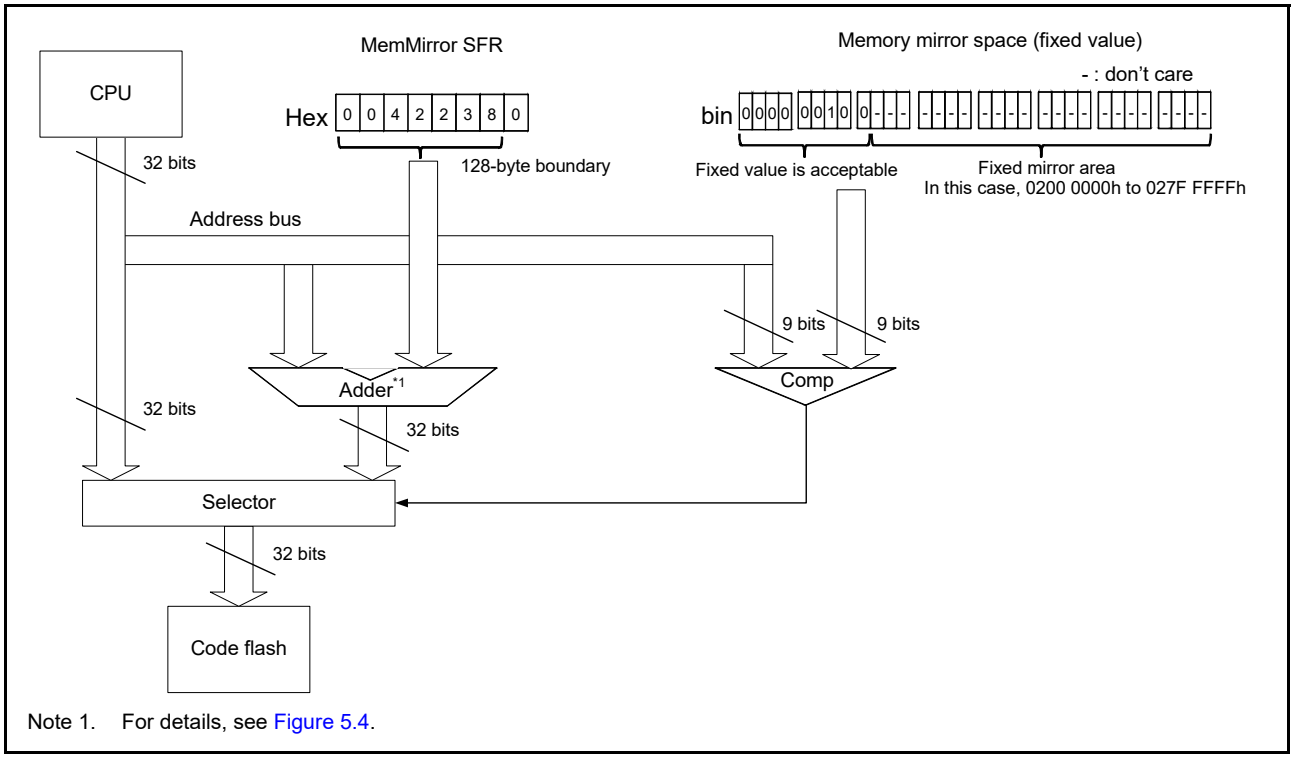


Figure 5.1 MMF operation



Note 1. For details, see Figure 5.4.

Figure 5.2 MMF block diagram

Figure 5.3 shows the addresses handled by each module. The Arm® MPU uses the original address of the CPU. The Security MPU and code flash memory each use an address after conversion through the Memory Mirror Function.

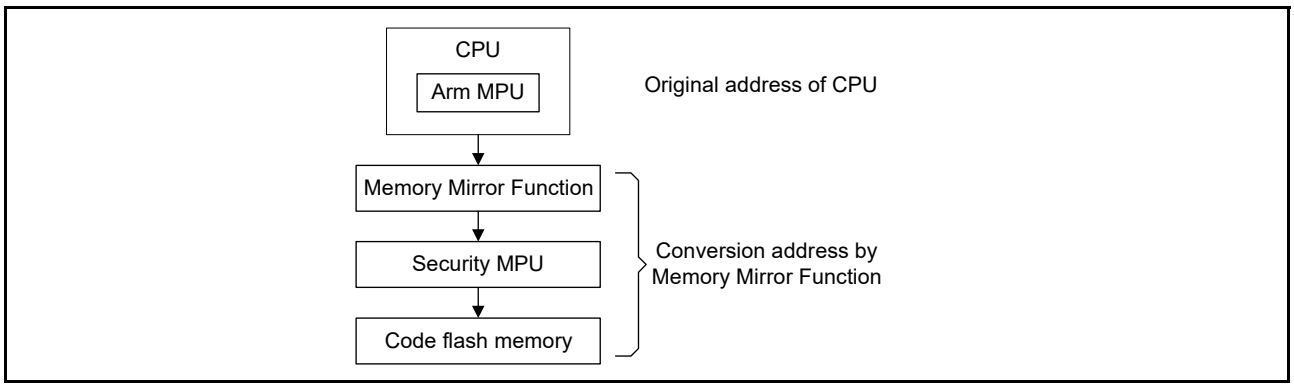


Figure 5.3 Address handling by each module

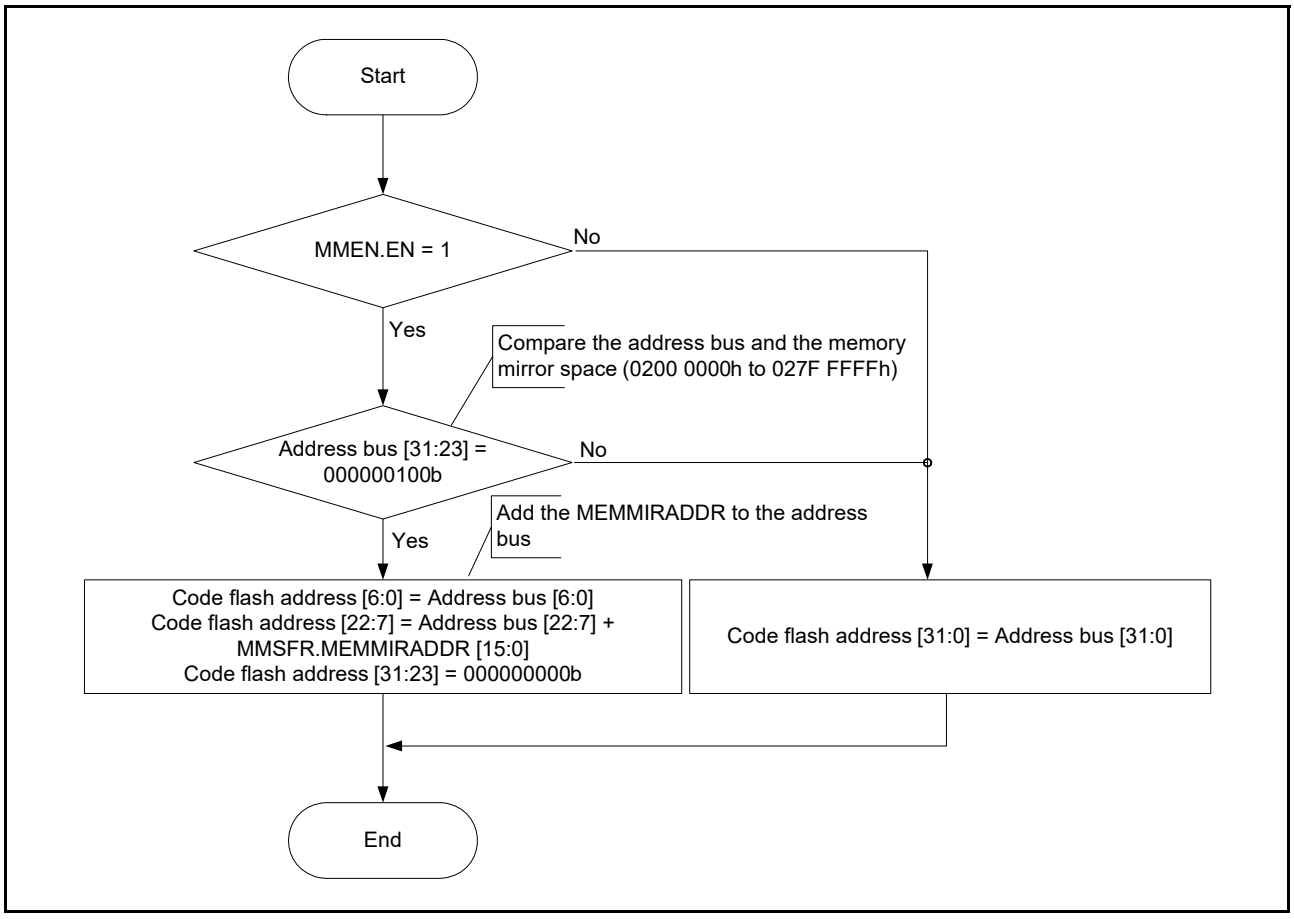


Figure 5.4 MMF operation flow

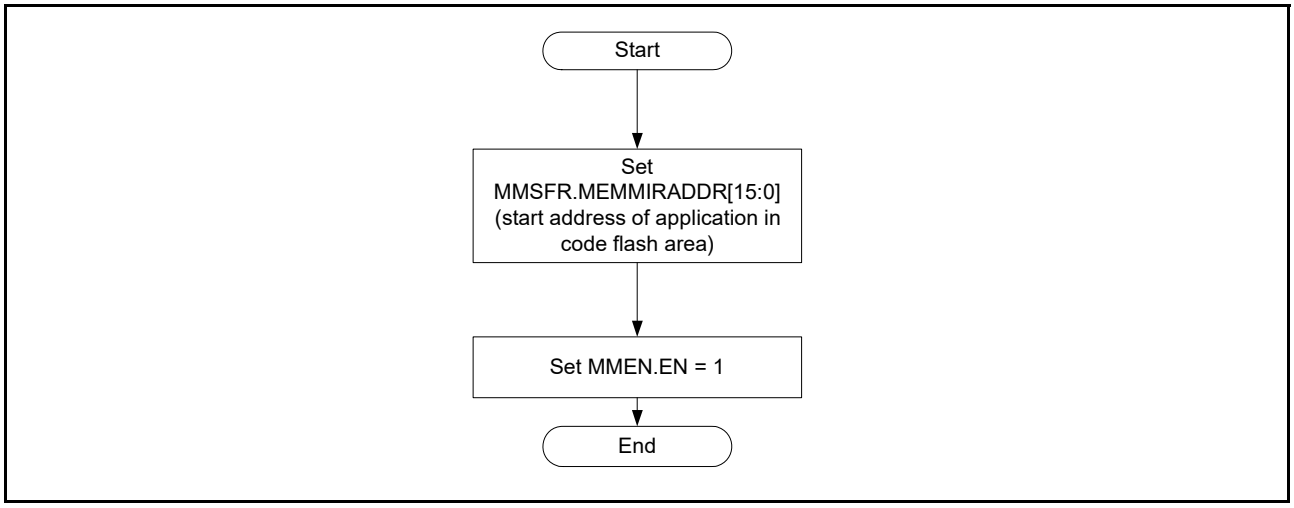


Figure 5.5 MMF setup flow

5.3.2 Setting Example

The application code on the code flash can be accessed from the address 0200 0000h on the memory mirror space by setting up the code flash address in MMSFR.MEMMIRADDR, and setting MMEN.EN to 1.

Figure 5.6 shows an example of how to use the MMF.

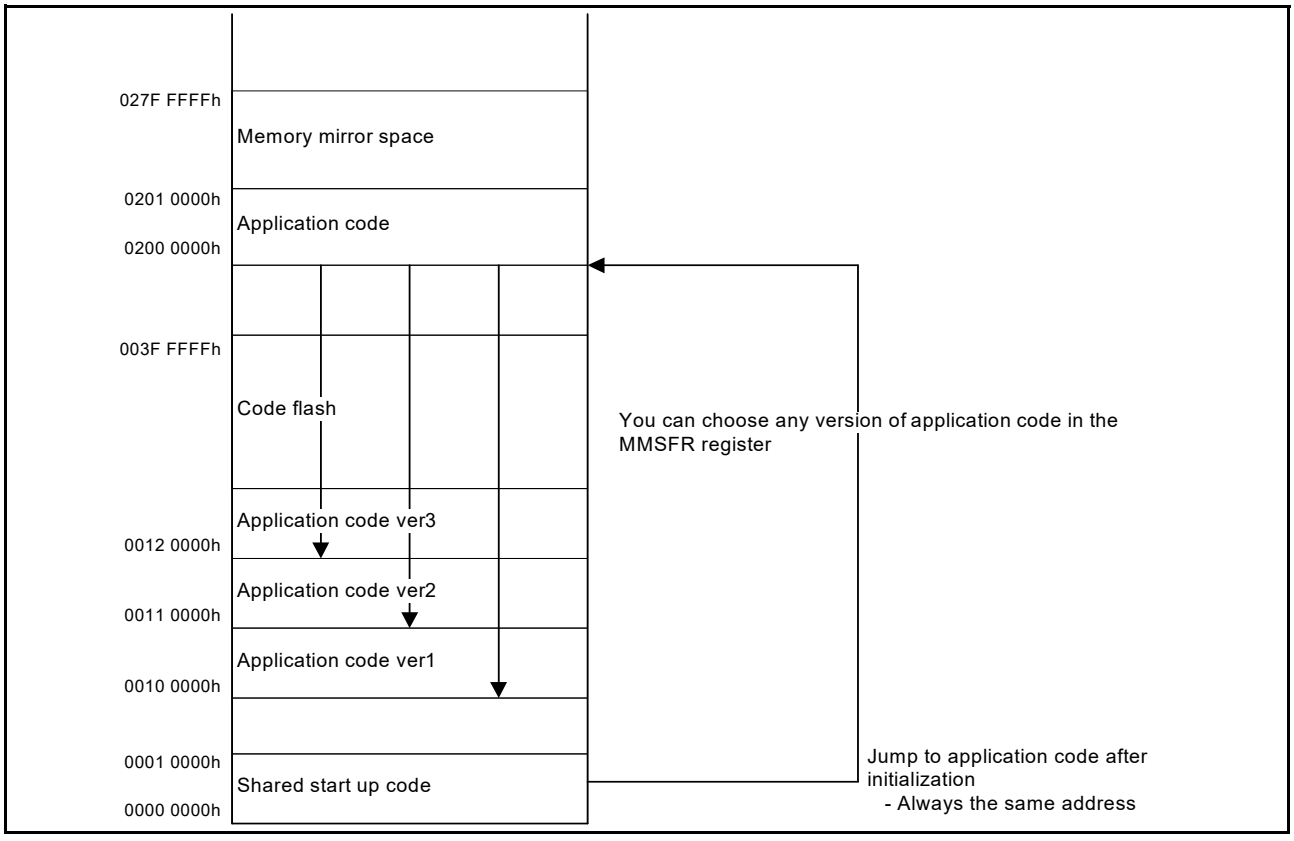


Figure 5.6 MMF setting example

Set the MMSFR register to DB10 0000h to use the application code ver1.

Set the MMSFR register to DB11 0000h to use the application code ver2.

Set the MMSFR register to DB12 0000h to use the application code ver3.

6. Resets

6.1 Overview

The MCU provides 14 resets:

- RES pin reset
- Power-on reset
- VBATT-selected voltage power-on reset
- Independent watchdog timer reset
- Watchdog timer reset
- Voltage monitor 0 reset
- Voltage monitor 1 reset
- Voltage monitor 2 reset
- SRAM parity error reset
- SRAM ECC error reset‘
- Bus master MPU error reset
- Bus slave MPU error reset
- CPU stack pointer error reset
- Software reset.

Table 6.1 lists the reset names and sources.

Table 6.1 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
VBATT selected voltage power-on reset	VCC fall (voltage detection: $V_{DETBATT}$)*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
SRAM ECC error reset	SRAM ECC 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}, V_{det2}, and $V_{DETBATT}$), see section 8, Low Voltage Detection (LVD), and section 52, Electrical Characteristics.

The internal state and pins are initialized by a reset. Table 6.2 and Table 6.3 list the targets initialized by resets.

Table 6.2 Reset detect flags initialized by each reset source

Flags to be initialized	Reset source							
	RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset	Voltage monitor 1 reset	Voltage monitor 2 reset	Software 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)	✓	✓	✓	×	×	×	×	×
SRAM ECC Error Reset Detect Flag (RSTSR1.REERF)	✓	✓	✓	×	×	×	×	×
Bus Slave MPU Error Reset Detect Flag (RSTSR1.BUSSRF)	✓	✓	✓	×	×	×	×	×
Bus Master MPU Error Reset Detect Flag (RSTSR1.BUSMRF)	✓	✓	✓	×	×	×	×	×
Stack Pointer Error Reset Detect Flag (RSTSR1.SPERF)	✓	✓	✓	×	×	×	×	×
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	×	✓	×	×	×	×	×	×

Flags to be initialized	Reset source					
	SRAM parity error reset	SRAM ECC error reset	Bus master MPU error reset	Bus slave MPU error reset	Stack pointer error reset	VBATT_POR*1
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)	×	×	×	×	×	×
SRAM ECC Error Reset Detect Flag (RSTSR1.REERF)	×	×	×	×	×	×
Bus Slave MPU Error Reset Detect Flag (RSTSR1.BUSSRF)	×	×	×	×	×	×
Bus Master MPU Error Reset Detect Flag (RSTSR1.BUSMRF)	×	×	×	×	×	×
Stack Pointer Error Reset Detect Flag (RSTSR1.SPERF)	×	×	×	×	×	×
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	×	×	×	×	×	×

✓: Initialized to 0
×: Not initialized

Note 1. For VBATT_POR details, see section 12, Battery Backup Function.

Table 6.3 Module-related registers initialized by each reset source

Registers to be initialized		Reset source							
		RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset	Voltage monitor 1 reset	Voltage monitor 2 reset	Software reset
Watchdog timer registers	WDTRR, WDTCR, WDTSR, WDTRCR, WDTCSTPR	✓	✓	✓	✓	✓	✓	✓	✓
Voltage monitor function 1 registers	LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL	✓	✓	✓	✓	✓	×	×	×
	LVD1CR1/LVD1SR	✓	✓	✓	✓	✓	×	×	×
Voltage monitor function 2 registers	LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL	✓	✓	✓	✓	✓	×	×	×
	LVD2CR1/LVD2SR	✓	✓	✓	✓	✓	×	×	×
SOSC registers	SOSCCR	×	×	×	×	×	×	×	×
	SOMCR	×	×	×	×	×	×	×	×
LOCO registers	LOCOCR	✓	✓	✓	✓	✓	✓	✓	✓
	LOCOUTCR	×	×	×	×	×	×	×	×
MOSC registers	MOMCR	✓	✓	✓	✓	✓	✓	✓	✓
Realtime clock*2 registers		×	×	×	×	×	×	×	×
AGT registers		×	×	×	×	×	×	×	×
MPU registers		✓	✓	✓	✓	✓	✓	✓	✓
Pin states (except XCIN/XCOUT pin)		✓	✓	✓	✓	✓	✓	✓	✓
Pin states (XCIN/XCOUT pin)		×	×	×	×	×	×	×	×
Battery backup registers	VBTCR1	×	✓	×	×	×	×	×	×
	VBTCR2, VBTSR, VBTCMPCR, VBTLVDICR, VBTWCTLR, VBTWCH0OTSR, VBTWCH1OTSR, VBTWCH2OTSR, VBTICTLR, VBTOCTLR, VBTWTER, VBTWEGR, VBTWFR	×	×	×	×	×	×	×	×
	VBTBKRn(n = 0 to 511)	×	×	×	×	×	×	×	×
Registers other than those shown, CPU, and internal state		✓	✓	✓	✓	✓	✓	✓	✓

Registers to be initialized		Reset source					
		SRAM parity error reset	SRAM ECC error reset	Bus master MPU error reset	Bus slave MPU error reset	Stack pointer error reset	VBATT_POR*3
Watchdog timer registers	WDTRR, WDTCR, WDTSR, WDTRCR, WDTCSTPR	✓	✓	✓	✓	✓	×
Voltage monitor function 1 registers	LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL	×	×	×	×	×	×
	LVD1CR1/LVD1SR	×	×	×	×	×	×
Voltage monitor function 2 registers	LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL	×	×	×	×	×	×
	LVD2CR1/LVD2SR	×	×	×	×	×	×
SOSC registers	SOSCCR	×	×	×	×	×	✓*1
	SOMCR	×	×	×	×	×	✓
LOCO registers	LOCOCR	✓	✓	✓	✓	✓	✓
	LOCOUTCR	×	×	×	×	×	✓
MOSC registers	MOMCR	✓	✓	✓	✓	✓	×
Realtime Clock (RTC) register*2		×	×	×	×	×	×
AGT register		×	×	×	×	×	✓
MPU register		✓	✓	×	×	×	×
Pin states (except XCIN/XCOUT pin)		✓	✓	✓	✓	✓	×
Pin states (XCIN/XCOUT pin)		×	×	×	×	×	✓
Battery backup registers	VBTCR1	×	×	×	×	×	×
	VBTCR2, VBTSR, VBTCMPCR, VBTLVDICR, VBTWCTLR, VBTWCH0OTSR, VBTWCH1OTSR, VBTWCH2OTSR, VBTICTLR, VBTOCTLR, VBTWTER, VBTWEGR, VBTWFR	×	×	×	×	×	✓

Registers to be initialized		Reset source					
		SRAM parity error reset	SRAM ECC error reset	Bus master MPU error reset	Bus slave MPU error reset	Stack pointer error reset	VBATT_POR*3
Battery backup registers	VBTBKRn(n = 0 to 511)	×	×	×	×	×	×
Registers other than those shown, CPU, and internal state		✓	✓	✓	✓	✓	×

✓: Initialized

×: Not initialized

Note 1. For the initial value of each register, see section 9, Clock Generation Circuit.

Note 2. The RTC has a software reset. RCR1.RTCOS, RCR1.CIE, RCR2.RTCOE, RCR2.ADJ30, and RCR2.RESET are initialized by all types of resets. For details on the target bits, see section 25, Realtime Clock (RTC).

Note 3. For VBATT_POR details, see section 12, Battery Backup Function.

RTC is not initialized by any reset source. AGT are not initialized by any reset source except VBATT_POR. SOSC and LOCO can be selected as the clock sources of RTC and AGT. Table 6.4 and Table 6.5 show the states of SOSC and LOCO when a reset occurs.

Table 6.4 States of SOSC when a reset occurs

State		Reset Source	
		VBATT_POR	Other
SOSC	Enable or disable	Initialized to disable	Continue with the state that was selected before the reset occurred
	Drive capability	Initialized to Normal mode	Continue with the state that was selected before the reset occurred
	XCIN/XCOUT	Initialized to general-purpose input pins	Continue with the state that was selected before the reset occurred

Table 6.5 States of LOCO when a reset occurs

State		Reset source	
		VBATT_POR	Other
LOCO	Enable or disable	Initialized to enable	
	Oscillation accuracy	Initialized to accuracy before trimming by LOCOUTCR (accuracy: +/- 15%)	Continue the accuracy that was trimmed by LOCOUTCR

When a reset is canceled, a reset exception handling starts.

Table 6.6 lists the pin related to the reset function.

Table 6.6 Reset I/O pin

Pin name	I/O	Function
RES	Input	Reset pin

6.2 Register Descriptions

6.2.1 Reset Status Register 0 (RSTSR0)

Address(es): SYSTEM.RSTSR0 4001 E410h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	LVD2RF	LVD1RF	LVD0RF	PORF
Value after reset:	0	0	0	0	x*1	x*1	x*1	x*1

Bit	Symbol	Bit name	Description	R/W
b0	PORF	Power-On Reset Detect Flag	0: Power-on reset not detected 1: Power-on reset detected.	R(/W)*2
b1	LVD0RF	Voltage Monitor 0 Reset Detect Flag	0: Voltage monitor 0 reset not detected 1: Voltage monitor 0 reset detected.	R(/W)*2
b2	LVD1RF	Voltage Monitor 1 Reset Detect Flag	0: Voltage monitor 1 reset not detected 1: Voltage monitor 1 reset detected.	R(/W)*2
b3	LVD2RF	Voltage Monitor 2 Reset Detect Flag	0: Voltage monitor 2 reset not detected 1: Voltage monitor 2 reset detected.	R(/W)*2
b7 to b4	—	Reserved	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.

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 Table 6.2 occurs
- When 1 is read from 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 Table 6.2 occurs
- When 1 is read from and then 0 is written to LVD0RF.

LVD1RF flag (Voltage Monitor 1 Reset Detect Flag)

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

[Setting condition]

- When a voltage monitor 1 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to LVD1RF.

LVD2RF flag (Voltage Monitor 2 Reset Detect Flag)

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

[Setting condition]

- When a voltage monitor 2 reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to LVD2RF.

6.2.2 Reset Status Register 1 (RSTSR1)

Address(es): SYSTEM.RSTSR1 4001 E0C0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SPERF	BUSMRF	BUSSRF	REERF	RPERF	—	—	—	—	—	SWRF	WDTRF	IWDTRF
Value after reset:	0	0	0	x*1	x*1	x*1	x*1	x*1	0	0	0	0	0	x*1	x*1	x*1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	IWDTRF	Independent Watchdog Timer Reset Detect Flag	0: Independent watchdog timer reset not detected 1: Independent watchdog timer reset detected.	R(/W)*2
b1	WDTRF	Watchdog Timer Reset Detect Flag	0: Watchdog timer reset not detected 1: Watchdog timer reset detected.	R(/W)*2
b2	SWRF	Software Reset Detect Flag	0: Software reset not detected 1: Software reset detected.	R(/W)*2
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	RPERF	SRAM Parity Error Reset Detect Flag	0: SRAM parity error reset not detected 1: SRAM parity error reset detected.	R(/W)*2
b9	REERF	SRAM ECC Error Reset Detect Flag	0: SRAM ECC error reset not detected 1: SRAM ECC error reset detected.	R(/W)*2
b10	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
b11	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
b12	SPERF	SP Error Reset Detect Flag	0: SP error reset not detected 1: SP error reset detected.	R(/W)*2
b15 to b13	—	Reserved	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 occurred.

[Setting condition]

- When an independent watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to IWDTRF.

WDTRF flag (Watchdog Timer Reset Detect Flag)

The WDTRF flag indicates that a watchdog timer reset occurred.

[Setting condition]

- When a watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to WDTRF.

SWRF flag (Software Reset Detect Flag)

The SWRF flag indicates that a software reset occurred.

[Setting condition]

- When a software reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from 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 occurred.

[Setting condition]

- When an SRAM parity error reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to RPERF.

REERF flag (SRAM ECC Error Reset Detect Flag)

The REERF flag indicates that an SRAM ECC error reset occurred.

[Setting condition]

- When a SRAM ECC error reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to REERF.

BUSSRF flag (Bus Slave MPU Error Reset Detect Flag)

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

[Setting condition]

- When a bus slave MPU error reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from 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 occurred.

[Setting condition]

- When a bus master MPU error reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When 1 is read from and then 0 is written to BUSMRF.

SPERF flag (SP Error Reset Detect Flag)

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

[Setting condition]

- When a stack pointer error reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs
- When 1 is read from and then 0 is written to SPERF.

6.2.3 Reset Status Register 2 (RSTSR2)

Address(es): SYSTEM.RSTSR2 4001 E411h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CWSF
Value after reset:	0	0	0	0	0	0	0	x*1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	CWSF	Cold/Warm Start Determination Flag	0: Cold start 1: Warm start.	R(/W)*2
b7 to b1	—	Reserved	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 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. Writing 0 to CWSF does not set it to 0.

[Clearing condition]

- When a reset listed in Table 6.2 occurs.

6.3 Operation

6.3.1 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. For a successful MCU reset, the RES pin must be held low for the power supply stabilization time specified for 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 52, Electrical Characteristics.

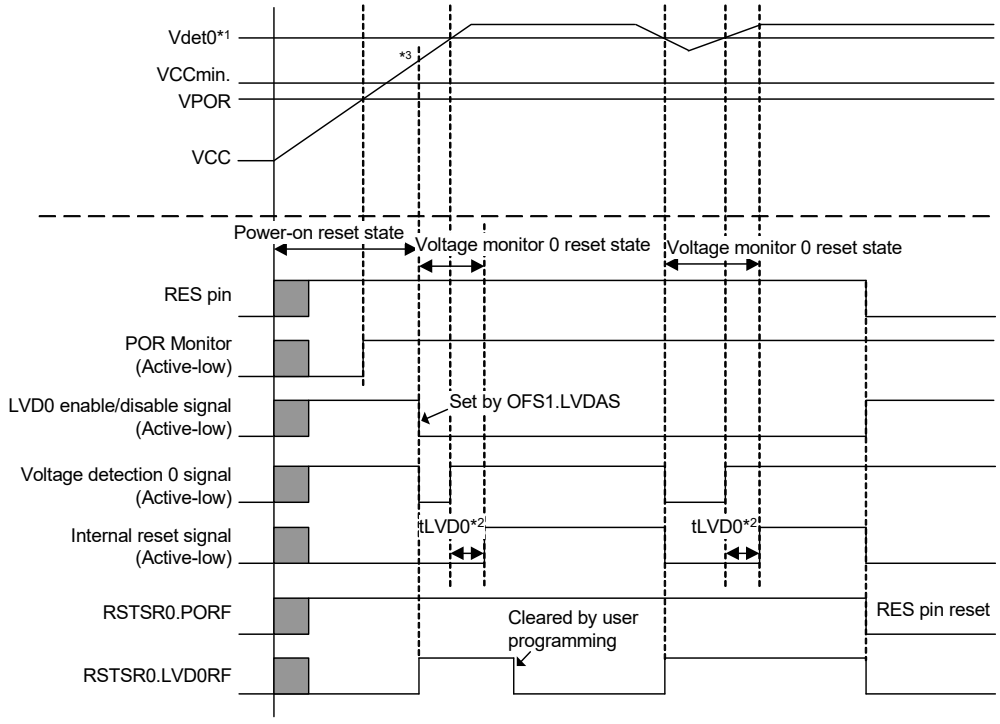
6.3.2 Power-On Reset

The power-on reset (POR) is an internal reset generated by the power-on reset circuit. If the RES pin is in a high level state when power is supplied, a power-on reset is generated. After VCC exceeds V_{POR} and the specified power-on reset time elapses, the internal reset is canceled and the CPU starts the reset exception handling. The power-on reset time is the stabilization period for 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.

The voltage monitor 0 reset is an internal reset generated by the voltage monitor circuit. If the Voltage Detection 0 Circuit Start (LVDAS) bit in 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. The V_{det0} voltage detection level can be changed by the setting in the VDSEL1[2:0] bits in Option Function Select Register 1 (OFS1).

Figure 6.1 shows examples of operations during a power-on reset and voltage monitor 0 reset.



Note: For details on the electrical characteristics, see section 52, Electrical Characteristics.
Note 1. V_{det0} shows a voltage monitor 0 reset detection level, and V_{POR} shows a power-on reset detection level, and VCCmin shows the minimum guaranteed voltage of the MCU.
Note 2. t_{LVD0} shows a time for voltage monitor 0 reset.
Note 3. At power-on, VCC should have risen to the minimum guaranteed voltage before the POR reset is released.

Figure 6.1 Example of operations during power-on and voltage monitor 0 resets

6.3.3 Voltage Monitor Reset

The voltage monitor 0 reset is an internal reset generated by the voltage monitor circuit. If the Voltage Detection 0 Circuit Start (LVDAS) bit in 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 a 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 the 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}.

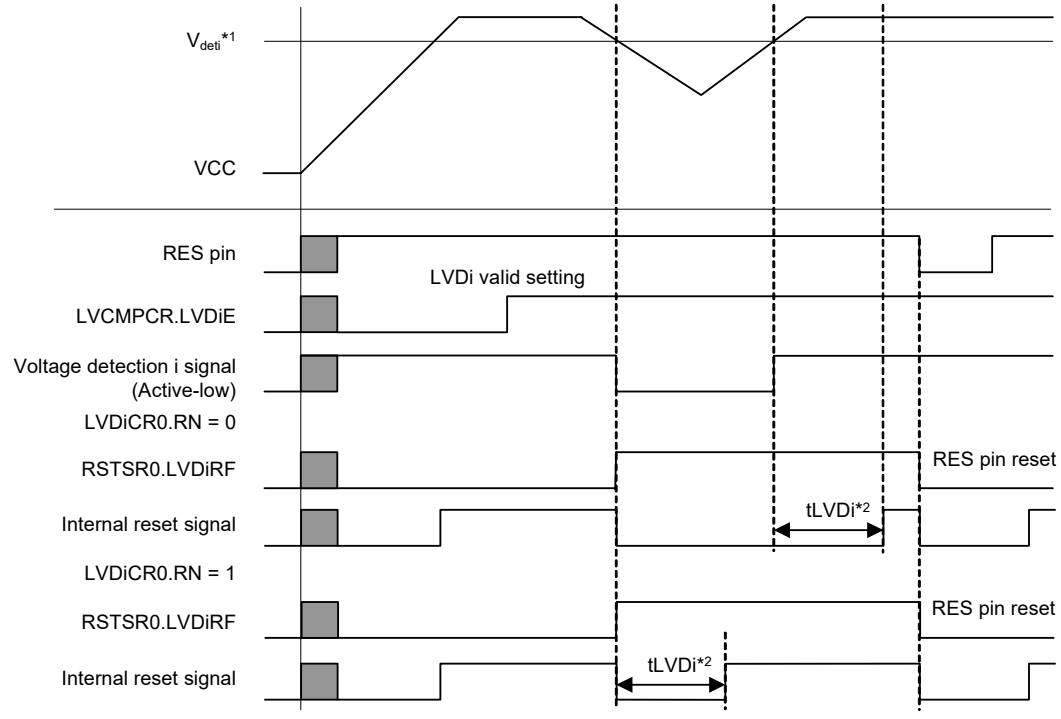
Likewise, when the Voltage Monitor 2 Interrupt/reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 2 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) 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 in 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 (tLVD1) 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 (tLVD1) elapses.

Likewise, timing for release from the voltage monitor 2 reset state is selectable in the Voltage Monitor 2 Reset Negate Select bit (RN) in the LVD2CR0 register.

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

Figure 6.2 shows examples of operation during voltage monitor 1 and 2 resets. For details on the voltage monitor 1 and voltage monitor 2 resets, see section 8, Low Voltage Detection (LVD).



Note: For details on the electrical characteristics, see section 52, Electrical Characteristics.
Note 1. Vdeti indicates the detection level of a voltage monitor 1 reset and voltage monitor 2 reset (i = 1, 2).
Note 2. t_{LVDi} indicates the time for a voltage monitor 1 reset and voltage monitor 2 reset (i = 1, 2).

Figure 6.2 Example of operations during voltage monitor 1 and voltage monitor 2 resets

6.3.4 Independent Watchdog Timer Reset

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

When output of the IWDT 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 27, Independent Watchdog Timer (IWDT).

6.3.5 Watchdog Timer Reset

The watchdog timer reset is an internal reset generated from the Watchdog Timer (WDT). Output of the watchdog timer 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, the 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 26, Watchdog Timer (WDT).

6.3.6 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®-M4 Technical Reference Manual*.

6.3.7 Determination of Cold/Warm Start

Read the CWSF flag in RSTSR2 to determine the cause of reset processing. The 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 6.3 shows an example of cold/warm start determination operation.

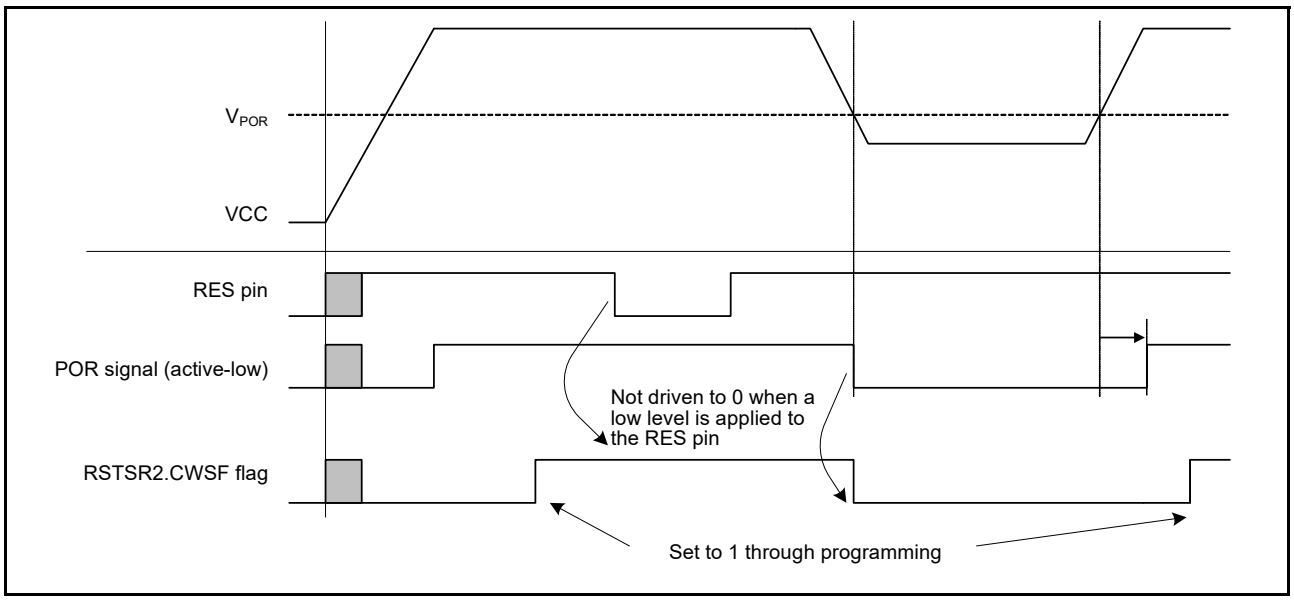
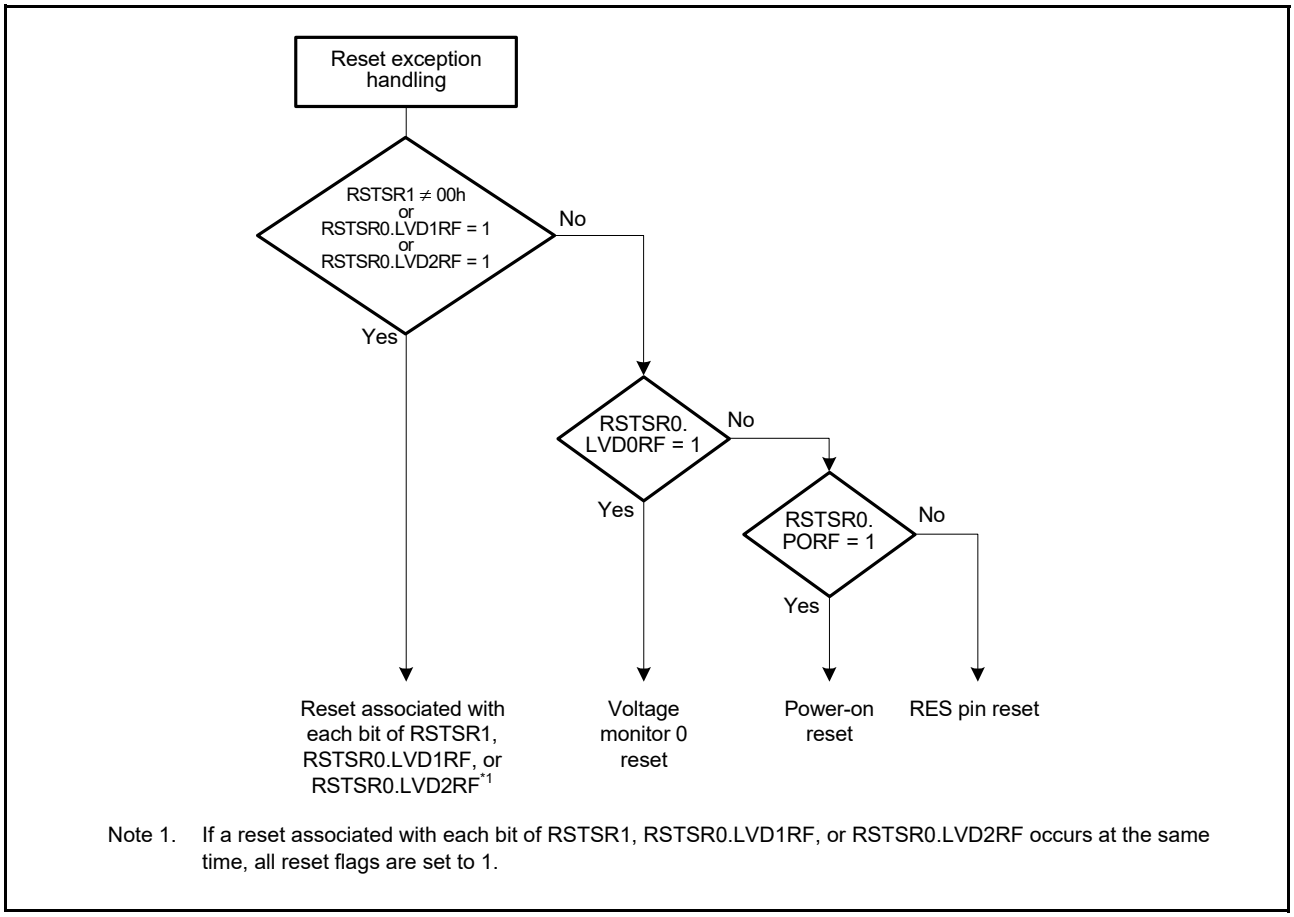


Figure 6.3 Example of cold/warm start determination operation

6.3.8 Determination of Reset Generation Source

Read RSTSR0 and RSTSR1 to determine which reset is used to execute the reset exception handling. Figure 6.4 shows an example flow to identify a reset generation source. The reset flag must be written with 0 after the reset flag is read as 1.



Note 1. If a reset associated with each bit of RSTSR1, RSTSR0.LVD1RF, or RSTSR0.LVD2RF occurs at the same time, all reset flags are set to 1.

Figure 6.4 Example of reset generation source determination flow

7. Option-Setting Memory

7.1 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 7.1 shows the option-setting memory area.

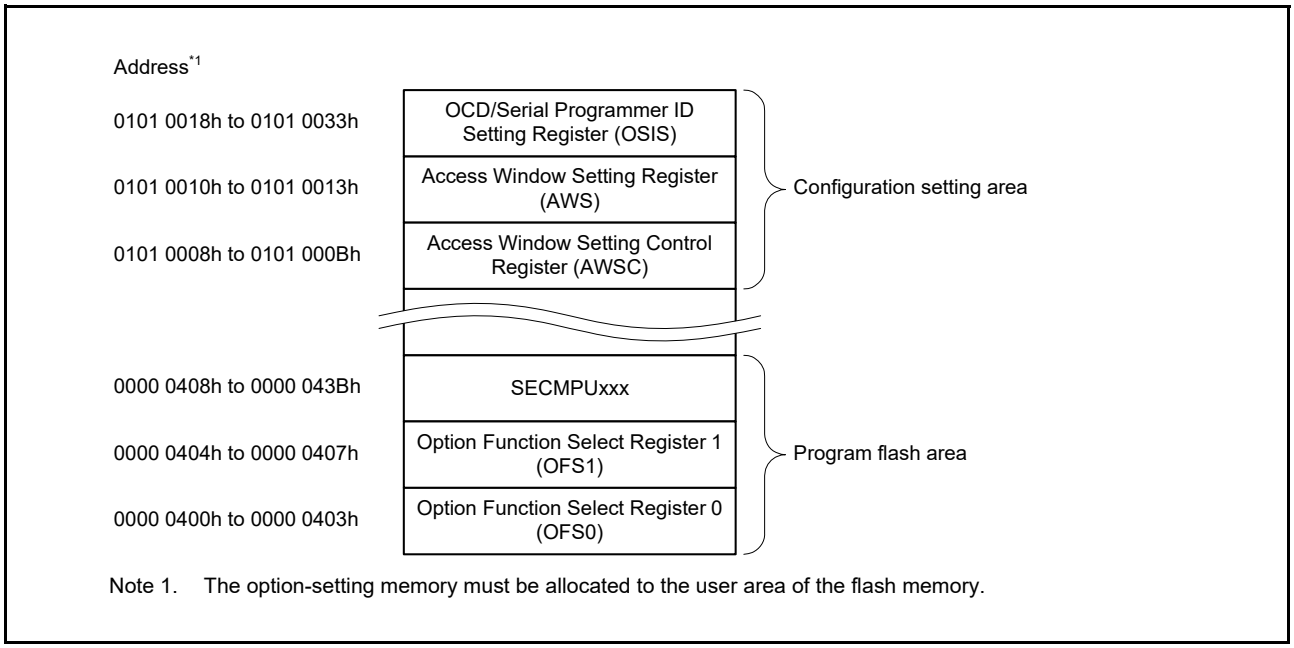


Figure 7.1 Option-setting memory area

7.2 Register Descriptions

7.2.1 Option Function Select Register 0 (OFS0)

Address(es): OFS0 0000 0400h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	WDTSTPCTL	—	WDTRSTIRQS	WDTRPSS[1:0]		WDTRPES[1:0]		WDTCKS[3:0]				WDTTOPS[1:0]		WDTSTRT	—
Value after reset:	The value set by the user*1														

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	IWDTSTPCTL	—	IWDTRSTIRQS	IWDTRPSS[1:0]		IWDTRPES[1:0]		IWDTCKS[3:0]				IWDTTOPS[1:0]		IWDTSTRT	—
Value after reset:	The value set by the user*1														

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	When read, this bit returns the written value. The write value should be 1.	R
b1	IWDTSTRT	IWDT Start Mode Select	0: Automatically activate IWDT after a reset (auto-start mode) 1: Disable IWDT.	R
b3, b2	IWDTTOPS[1:0]	IWDT Timeout Period Select	b3 b2 0 0: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh).	R
b7 to b4	IWDTCKS[3:0]	IWDT-Dedicated Clock Frequency Division Ratio Select	b7 b4 0 0 0 0: × 1 0 0 1 0: × 1/16 0 0 1 1: × 1/32 0 1 0 0: × 1/64 1 1 1 1: × 1/128 0 1 0 1: × 1/256. Other settings are prohibited.	R
b9, b8	IWDTRPES[1:0]	IWDT Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (no window end position setting).	R
b11, b10	IWDTRPSS[1:0]	IWDT Window Start Position Select	b11 b10 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (no window start position setting).	R
b12	IWDTRSTIRQS	IWDT Reset Interrupt Request Select	0: Enable non-maskable interrupt request or interrupt request 1: Enable reset.	R
b13	—	Reserved	When read, this bit returns the written value. The write value should be 1.	R
b14	IWDTSTPCTL	IWDT Stop Control	0: Continue counting 1: Stop counting when in Sleep mode, Snooze mode, or Software Standby mode.	R
b16, b15	—	Reserved	When read, these bits return the written value. The write value should be 1.	R
b17	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

Bit	Symbol	Bit name	Description	R/W
b19, b18	WDTTOPS[1:0]	WDT Timeout Period Select	b19 b18 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh).	R
b23 to b20	WDTCKS[3:0]	WDT Clock Frequency Division Ratio Select	b23 b20 0 0 0 1: PCLKB divided by 4 0 1 0 0: PCLKB divided by 64 1 1 1 1: PCLKB divided by 128 0 1 1 0: PCLKB divided by 512 0 1 1 1: PCLKB divided by 2048 1 0 0 0: PCLKB divided by 8192. Other settings are prohibited.	R
b25, b24	WDTRPES[1:0]	WDT Window End Position Select	b25 b24 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (No window end position setting).	R
b27, b26	WDTRPSS[1:0]	WDT Window Start Position Select	b27 b26 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (No window start position setting).	R
b28	WDTRSTIRQS	WDT Reset Interrupt Request Select	WDT Behavior Select: 0: NMI 1: Reset.	R
b29	—	Reserved	When read, this bit returns the written value. The write value should be 1.	R
b30	WDTSTPCTL	WDT Stop Control	0: Continue counting 1: Stop counting when entering Sleep mode.	R
b31	—	Reserved	When read, this bit returns the written value. The write value should be 1.	R

Note 1. The value in a blank product is FFFF FFFFh. 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 select the timeout period, 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 number of clock cycles that the IWDT takes to underflow after a refresh operation is determined by the combination of the IWDTCKS[3:0] and IWDTTOPS[1:0] bits.

See section 27, Independent Watchdog Timer (IWDT) for details.

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, or 1/256. Using this setting combined with the IWDTTOPS[1:0] bit setting, the IWDT counting period can be set from 128 to 524,288 IWDT clock cycles.

See section 27, Independent Watchdog Timer (IWDT) for details.

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

The IWDTRPES[1:0] bits select 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 depending on the setting in the IWDTTOPS[1:0] bits.

See section 27, Independent Watchdog Timer (IWDT) for details.

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

The IWDTRPSS[1:0] bits select 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 start and end positions becomes the period in which a refresh is possible. Refresh is not possible outside this period.

See section 27, Independent Watchdog Timer (IWDT) for details.

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.

See section 27, Independent Watchdog Timer (IWDT) for details.

IWDTSTPCTL bit (IWDT Stop Control)

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

See section 27, Independent Watchdog Timer (IWDT) for details.

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, the time it takes for the down counter to underflow, as 1,024, 4,096, 8,192, or 16,384 cycles of the frequency-divided clock set in the WDTCKS[3:0] bits. The number of PCLKB cycles that the counter takes to underflow after a refresh operation is determined by a combination of the WDTCKS[3:0] and WDTTOPS[1:0] bits.

See section 26, Watchdog Timer (WDT) for details.

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

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

See section 26, Watchdog Timer (WDT) for details.

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

The WDTRPES[1:0] bits specify the position where the window for 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 corresponding to 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.

See section 26, Watchdog Timer (WDT) for details.

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

The WDTRPSS[1:0] bits select 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 start and end positions becomes the period in which a refresh is possible. However, refresh is not possible outside this period. See section 26, Watchdog Timer (WDT) for details.

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. See section 26, Watchdog Timer (WDT) for details.

WDTSTPCTL bit (WDT Stop Control)

The WDTSTPCTL bit specifies whether to stop counting when entering Sleep mode.
See section 26, Watchdog Timer (WDT) for details.

7.2.2 Option Function Select Register 1 (OFS1)

Address(es): OFS1 0000 0404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	HOCOFRQ1[2:0]			—	—	—	HOCO EN	—	—	VDSEL1[2:0]			LVDAS	—	—
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	When read, these bits return the written value. The write value should be 1.	R
b2	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
b5 to b3	VDSEL1[2:0]	Voltage Detection 0 Level Select	b5 b3 0 0 0: Selects 3.84 V 0 0 1: Selects 2.82 V 0 1 0: Selects 2.51 V 0 1 1: Selects 1.90 V 1 0 0: Selects 1.70 V. Other settings are prohibited.	
b7, b6	—	Reserved	When read, these bits return the written value. The write value should be 1.	R
b8	HOCOEN	HOCO Oscillation Enable	0: Enable HOCO oscillation after a reset 1: Disable HOCO oscillation after a reset.	R
b11 to b9	—	Reserved	When read, these bit return the written value. The write value should be 1.	R
b14 to b12	HOCOFRQ1[2:0]	HOCO Frequency Setting 1	b14 b12 0 0 0: 24 MHz 0 1 0: 32 MHz 1 0 0: 48 MHz 1 0 1: 64 MHz. Other settings are prohibited.	R
b31 to b15	—	Reserved	When read, these bits return the written value. The write value should be 1.	R

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

LVDAS bit (Voltage Detection 0 Circuit Start)

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

VDSEL1[2:0] bits (Voltage Detection 0 Level Select)

The VDSEL1[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 is enabled or disabled 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, you must set the OFS1.HOCOFRQ1 bit to an optimum value.

After a reset release, operation is in the low-voltage mode, and therefore HOCOCR.HCSTP must be immediately set to 0.

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.

7.2.3 MPU Registers

Table 7.1 indicates the registers related to the MPU function. See section 16, Memory Protection Unit (MPU) for details.

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 16, Memory Protection Unit (MPU) to set the proper data.

Table 7.1 MPU registers

Register name	Symbol	Function	Address	Size (byte)
Security MPU Program Counter Start Address Register 0	SECMPUPCS0	Specifies the security fetch region of code flash or SRAM	0000 0408h	4
Security MPU Program Counter End Address Register 0	SECMPUPCE0	Specifies the security fetch region of code flash or SRAM	0000 040Ch	4
Security MPU Program Counter Start Address Register 1	SECMPUPCS1	Specifies the security fetch region of code flash or SRAM	0000 0410h	4
Security MPU Program Counter End Address Register 1	SECMPUPCE1	Specifies the security fetch region of code flash or SRAM	0000 0414h	4
Security MPU Region 0 Start Address Register	SECMPUS0	Specifies the security program and code flash data	0000 0418h	4
Security MPU Region 0 End Address Register	SECMPUE0	Specifies the security program and code flash data	0000 041Ch	4
Security MPU Access Control Register	SECMPUAC	Specifies the security enabled/disabled region	0000 0438h	4

7.2.4 Access Window Setting Control Register (AWSC)

Address(es): AWSC 0101 0008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	FSPR	—	—	—	—	—	BTFLG	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user															

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	Reserved	When read, these bits return the written value. The write value should be 1.	R
b8	BTFLG	Startup Area Select Flag	This bit specifies whether the address of the startup area is exchanged for the boot swap function. 0: The first 8-KB area (0000 0000h to 0000 1FFFh) and second 8-KB area (0000 2000h to 0000 3FFFh) are exchanged 1: The first 8-KB area (0000 0000h to 0000 1FFFh) and second 8-KB area (0000 2000h to 0000 3FFFh) are not exchanged.	R
b13 to b9	—	Reserved	When read, these bits return the value written by the user. The write value should be 1.	R

Bit	Symbol	Bit name	Description	R/W
b14	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 temporary boot swap control. If this bit is set to 0, it cannot be changed to 1. 0: Executing the configuration setting command for programming the access window (FAWE[11:0], FAWS[11:0]) and the Startup Area Select Flag (BTFLG) is invalid. 1: Executing the configuration setting command for programming the access window (FAWE[11:0], FAWS[11:0]) and the Startup Area Select Flag (BTFLG) is valid.	R
b31 to b15	—	Reserved	When read, these bits return the written value. The write value should be 1.	R

7.2.5 Access Window Setting Register (AWS)

Address(es): AWS 0101 0010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	FAWE[11:0] *1											
Value after reset:	The value set by the user															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	FAWS[11:0] *1											
Value after reset:	The value set by the user															

Bit	Symbol	Bit name	Description	R/W
b11 to b0	FAWS[11:0]	Access Window Start Block Address*1	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 [21:10].	R
b15 to b12	—	Reserved	When read, these bits return the written value. The write value should be 1.	R
b27 to b16	FAWE[11:0]	Access Window End Block Address*1	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 P/E acceptable region defined by the access window. The block address specifies the first address of the block and consists of the address bits [21:10].	R
b31 to b28	—	Reserved	When read, these bits return the written value. The write value should be 1.	R

Note 1. The write value should be 0 for FAWE[0] and FAWS[0].

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[11:0] and FAWE[11:0] bits.

The following describes how to set the FAWS[11:0] bits and the FAWE[11:0] bits.

FAWE[11:0] = FAWS[11:0]: The P/E command is allowed to execute in the full program flash area.

FAWE[11:0] > FAWS[11:0]: The P/E command is only allowed to execute in the window from the block pointed to by the FAWS bits to the block one lower than the block pointed to by the FAWE[11:0] bits.

FAWE[11:0] < FAWS[11:0]: The P/E command is not allowed to execute in the program flash area.

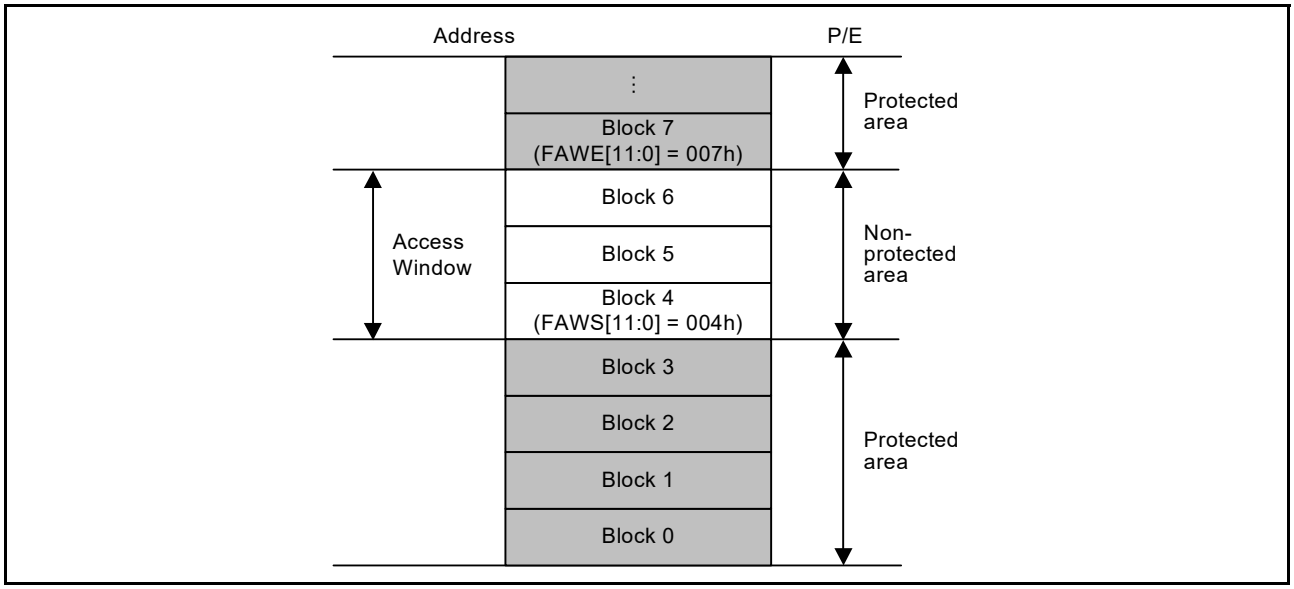


Figure 7.2 Access window overview

7.2.6 OCD/Serial Programmer ID Setting Register (OSIS)

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. This register checks whether a code transmitted from the OCD/serial programmer matches the ID code in the option-setting memory. When the ID codes match, the connection of the OCD/serial programmer is permitted. When ID codes do not match, connection with the OCD/serial programmer is not possible. The OSIS register must be set in 32-bit units.

Address(es): OSIS 0101 0018h, OSIS 0101 0020h, OSIS 0101 0028h, OSIS 0101 0030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	The value set by the user															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	The value set by the user															

These fields store 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 method of authentication to use with the host. Table 7.2 shows how the ID code determines the method of authentication.

Table 7.2 Specifications for ID code protection (1 of 2)

Operating mode on boot up	ID code	State of protection	Operations on connection to programmer or on-chip debugger
Serial programming mode (SCI/USB boot mode) On-chip debug mode (JTAG/SWD Boot mode)	FFh, …, FFh (all bytes FFh)	Protection disabled	The ID code is not checked, the ID code always matches, and connection to the programmer or on-chip debugger is permitted

Table 7.2 Specifications for ID code protection (2 of 2)

Operating mode on boot up	ID code	State of protection	Operations on connection to programmer or on-chip debugger
Serial programming mode (SCI/USB boot mode) On-chip debug mode (JTAG/SWD Boot mode)	Bit [127] = 1, bit [126] = 1, and at least one of the 16 bytes is not FFh	Protection enabled	Matching ID code indicates that an authentication is complete and connection to the programmer or the on-chip debugger is permitted. Mismatching ID code indicates a transition to the ID code protection wait state. When the ID code sent from the programmer or the on-chip debugger is ALeRASE in ASCII code (414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFF h), the content of the user flash (code and data) area and configuration area are erased. However, forced erasure is not executed when the FSPR bit is 0.
	Bit [127] = 1 and bit [126] = 0	Protection enabled	Matching ID code indicates that authentication is complete and connection with the 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, and the connection with the programmer or the on-chip debugger is prohibited.

7.3 Setting Option-Setting Memory

7.3.1 Allocation of Data in Option-Setting Memory

Programming data is allocated to the addresses in the option-setting memory shown in Figure 7.1. 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.

7.3.2 Setting Data for Programming Option-Setting Memory

Allocating data according to the procedure described in section 7.3.1, 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.

See section 48, Flash Memory for details on the programming command, the configuration setting command, and the startup area select function.

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

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

The MCU provides two setting procedures as follows:

- Read the data allocated as described in section 7.3.1, 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 allocated as described in section 7.3.1, Allocation of Data in Option-Setting Memory.

7.4 Usage Note

7.4.1 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 in the reserved areas and all reserved bits. If 0 is written to these bits, normal operation cannot be guaranteed.

8. Low Voltage Detection (LVD)

8.1 Overview

The Low Voltage Detection (LVD) module monitors the voltage level input to the VCC pin, and the detection level can be selected using a software program. The LVD module consists of three separate voltage level detectors, 0, 1, and 2, which measure the voltage level input to the VCC pin. LVD voltage detection registers allow your application to configure the detection of VCC changes at various voltage thresholds.

Each voltage level detector has a voltage monitor associated with it, called voltage monitor 0, 1, and 2. Voltage monitor registers configure the LVD to trigger an interrupt, event link output, or reset when the thresholds are crossed.

Table 8.1 lists the LVD specifications. Figure 8.1 shows a block diagram of voltage detectors 0,1, and 2, Figure 8.2 shows a block diagram of the voltage monitor 1 interrupt/reset circuit, and Figure 8.3 shows a block diagram of the voltage monitor 2 interrupt/reset circuit.

Table 8.1 LVD specifications

Item		Voltage monitor 0	Voltage monitor 1	Voltage monitor 2
VCC monitoring	Monitored voltage	V_{det0}	V_{det1}	V_{det2}
	Detected event	Voltage falls below V_{det0}	Voltage rises or falls past V_{det1}	Voltage rises or falls past V_{det2}
	Detection voltage	Selectable from five different levels in the OFS1.VDSEL1[2:0] bits	Selectable from 16 different levels in the LVDLVLR.LVD1LVL[4:0] bits	Selectable from four different levels in the LVDLVLR.LVD2LVL[2:0] bits
	Monitor 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 upon 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 VCC > V_{det2} or V_{det2} > VCC
	Interrupt	No interrupt	Voltage monitor 1 interrupt	Voltage monitor 2 interrupt
			Non-maskable interrupt or maskable interrupt selectable	Non-maskable interrupt or maskable interrupt selectable
			Interrupt request issued when V_{det1} > VCC or VCC > V_{det1}	Interrupt request issued when V_{det2} > VCC or VCC > V_{det2}
Event linking		None	Available Output of event signals on detection of V_{det1} crossings	Available Output of event signals on detection of V_{det2} crossings

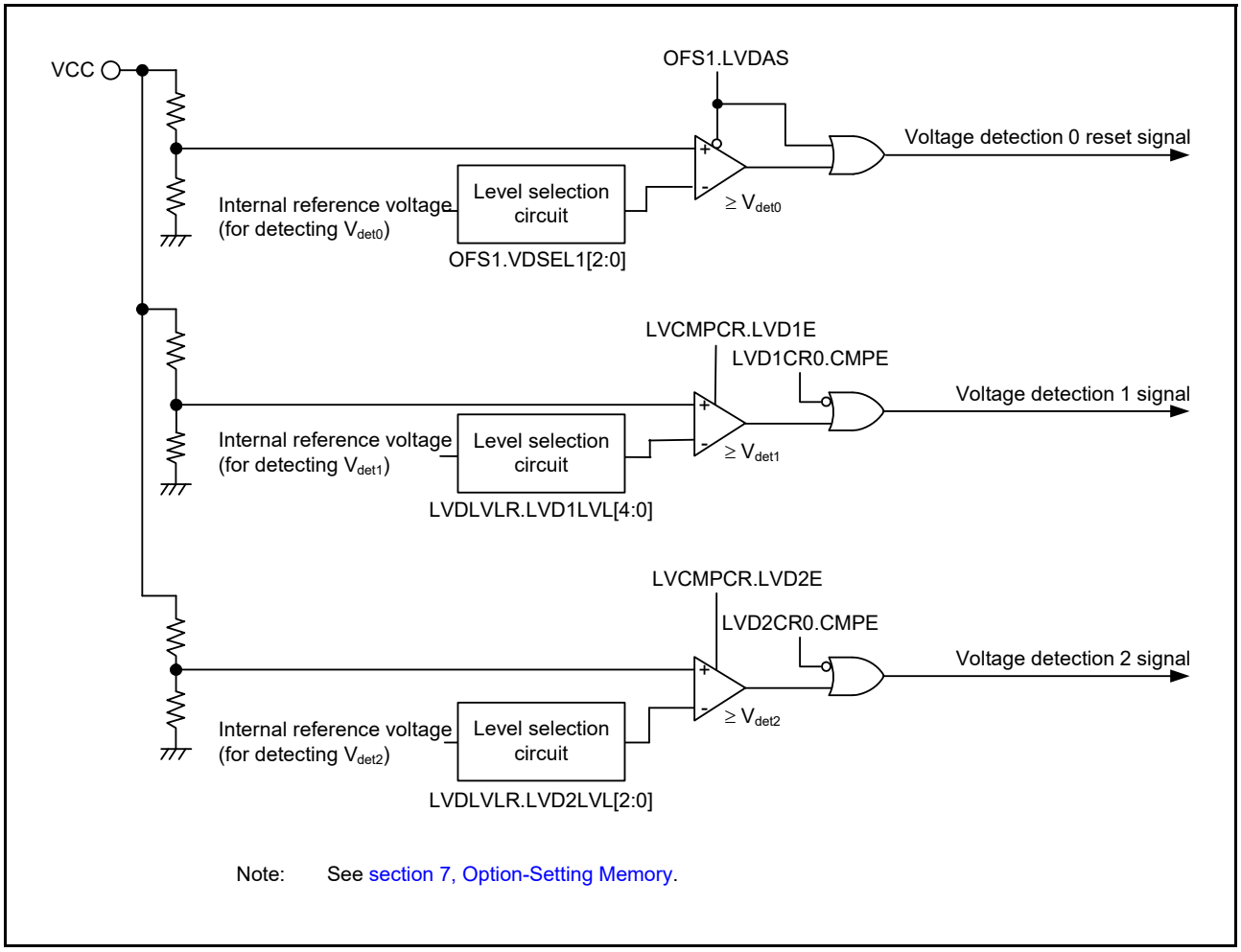


Figure 8.1 Voltage detection 0, 1, and 2 block diagram

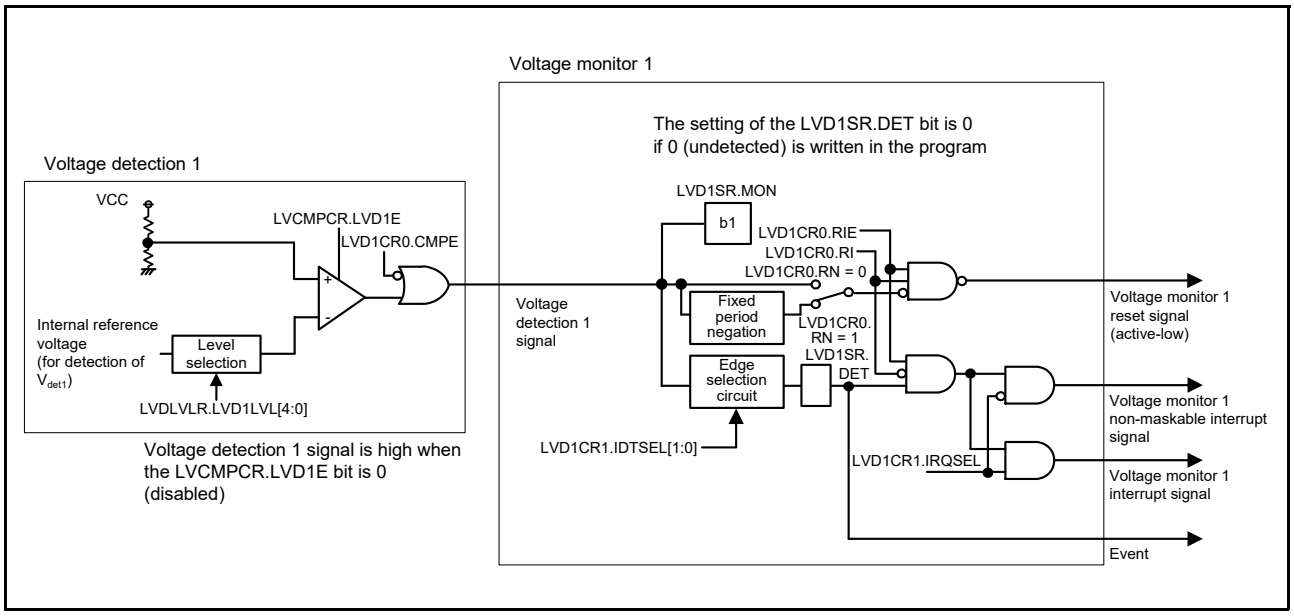


Figure 8.2 Voltage monitor 1 interrupt/reset circuit block diagram

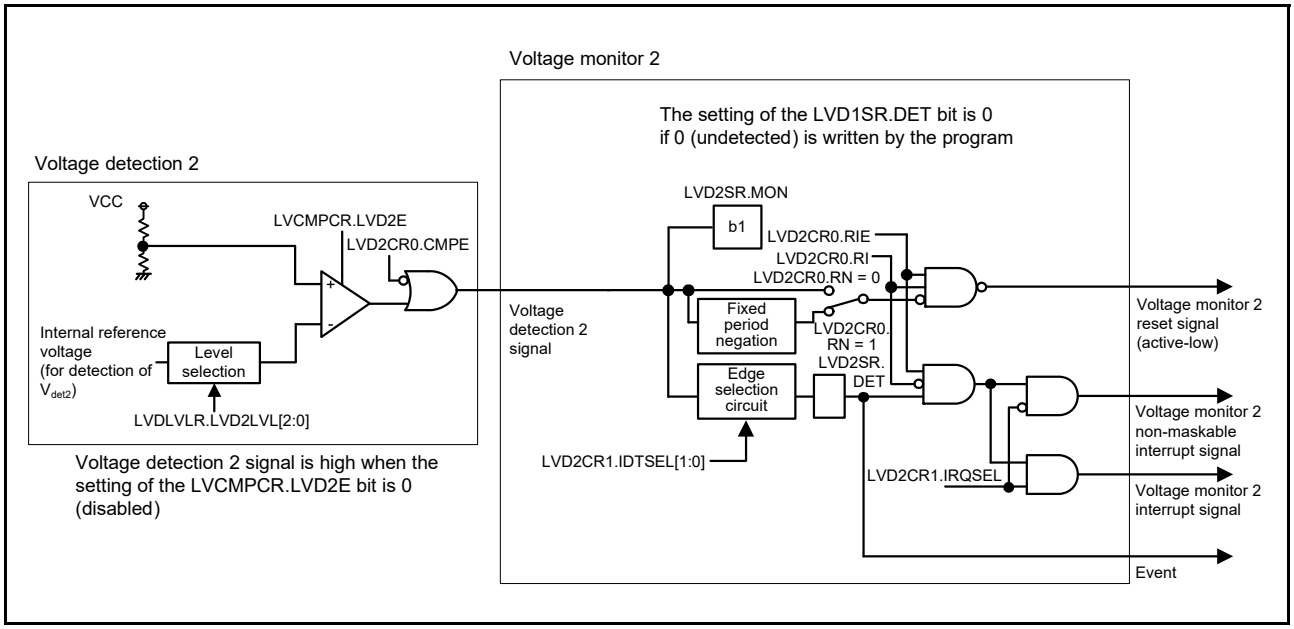


Figure 8.3 Voltage monitor 2 interrupt/reset circuit block diagram

8.2 Register Descriptions

8.2.1 Voltage Monitor 1 Circuit Control Register 1 (LVD1CR1)

Address(es): SYSTEM.LVD1CR1 4001 E0E0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	IRQSEL	IDTSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b1, b0	IDTSEL[1:0]	Voltage Monitor 1 Interrupt Generation Condition Select	b1 b0 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
b2	IRQSEL	Voltage Monitor 1 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt*1.	R/W
b7 to b3	—	Reserved	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. When enabling maskable interrupts, do not change the NMIER.LVD1EN bit value in the ICU from the reset state.

8.2.2 Voltage Monitor 1 Circuit Status Register (LVD1SR)

Address(es): SYSTEM.LVD1SR 4001 E0E1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MON	DET
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b0	DET	Voltage Monitor 1 Voltage Change Detection Flag	0: Not detected 1: V_{det1} passage detected.	R(/W) *1
b1	MON	Voltage Monitor 1 Signal Monitor Flag	0: VCC < V_{det1} 1: VCC ≥ V_{det1} or MON is disabled.	R
b7 to b2	—	Reserved	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 Change 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).

The DET flag must be set to 0 after LVD1CR0.RIE is set to 0 (disabled). LVD1CR0.RIE can be set to 1 (enabled) after 2 or more PCLKB cycles have elapsed.

A wait time of 2 or more PCLKB cycles might be required, depending on the number of PCLKB cycles required to read a given I/O register.

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).

8.2.3 Voltage Monitor 2 Circuit Control Register 1 (LVD2CR1)

Address(es): SYSTEM.LVD2CR1 4001 E0E2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	IRQSEL	IDTSEL [1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b1, b0	IDTSEL [1:0]	Voltage Monitor 2 Interrupt Generation Condition Select	b1 b0 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
b2	IRQSEL	Voltage Monitor 2 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt*1.	R/W
b7 to b3	—	Reserved	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. When enabling maskable interrupts, do not change the value of the NMIER.LVD1EN bit in the ICU from the reset state.

8.2.4 Voltage Monitor 2 Circuit Status Register (LVD2SR)

Address(es): SYSTEM.LVD2SR 4001 E0E3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MON	DET
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b0	DET	Voltage Monitor 2 Voltage Change Detection Flag	0: Not detected 1: V_{det2} passage detected.	R/W*1
b1	MON	Voltage Monitor 2 Signal Monitor Flag	0: VCC < V_{det2} 1: VCC ≥ V_{det2} or MON is disabled.	R
b7 to b2	—	Reserved	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 Change 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).

The DET flag must be set to 0 after LVD2CR0.RIE is set to 0 (disabled). LVD2CR0.RIE can be set to 1 (enabled) after 2 or more PCLKB cycles have elapsed.

A wait time of 2 or more PCLKB cycles might be required, depending on the number of PCLKB cycles required to read a given I/O register.

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).

8.2.5 Voltage Monitor Circuit Control Register (LVCMPCR)

Address(es): SYSTEM.LVCMPCR 4001 E417h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	LVD2E	LVD1E	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	LVD1E	Voltage Detection 1 Enable	0: Disable voltage detection 1 circuit 1: Enable voltage detection 1 circuit.	R/W
b6	LVD2E	Voltage Detection 2 Enable	0: Disable voltage detection 2 circuit 1: Enable voltage detection 2 circuit.	R/W
b7	—	Reserved	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)

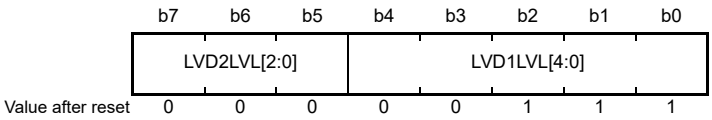
When using voltage detection 1 interrupt/reset or the LVD1SR.MON bit, set the LVD1E bit to 1. The voltage detection 1 circuit starts when td(E-A) elapses after the LVD1E bit value is changed from 0 to 1.

LVD2E bit (Voltage Detection 2 Enable)

When using voltage detection 2 interrupt/reset or the LVD2SR.MON bit, set the LVD2E bit to 1. The voltage detection 2 circuit starts when td(E-A) elapses after the LVD2E bit value is changed from 0 to 1.

8.2.6 Voltage Detection Level Select Register (LVDLVLR)

Address(es): SYSTEM.LVDLVLR 4001 E418h



Bit	Symbol	Bit name	Description	R/W
b4 to b0	LVD1LVL[4:0]	Voltage Detection 1 Level Select (Standard voltage during fall in voltage)	b4 b0 0 0 0 0 0: 4.29 V (V_{det1_0}) 0 0 0 0 1: 4.14 V (V_{det1_1}) 0 0 0 1 0: 4.02 V (V_{det1_2}) 0 0 0 1 1: 3.84 V (V_{det1_3}) 0 0 1 0 0: 3.10 V (V_{det1_4}) 0 0 1 0 1: 3.00 V (V_{det1_5}) 0 0 1 1 0: 2.90 V (V_{det1_6}) 0 0 1 1 1: 2.79 V (V_{det1_7}) 0 1 0 0 0: 2.68 V (V_{det1_8}) 0 1 0 0 1: 2.58 V (V_{det1_9}) 0 1 0 1 0: 2.48 V (V_{det1_A}) 0 1 0 1 1: 2.20 V (V_{det1_B}) 0 1 1 0 0: 1.96 V (V_{det1_C}) 0 1 1 0 1: 1.86 V (V_{det1_D}) 0 1 1 1 0: 1.75 V (V_{det1_E}) 0 1 1 1 1: 1.65 V (V_{det1_F}). Other settings are prohibited.	R/W
b7 to b5	LVD2LVL[2:0]	Voltage Detection 2 Level Select (Standard voltage during fall in voltage)	b7 b5 0 0 0: 4.29 V (V_{det2_0}) 0 0 1: 4.14 V (V_{det2_1}) 0 1 0: 4.02 V (V_{det2_2}) 0 1 1: 3.84 V (V_{det2_3}) 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Setting prohibited.	R/W

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

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.

8.2.7 Voltage Monitor 1 Circuit Control Register 0 (LVD1CR0)

Address(es): SYSTEM.LVD1CR0 4001 E41Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	RN	RI	—	—	—	CMPE	—	RIE
Value after reset:	1	0	0	0	x	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RIE	Voltage Monitor 1 Interrupt/Reset Enable	0: Disable 1: Enable.	R/W
b1	—	Reserved	The read value is 0. The write value should be 0.	R/W
b2	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
b3	—	Reserved	The read value is undefined. The write value should be 1.	R/W
b5 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	RI	Voltage Monitor 1 Circuit Mode Select	0: Generate voltage monitor 1 interrupt on V_{det1} passage 1: Enable voltage monitor 1 reset when the voltage falls to and below V_{det1}.	R/W
b7	RN	Voltage Monitor 1 Reset Negate Select	0: Negate after a stabilization time (tLVD1) when VCC > V_{det1} is detected 1: Negate after a stabilization time (tLVD1) 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 the voltage monitor 1 interrupt/reset. Set this bit to ensure that neither a voltage monitor 1 interrupt nor a voltage monitor 1reset is generated during programming or erasure of the flash memory.

RN bit (Voltage Monitor 1 Reset Negate Select)

If the RN bit is to be 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). Additionally, if a transition to Software Standby is to be made, the only possible value for the RN bit is 0 (negation follows a stabilization time when VCC > V_{det1} is detected). Do not set the RN bit to 1 (negation follows a stabilization time on assertion of the LVD1 reset signal) when this is the case.

8.2.8 Voltage Monitor 2 Circuit Control Register 0 (LVD2CR0)

Address(es): SYSTEM.LVD2CR0 4001 E41Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	RN	RI	—	—	—	CMPE	—	RIE
Value after reset:	1	0	0	0	x	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RIE	Voltage Monitor 2 Interrupt/Reset Enable	0: Disable 1: Enable.	R/W
b1	—	Reserved	The read value is 0. The write value should be 0.	R/W
b2	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
b3	—	Reserved	The read value is undefined. The write value should be 1.	R/W
b5 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b6	RI	Voltage Monitor 2 Circuit Mode Select	0: Generate voltage monitor 2 interrupt on V_{det2} passage 1: Enable voltage monitor 2 reset when the voltage falls to or below V_{det2}.	R/W
b7	RN	Voltage Monitor 2 Reset Negate Select	0: Negate after a stabilization time (tLVD2) after VCC > V_{det2} is detected 1: Negate after a stabilization time (tLVD2) 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. Set this bit to ensure that neither a voltage monitor 2 interrupt nor a voltage monitor 2 reset is generated during programming or erasure of the flash memory.

RN bit (Voltage Monitor 2 Reset Negate Select)

If the RN bit is to be set to 1 (negation follows a stabilization time after the assertion of the LVD2 reset signal), set the MOCOCR.MCSTP bit to 0 (the MOCO operates). Additionally, if a transition to Software Standby is to be made, 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 the assertion of the LVD2 reset signal) when this is the case.

8.3 VCC Input Voltage Monitor

8.3.1 Monitoring V_{det0}

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

8.3.2 Monitoring V_{det1}

Table 8.2 lists the procedures to set up monitoring against V_{det1}. After the settings are complete, the comparison results from the voltage monitor 1 can be monitored with the LVD1SR.MON flag.

Table 8.2 Procedures to set up monitoring against V_{det1}

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

8.3.3 Monitoring V_{det2}

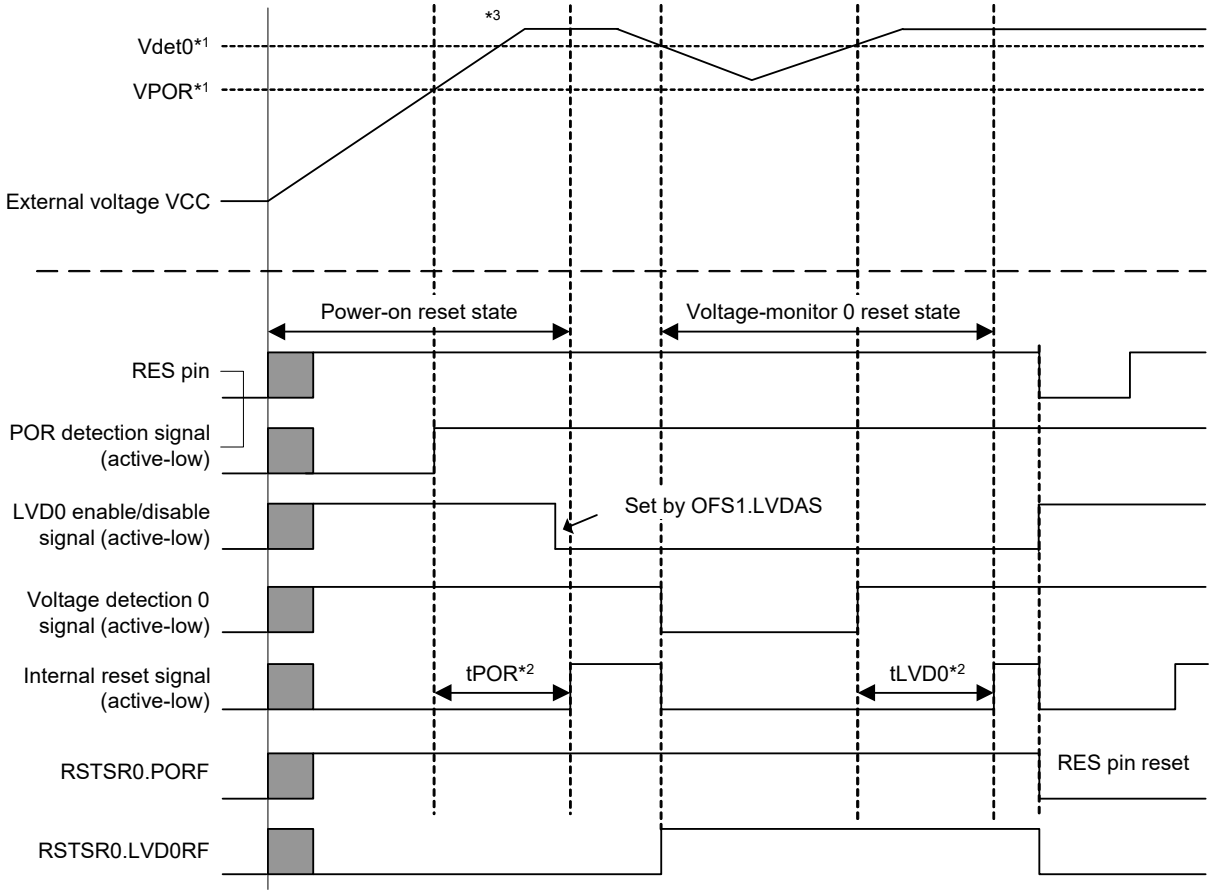
Table 8.3 lists the procedures to set up monitoring against V_{det2}. After the settings are complete, the comparison results from the voltage monitor 2 can be monitored with the LVD2SR.MON flag.

Table 8.3 Procedures to set up monitoring against V_{det2}

Step		Monitoring the comparison results from voltage monitor 2
Setting 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 td(E-A) for the LVD operation stabilization after LVD is enabled.
Enabling output	5	Set LVD2CR0.CMPE = 1 to enable output of the results of comparison from voltage monitor 2.

8.4 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 8.4 shows an example operation of a voltage monitor 0 reset.



Note: For details of the electrical characteristics, see section 52, 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} indicates the period of a power-on reset and t_{LVD0} indicates the period of a voltage monitor 0 reset.
Note 3. At power-on, VCC should be brought to the minimum guaranteed voltage before releasing the POR reset.

Figure 8.4 Example operation of voltage monitor 0 reset

8.5 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 8.4 shows the procedures for setting bits related to the voltage monitor 1 interrupt/reset so that voltage monitoring operates. Table 8.5 shows the procedure for setting bits related to the voltage monitor 1 interrupt/reset so that voltage monitoring stops. Figure 8.5 shows an example of operations for a voltage monitor 1 interrupt. For the operation of the voltage monitor 1 reset, see Figure 6.2 in section 6, Resets.

When using the voltage monitor 1 circuit in Software Standby, set up the circuit with the following procedures.

(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 8.4 Procedures for setting bits related to the voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitor operates

Step		Voltage monitor 1 interrupt (voltage monitor 1 ELC event output)	voltage monitor 1 reset
Setting 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 by setting the LVDLVLR.LVD1LVL[3:0] bits.	
	3	Set LVCMPCR.LVD1E = 1 to enable the voltage detection 1 circuit.	
	4	Wait for at least td(E-A) for the LVD operation stabilization after LVD is enabled.*1	
Setting the voltage monitor 1 interrupt/ 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 the reset negation by setting the LVD1CR0.RN bit.
	6	• Select the timing of interrupt requests by setting the LVD1CR1.IDTSEL[1:0] bits • Select the type of interrupt by setting 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 of step 4. For details of td(E-A), see section 52, Electrical Characteristics.
Note 2. Step 8 is not required if only the ELC event signal is to be output.

Table 8.5 Procedures for setting bits related to the voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitor stops

Step		Voltage monitor 1 interrupt (voltage monitor 1 ELC event output), voltage monitor 1 reset
Settings to stop enabling of 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 voltage monitor 1 reset setting is to be made again after it is used and stopped once, omit the following steps in the procedures for stopping and setting, depending on the conditions:

- Setting or stopping the voltage detection 1 circuit is not required if the settings for the voltage detection 1 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 8.5 shows an example of the voltage monitor 1 interrupt operation.

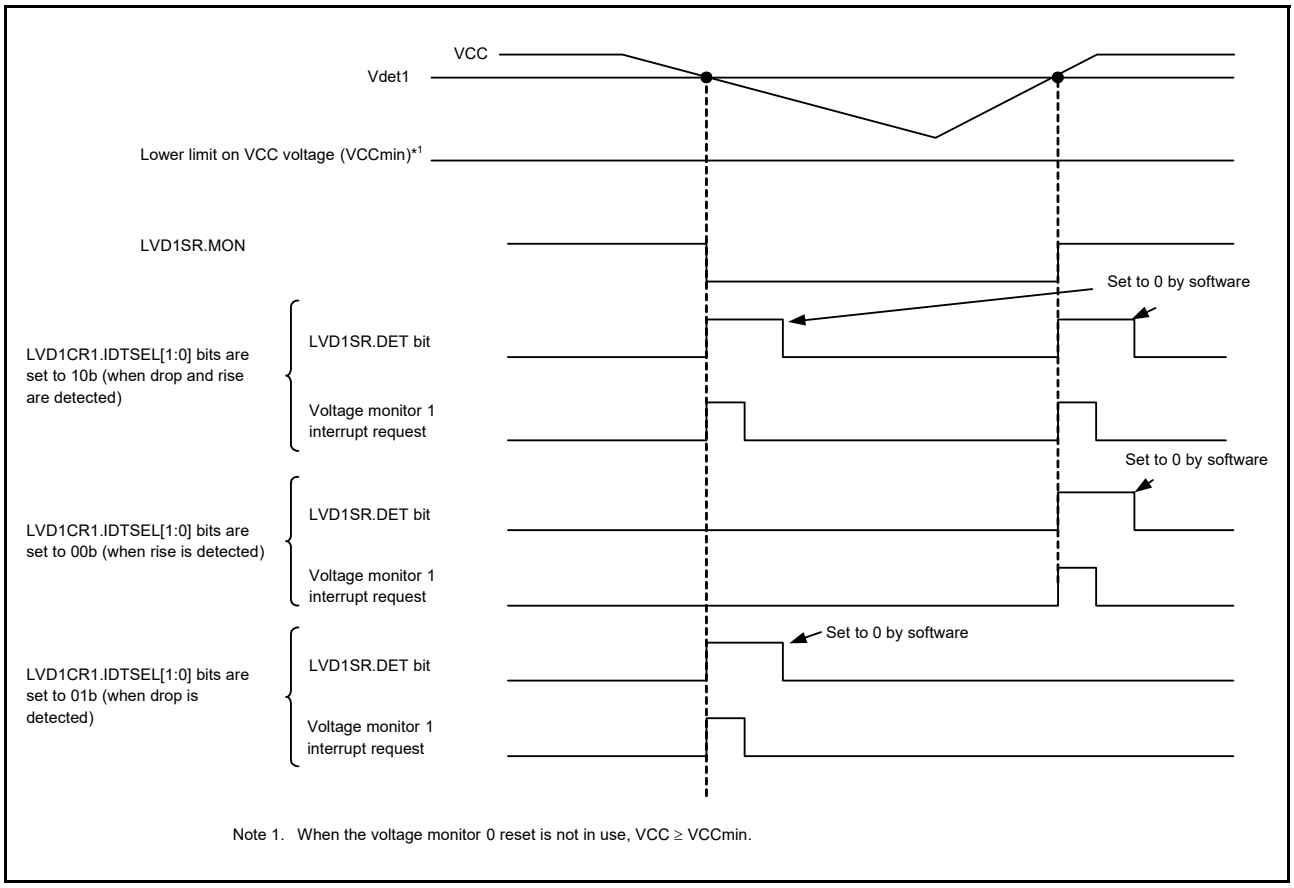


Figure 8.5 Voltage monitor 1 interrupt operation example

8.6 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 8.6 shows the procedures for setting bits related to the voltage monitor 2 interrupt and voltage monitor 2 reset so that the voltage monitor operates. Table 8.7 shows the procedure for setting bits related to the voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitor stops. Figure 8.6 shows an example of operation of the voltage monitor 2 interrupt. For the operation of the voltage monitor 2 reset, see Figure 6.2 in section 6, Resets.

When using the voltage monitor 2 circuit in Software Standby, set up the voltage monitor 2 circuit using the following procedures:

(1) Setting in Software Standby mode

- When VCC > V_{det2} is detected, clear the LVDD2CR0.RN bit (LVD2CR0.RN = 0) following a stabilization time.

Table 8.6 Procedures for setting bits related to voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitor operates (1 of 2)

Step		Voltage monitor 2 interrupt (voltage monitor 2 ELC event output)	Voltage monitor 2 reset
Setting 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 by setting the LVDLVLR.LVD2LVL[2:0] bits.	
	3	Set LVCMPCR.LVD2E = 1 to enable the voltage detection 2 circuit.	
	4	Wait for at least td(E-A) for LVD operation stabilization after LVD is enabled.*1	

Table 8.6 Procedures for setting bits related to voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitor operates (2 of 2)

Step		Voltage monitor 2 interrupt (voltage monitor 2 ELC event output)	Voltage monitor 2 reset
Setting 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 the reset negation by setting the LVD2CR0.RN bit.
	6	• Select the timing of interrupt requests by setting the LVD2CR1.IDTSEL[1:0] bits • Select the type of interrupt by setting 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 of step 4. For details of td(E-A), see section 52, Electrical Characteristics.
Note 2. Step 8 is not required if only the ELC event signal is to be output.

Table 8.7 Procedures for setting bits related to voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitor stops

Step		Voltage monitor 2 interrupt (voltage monitor 2 ELC event output), voltage monitor 2 reset
Settings to stop enabling of 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 1 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, omit the following steps in the procedures for stopping and setting, depending on the conditions:

- Setting or stopping the voltage detection 2 circuit 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 do not change.

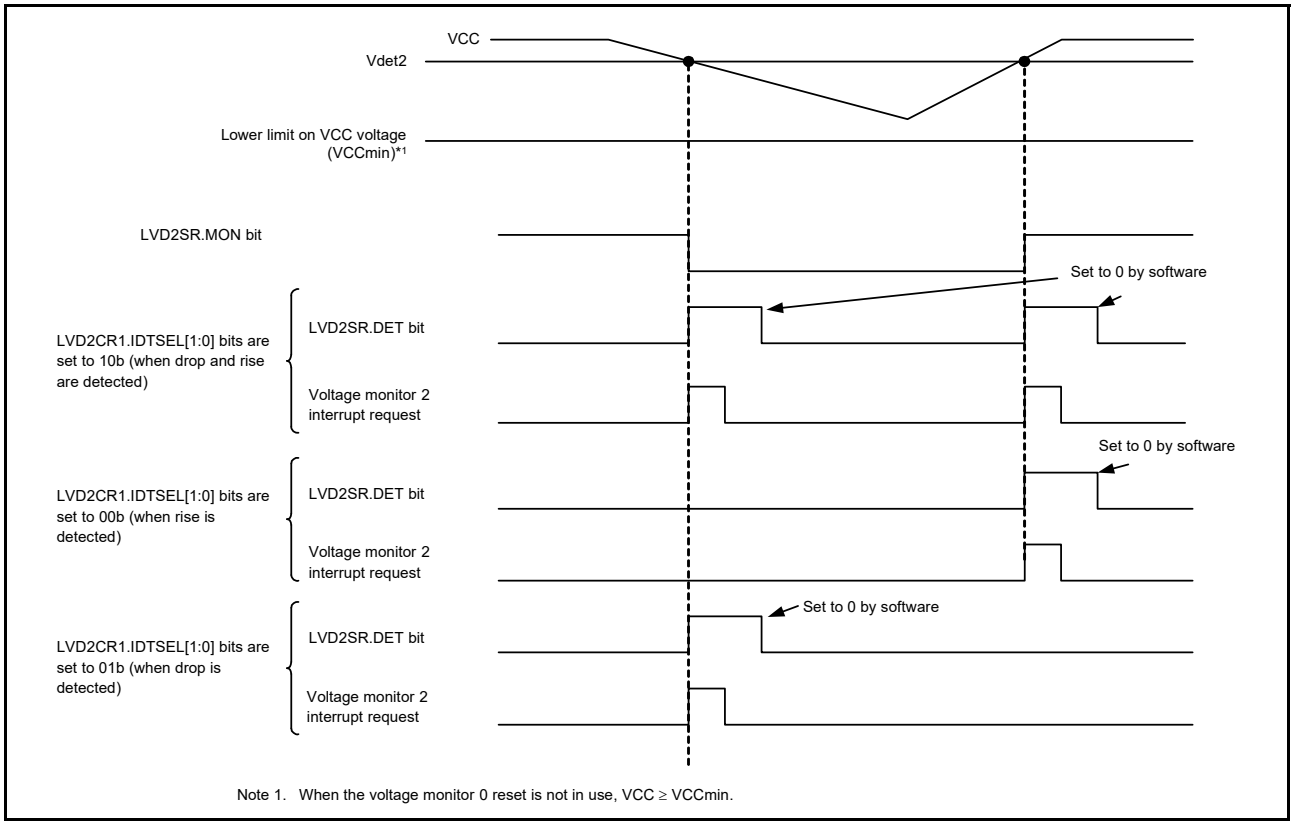


Figure 8.6 Example of voltage monitor 2 interrupt operation

8.7 Event Link 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.

8.7.1 Interrupt Handling and Event Linking

The LVD provides bits to individually 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 (LVD1CR0.RIE or LVD2CR0.RIE) is output to the CPU.

On the other hand, as soon as an interrupt source is generated, the 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. The event signals for the ELC in Software Standby mode are output as follows:

- When a V_{det1} or V_{det2} passage event 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.

9. Clock Generation Circuit

9.1 Overview

The MCU provides a clock generation circuit.

Table 9.1 and Table 9.2 list clock generation circuit specifications. Figure 9.1 shows the block diagram, and Table 9.3 lists the I/O pins.

Table 9.1 Clock Generation Circuit specifications for the clock sources

Clock source	Description	Specification
Main clock oscillator (MOSC)	Resonator frequency	• 1 MHz to 20 MHz (Up to 5.5 V) • 1 MHz to 8 MHz (Up to 2.4 V) • 4, 6, 8, 12 MHz (USB boot mode).
	External clock input frequency	Up to 20 MHz
	External resonator or additional circuit: ceramic resonator, crystal	Available
	Connection pins: EXTAL, XTAL	
	Drive capability switching	
	Oscillation stop detection function	
Sub-clock oscillator (SOSC)	Resonator frequency	32.768 kHz
	External resonator or additional circuit: crystal resonator	Available
	Connection pins: XCIN, XCOUT	
	Drive capability switching	
PLL circuit	Input clock source	MOSC
	Input frequency	4 MHz to 12.5 MHz
	Frequency multiplication ratio	Selectable from 8 to 31 (1 step) (multiplication frequency is up to 64 MHz)
	Output pulse frequency division ratio	Selectable from 1, 2, and 4
	PLL output frequency	• 24 MHz to 64 MHz (output frequency division ratio: 2) • 24 MHz to 32 MHz (output frequency division ratio: 4).
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	Not available
External clock input for JTAG (TCK)	Input clock frequency	Up to 12.5 MHz
External clock input for SWD (SWCLK)	Input clock frequency	Up to 12.5 MHz

Table 9.2 Clock Generation Circuit Specifications (internal clock)

Parameter	Clock source	Clock supply	Specification
System clock (ICLK)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	CPU, DTC, DMAC, Flash, SRAM	Up to 48 MHz Division ratios: 1, 2, 4, 8, 16, 32, 64
Peripheral module clock A (PCLKA)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	Peripheral modules (QSPI, SPI, SCI, SCE5, SDHI, CRC, IrDA, GPT bus-clock)	Up to 48 MHz Division ratios: 1, 2, 4, 8, 16, 32, 64
Peripheral module clock B (PCLKB)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	Peripheral modules (DAC12, IIC, SSI, DOC, CAC, CAN, AGT, POEG, CTSU, ELC, I/O Ports, RTC, WDT, IWDT, ADC14, KINT, USBFS, ACMPLP, ACMPHS, and SLCDC)	Up to 32 MHz Division ratios: 1, 2, 4, 8, 16, 32, 64
Peripheral module clock C (PCLKC)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	Peripheral module (ADC14 conversion clock)	Up to 64 MHz Division ratios: 1, 2, 4, 8, 16, 32, 64
Peripheral module clock D (PCLKD)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	Peripheral module (GPT count clock)	Up to 64 MHz Division ratios: 1, 2, 4, 8, 16, 32, 64
FlashIF clock (FCLK)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	Flash interface	1 MHz to 32 MHz (P/E) Up to 32 MHz (Read) Division ratios: 1, 2, 4, 8, 16, 32, 64
External bus clock (BCLK)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	External bus	Up to 24 MHz Division ratios: 1, 2, 4, 8, 16, 32, 64
EBCLK pin output (EBCLK)	BCLK or 1/2 BCLK	EBCLK pin	Up to 12 MHz Division ratios: 1 or 2
USB clock (UCLK)	PLL	USBFS	48 MHz
CAN clock (CANMCLK)	MOSC	CAN	1 MHz to 20 MHz
Segment LCD clock (LCDSRCCLK)	MOSC, SOSC, HOCO, MOCO, LOCO	SLCDC	Up to 64 MHz
AGT clock (AGTSCLK/AGTLCLK)	SOSC, LOCO	AGT	32.768 kHz
CAC Main clock (CACMCLK)	MOSC	CAC	Up to 20 MHz
CAC Sub clock (CACSCLK)	SOSC	CAC	32.768 kHz
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
RTC clock (RTCSCLK/ RTCLCLK)	SOSC, LOCO	RTC	32.768 kHz
IWDT clock (IWDTCLK)	IWDTLOCO	IWDT	15 kHz
SysTick Timer clock (SYSTICCLK)	LOCO	SysTick Timer	32.768 kHz
JTAG clock (JTAGTCK)	TCK pin	JTAG	Up to 12.5 MHz
Clock/buzzer output (CLKOUT)	MOSC, SOSC, 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
Trace clock (TRCLK)	MOSC, SOSC, HOCO, MOCO, LOCO, PLL	CPU-OCD	Up to 48 MHz Division ratios: 1, 2, 4

Note: Restrictions on clock frequency settings: ICLK ≥ PCLKA ≥ PCLKB, PCLKD ≥ PCLKA ≥ PCLKB, ICLK ≥ FCLK, ICLK ≥ BCLK
Restrictions on clock frequency ratio: (N: integer, and up to 64)
ICLK:FCLK = N:1, ICLK:BCLK = N: 1, ICLK:PCLKA = N: 1, ICLK:PCLKB = N: 1
ICLK:PCLKC = N:1 or 1:N, ICLK:PCLKD = N:1 or 1:N
PCLKB:PCLKC = 1:1 or 1:2 or 1:4 or 2:1 or 4:1 or 8:1

Note: The minimum FCLK frequency is 1 MHz in Programming/Erasure mode.

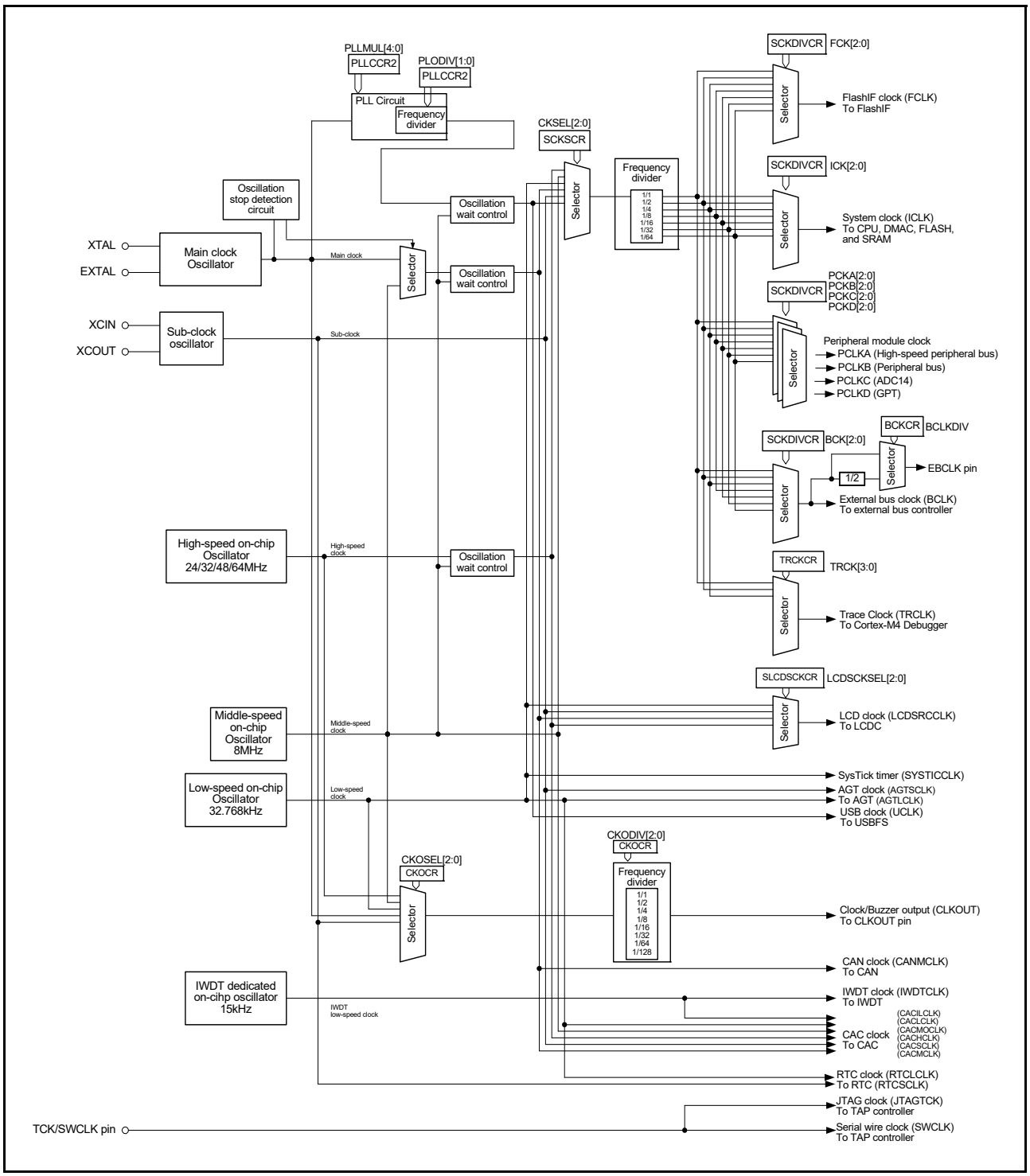


Figure 9.1 Clock generation circuit block diagram

Table 9.3 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, section 9.3.2, External Clock Input.
EXTAL	Input	
XCIN	Input	These pins are used to connect a 32.768-kHz crystal resonator
XCOUT	Output	
TCK/SWCLK	Input	This pin is used to input the clock for the JTAG
EBCLK	Output	This pin is used to supply external devices with the external bus clock (EBCLK)
CLKOUT	Output	This pin is used to output the CLKOUT/BUZZER clock

9.2 Register Descriptions

9.2.1 System Clock Division Control Register (SCKDIVCR)

Address(es): SYSTEM.SCKDIVCR 4001 E020h

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

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

Bit	Symbol	Bit name	Description	R/W
b2 to b0	PCKD[2:0]	Peripheral Module Clock D (PCLKD) Select*4	b2 b1 b0 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. Other settings are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	PCKC[2:0]	Peripheral Module Clock C (PCLKC) Select*4	b6 b5 b4 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. Other settings are prohibited.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	PCKB[2:0]	Peripheral Module Clock B (PCLKB) Select*3	b10 b9 b8 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. Other settings are prohibited.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b14 to b12	PCKA[2:0]	Peripheral Module Clock A (PCLKA) Select*3	b14 b12 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. Other settings are prohibited.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b18 to b16	BCK[2:0]	External Bus Clock (BCLK) Select*2	b18 b16 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. Other settings are prohibited.	R/W
b23 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b24	ICK[2:0]	System Clock (ICLK) Select*1, *2, *3, *4, *5	b26 b24 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. Other settings are prohibited.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30 to b28	FCK[2:0]	FlashIF Clock (FCLK) Select *1	b30 b28 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. Other settings are prohibited.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The following association is required between the frequencies of the system clock (ICLK) and the flash interface clock (FCLK):
ICLK:FCLK = N:1 (N: integer)
If a setting is written where ICLK < FCLK, the write is ignored.

Note 2. The following association is required between the frequencies of the system clock (ICLK) and the external bus clock (BCLK):
ICLK:BCLK = N:1 (N: integer)
If a setting is written where ICLK < BCLK, the write is ignored.

Note 3. The following association is required between the frequencies of the system clock (ICLK) and the peripheral module clocks (PCLKA, PCLKB): ICLK:PCLKA = N:1, ICLK:PCLKB = N:1 (N: integer)
If a setting is written where ICLK < PCLKA or ICLK < PCLKB, the write is ignored.

Note 4. The following association is required between the frequencies of the system clock (ICLK) and the peripheral module clocks (PCLKC, PCLKD): ICLK:PCLKC, PCLKD = N:1 or 1:N (N: integer).

Note 5. 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), the peripheral module clock (PCLKA, PCLKB, PCLKC, PCLKD), the flash interface clock (FCLK), and external bus clock (BCLK).

PCKD[2:0] bits (Peripheral Module Clock D (PCLKD) Select)

The PCKD[2:0] bits select the frequency for peripheral module clock D (PCLKD).

PCKC[2:0] bits (Peripheral Module Clock C (PCLKC) Select)

The PCKC[2:0] bits select the frequency for peripheral module clock C (PCLKC).

PCKB[2:0] bits (Peripheral Module Clock B (PCLKB) Select)

The PCKB[2:0] bits select the frequency for peripheral module clock B (PCLKB).

PCKA[2:0] bits (Peripheral Module Clock A (PCLKA) Select)

The PCKA[2:0] bits select the frequency for peripheral module clock A (PCLKA).

BCK[2:0] bits (External Bus Clock (BCLK) Select)

The BCK[2:0] bits select the frequency for the external bus clock (BCLK).

ICK[2:0] bits (System Clock (ICLK) Select)

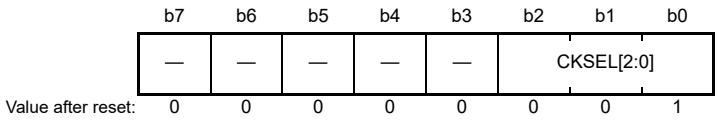
The ICK[2:0] bits select the frequency for the system clock for the CPU, DMAC, and DTC.

FCK[2:0] bits (FlashIF Clock (FCLK) Select)

The FCK[2:0] bits select the frequency for the flash interface clock (FCLK).

9.2.2 System Clock Source Control Register (SCKSCR)

Address(es): SYSTEM.SCKSCR 4001 E026h



Bit	Symbol	Bit name	Description	R/W
b2 to b0	CKSEL[2:0]	Clock Source Select*1	b2 b0 0 0 0: HOCO 0 0 1: MOCO 0 1 0: LOCO 0 1 1: Main clock oscillator (MOSC) 1 0 0: Sub-clock oscillator (SOSC) 1 0 1: PLL. Other settings are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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.

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 clock source for the following modules:

- System clock (ICLK)
- Peripheral module clocks (PCLKA, PCLKB, PCLKC, and PCLKD)
- Flash interface clock (FCLK)
- External bus clock (BCLK)

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)
- Sub-clock oscillator (SOSC)
- PLL circuit.

Transitions to clock sources that are not in operation are prohibited.

9.2.3 PLL Clock Control Register 2 (PLLCCR2)

Address(es): SYSTEM.PLLCCR2 4001 E02Bh

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

Bit	Symbol	Bit name	Description	R/W
b4 to b0	PLLMUL[4:0]	PLL Frequency Multiplication Factor Select*1	b4 b0 0 0 1 1 1: × 8 0 1 0 0 0: × 9 0 1 0 0 1: × 10 … 1 1 1 0 1: × 30 1 1 1 1 0: × 31 Other settings are prohibited.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	PLODIV[1:0]	PLL Output Frequency Division Ratio Select*1	b7 b6 0 0: Reserved 0 1: /2 1 0: /4 Other settings are prohibited.	R/W

Note 1. PLLMUL[4:0] and PLODIV[1:0] must be set so that the frequency of the PLL output signal is within the range specified in Table 9.1.

The PLLCCR2 register sets the operation of the PLL circuit. Writing to the PLLCCR2 is prohibited when the PLLCR.PLLSTP bit is 0, that is, when the PLL is operating.

PLLMUL[4:0] bits (PLL Frequency Multiplication Factor Select)

The PLLMUL[4:0] bits select the frequency multiplication factor of the PLL circuit.

PLODIV[1:0] bits (PLL Output Frequency Division Ratio Select)

The PLODIV[1:0] bits select the frequency division ratio of the PLL output.

9.2.4 PLL Control Register (PLLCR)

Address(es): SYSTEM.PLLCR 4001 E02Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PLLSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b0	PLLSTP	PLL Stop Control	0: Operate the PLL*1 1: Stop the PLL.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When operating the PLL, VCC must be more than 2.4V (VCC ≥ 2.4V), and operation power control mode must be set to high-speed mode or middle-speed mode.

The PLLCR register controls the operation of the PLL circuit.

PLLSTP bit (PLL Stop Control)

The PLLSTP bit starts or stops the PLL circuit.

After setting the PLLSTP bit to 0, you must confirm that the OSCSF.PLLSF bit is set to 1 before using the PLL clock. A fixed stabilization wait is required after setting the PLL to start operation. A fixed wait for the oscillations to stop is also required after stopping the PLL operation.

The following constraints apply when starting and stopping PLL operation:

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

Writing 1 to PLLSTP is prohibited during the following condition:

- SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL).

Make sure the following conditions apply before writing 0 to PLLSTP:

- OSCSF.MOSCSF bit is 1
- At least 4 µs elapses when PLLSTP is 1 (PLL is stopped)
- At least 1 µs elapses after PLLMUL[4:0] bits are set (to select the PLL frequency multiplication).

9.2.5 External Bus Clock Control Register (BCKCR)

Address(es): SYSTEM.BCKCR 4001 E030h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	BCLKDIV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	BCLKDIV	EBCLK Pin Output Select	0: BCLK 1: BCLK/2.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BCKCR register controls the external bus clock pin.

BCLKDIV bit (EBCLK Pin Output Select)

The BCLKDIV bit selects the clock signal for output from the EBCLK pin. The signal can be selected from either the BCLK clock with the frequency selected in the BCK[2:0] bits in SCKDIVCR or the BCLK clock divided by 2.

9.2.6 Memory Wait Cycle Control Register (MEMWAIT)

Address(es): SYSTEM.MEMWAIT 4001 E031h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MEMWAIT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MEMWAIT	Memory Wait Cycle Select	0: No wait 1: Wait.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 0 to the MEMWAIT bit is prohibited when SCKDIVCR.ICK selects division by 1 and SCKSCR.CKSEL[2:0] bits select the system clock source that is faster than 32 MHz (ICLK > 32 MHz).

This register controls the wait cycle of flash read access.

MEMWAIT bit (Memory Wait Cycle Select)

This bit selects the wait cycle of flash read access. The wait cycle of 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 constraints apply when setting the ICLK and operation power control mode, and the MEMWAIT bit:

- When setting the ICLK 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).

Note: When switching the operating power control mode, the flash cache function should be disabled by setting the CACHEE.FCACHEEN bit to 0 before switching the mode. For details, see section 48, Flash Memory.

Table 9.4 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	×	✓	✓

✓: Setting is possible.
×: Setting is not possible.

Figure 9.2 shows an example flow when setting the ICLK faster than 32 MHz.

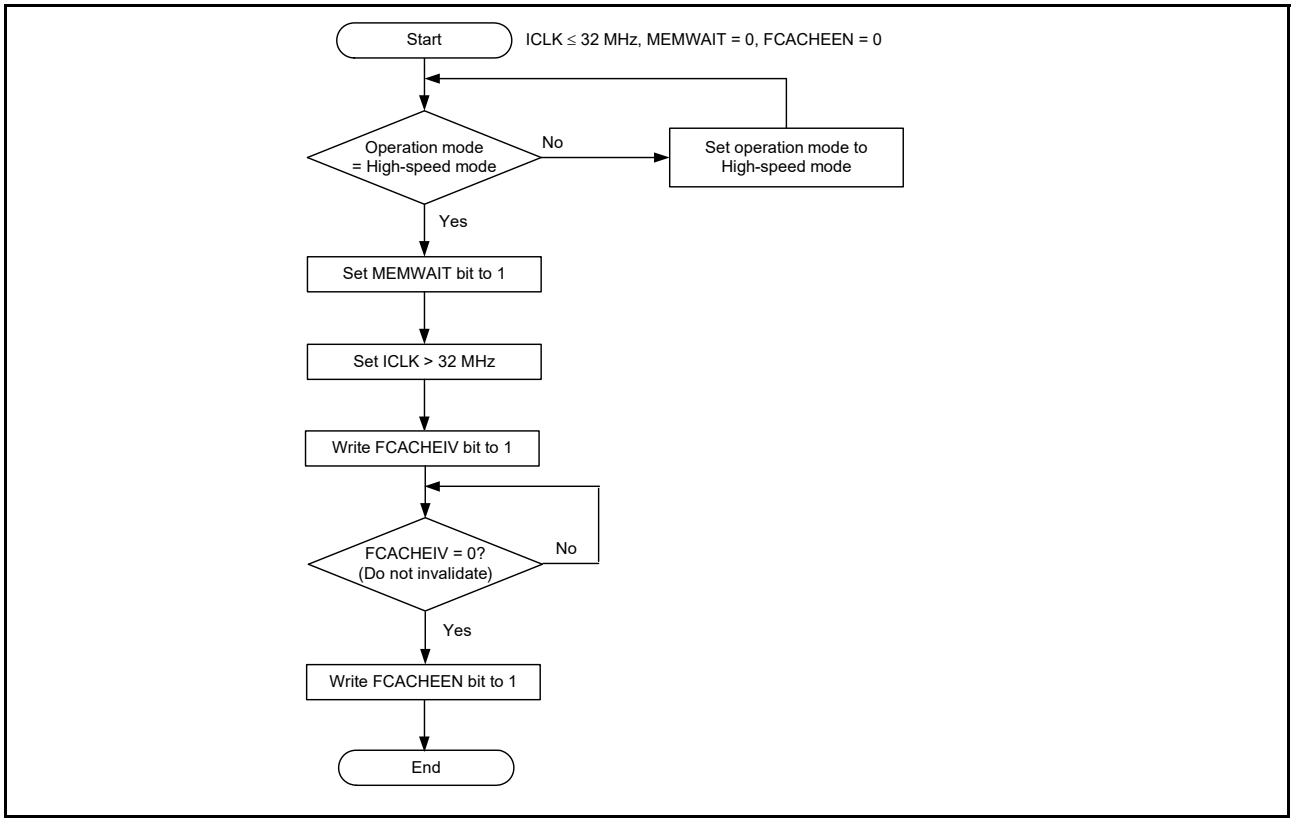


Figure 9.2 When setting the ICLK > 32 MHz

Figure 9.3 shows an example flow when setting the ICLK less than or equal to 32 MHz when ICLK is greater than 32 MHz.

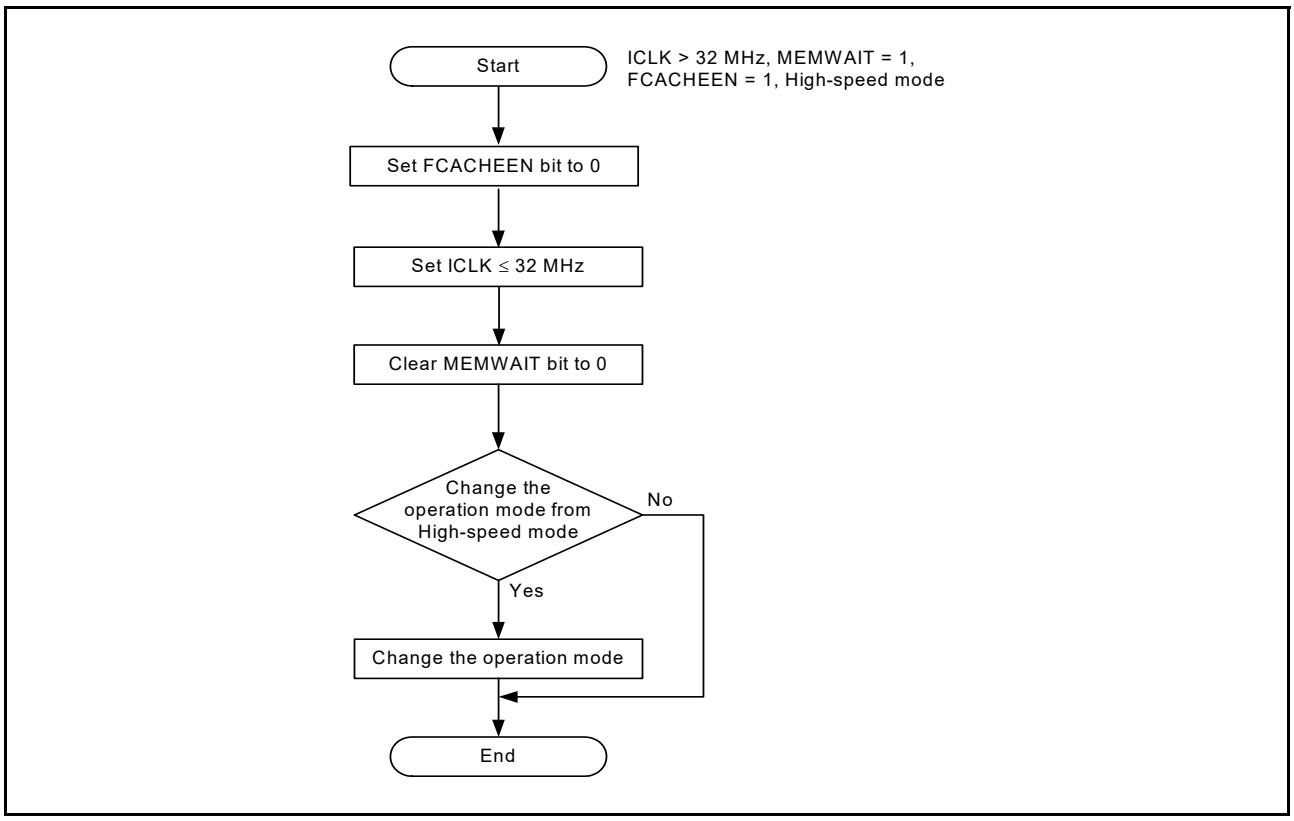


Figure 9.3 When setting the ICLK ≤ 32 MHz from ICLK > 32 MHz

9.2.7 Main Clock Oscillator Control Register (MOSCCR)

Address(es): SYSTEM.MOSCCR 4001 E032h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MOSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b0	MOSTP	Main Clock Oscillator Stop	0: Operate the main clock oscillator*1 1: Stop the main clock oscillator.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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.

The main clock oscillator can be started by setting the MOSTP bit to operate. When changing the value of the MOSTP bit, execute subsequent instructions only after reading the bit and checking that its 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. When the MOSCCR.MOSTP bit setting is modified for the main clock to run, use the main clock only after confirming that the OSCSF.MOSCSF bit is set to 1.

A fixed time is required for oscillations to become stable after setting the main clock oscillator. A fixed time is also required for oscillations to stop after stopping the main clock oscillator.

The following constraints 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 chip in Software Standby 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 conditions:

- SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC)
- SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL)
- PLLCR.PLLSTP = 0 (PLL operates).

9.2.8 Sub-Clock Oscillator Control Register (SOSCCR)

Address(es): SYSTEM.SOSCCR 4001 E480h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SOSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b0	SOSTP	Sub-Clock Oscillator Stop	0: Operate the sub-clock oscillator*1, *2 1: Stop the sub-clock oscillator.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The SOMCR register must be set before setting SOSTP to 0.
Note 2. The VBTCR1.BPWSWSTP bit must be set before setting the SOSC to operate when the VBATT function is not used. For VBTCR1.BPWSWSTP, see section 12, Battery Backup Function.

The SOSCCR register controls the sub-clock oscillator.

SOSTP bit (Sub-Clock Oscillator Stop)

The SOSTP bit starts or stops the sub-clock oscillator.

When changing the value of the SOSTP bit, execute subsequent instructions only after reading the bit and checking that its value is updated. Use the SOSTP bit when the sub-clock is used as the source for a peripheral module, for example, the RTC. When using the sub-clock, you must set the Sub Clock Oscillator Mode Control Register (SOMCR) before setting SOSTP to 0. After setting SOSTP to 0, use the sub-clock only after the sub-clock oscillation stabilization time (tSUBOSCOWT) elapses. A fixed stabilization wait time is required after selecting the sub-clock operation with the SOSTP bit. A fixed time is also required for oscillation to stop.

The following constraints apply when starting and stopping operation:

- When restarting the sub-clock oscillator after it stops, allow a period of at least 5 SOSC clock cycles for it to remain stopped.
- Confirm that sub-clock oscillator is stable when stopping the sub-clock oscillator.
- Regardless of whether the sub-clock oscillator is selected as the system clock, confirm that sub-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 a setting to stop the sub-clock oscillator, wait for at least 3 SOSC clock cycles before executing the WFI instruction.

Writing 1 to SOSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 100b (system clock source = SOSC).

9.2.9 Low-Speed On-Chip Oscillator Control Register (LOCOCR)

Address(es): SYSTEM.LOCOCR 4001 E490h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LCSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	LCSTP	LOCO Stop	0: Operate the LOCO clock*1 1: Stop the LOCO clock.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The VBTCR1.BPWSWSTP bit must be set before setting the LOCO to operate, when VBATT function is not used. For VBTCR1.BPWSWSTP, see section 12, Battery Backup Function.

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 start the LOCO clock, only use the clock after the LOCO clock oscillation stabilization waiting time (tLOCOWT) elapses. A fixed stabilization wait time is required after setting the LOCO to start operation. A fixed wait time is also required after setting the LOCO clock to stop.

The following constraints apply when starting and stopping operation:

- When restarting the LOCO clock after it stops, allow a stop interval of at least 5 LOCO cycles for it to remain stopped
- Confirm that LOCO oscillation is stable when stopping the LOCO clock
- Regardless of whether the LOCO clock is selected as the system clock, confirm that LOCO oscillation is stable before executing a WFI instruction to place the chip 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 LOSTP is prohibited under the following condition:

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

9.2.10 High-Speed On-Chip Oscillator Control Register (HOCOCR)

Address(es): SYSTEM.HOCOCR 4001 E036h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HCSTP
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Bit name	Description	R/W
b0	HCSTP	HOCO Stop	0: Operate the HOCO clock*2, *4 1: Stop the HOCO clock.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Writing to OPCCR.OPCM[1:0] is prohibited when 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. If the operating frequency of HOCO is 48 MHz, VCC must be more than 1.8 V (VCC ≥ 1.8 V) when operating the HOCO.
If the operating frequency of HOCO is 64 MHz, VCC must be more than 2.4 V (VCC ≥ 2.4 V) when operating the HOCO.

Note 3. Writing to HCSTP is prohibited while OPCCR.OPCMTSF = 1 or SOPCCR.SOPCMTSF = 1 (during transition of operating power control mode) or FLSTOP.CFLSTOPF = 1 (during transition of flash).

Note 4. When using the HOCO (HCSTP = 0), you must set the OFS1.HOCOFRQ1 bit to an optimum value. During low-voltage mode, HOCOCR.HCSTP must always be 0.

The HOCOCR register controls the HOCO clock.

HCSTP bit (HOCO Stop)

The HCSTP bit starts or stops the HOCO clock.

For the HOCO to operate, the High-Speed On-Chip Oscillator Wait Control Register (HOCOWTCR) must also be set. After setting the HCSTP bit to start the HOCO clock, confirm that the OSCSF.HOCOSF bit is set to 1 before using the clock. When OFS1.HOCOEN is set to 1, confirm that the OSCSF.HOCOSF bit 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 time is also required for oscillation to stop after setting the HOCO clock to stop.

The following constraints apply when starting and stopping operation:

- After stopping the HOCO, confirm that the OSCSF.HOCOSF bit is 0 before restarting the HOCO clock
- Confirm that the HOCO operates and that the OSCSF.HOCOSF bit is 1 before stopping the HOCO clock
- Regardless of whether the HOCO is selected as the system clock, confirm that the OSCSF.HOCOSF bit is set to 1 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 of the HOCO to stop, confirm that the OSCSF.HOCOSF bit is set to 0 before executing the WFI instruction.

Writing 1 to HCSTP is prohibited under the following condition:

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

9.2.11 Middle-Speed On-Chip Oscillator Control Register (MOCOCR)

Address(es): SYSTEM.MOCOCR 4001 E038h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MCSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MCSTP	MOCO Stop	0: Operate the MOCO clock 1: Stop the MOCO clock.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The MOCOCR register controls the MOCO.

MCSTP bit (MOCO Stop)

The MCSTP bit starts or stops the MOCO clock.

After setting MCSTP to 0 to start the MOCO clock, use the MOCO clock only after the MOCO clock oscillation stabilization time (tMOCOWT) 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 constraints 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 oscillation is stable when stopping the MOCO clock
- Regardless of whether the MOCO clock is selected as the system clock, confirm that MOCO 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 Enable bit in the Oscillation Stop Detection Control Register (OSTDCR.OSTDE).

Because the MOCO clock measures the waiting time for other oscillators, it continues to oscillate while measuring this time, regardless of the setting in the MOCOCR.MCSTP bit. Because of this, the MOCO clock may be unintentionally supplied even when the MCSTP is set to stop.

9.2.12 Oscillation Stabilization Flag Register (OSCSF)

Address(es): SYSTEM.OSCSF 4001 E03Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	PLLSF	—	MOSCSF	—	—	HOCOSF
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Bit name	Description	R/W
b0	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

Bit	Symbol	Bit name	Description	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	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
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	PLLSF	PLL Clock Oscillation Stabilization Flag	0: The PLL clock is stopped or is not yet stable 1: The PLL clock is stable, so the clock is available for use as the system clock.	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the OFS1.HOCOEN bits. When setting OFS1.HOCOEN = 1, the value of OSCSF.HOCOSF becomes 0 after reset is released, and OSCSF.HOCOSF value becomes 1 after HOCO oscillation stabilization wait time elapses.

Note 2. An appropriate value is set in the Wait Control register for the given oscillator. If the wait time 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 flags indicate the operating status of the counters in the oscillation stabilization wait circuits for the individual oscillators. After oscillation starts, the counters measure the wait time until each oscillator output clock is supplied to the internal circuits. An overflow of a counter indicates the start of the clock supply from the corresponding oscillator to the internal circuits.

HOCOSF flag (HOCO Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the wait time for the High-speed Clock Oscillator (HOCO).

When OFS1.HOCOEN is set to 1, confirm that the OSCSF.HOCOSF is also set to 1 before using the HOCO clock.

[Setting condition]

- After the HOCO clock stops and the HOCOCR.HCSTP bit is set to 0, the high-speed clock supply in the MCU starts after the middle-speed clock cycles set in the HOCOWTCR.HSTS[2:0] bits elapse.

[Clearing condition]

- When the high-speed clock oscillator is operating, it is deactivated when the HOCOCR.HCSTP bit is set to 1.

MOSCSF flag (Main Clock Oscillation Stabilization Flag)

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

[Setting condition]

- After the main clock oscillator stops and the MOSCCR.MOSTP bit is set to 0, supply of the main clock in the MCU starts after the number of middle-speed clock cycles associated with the setting in the MOSCWTCR.MSTS[3:0] bits are counted.

[Clearing condition]

- When the main clock oscillator is operating, it is deactivated when the MOSCCR.MOSTP bit is set to 1.

PLLSF flag (PLL Clock Oscillation Stabilization Flag)

The PLLSF flag indicates the state of operation of the counter that measures the wait time of the PLL.

[Setting condition]

- After the PLL stops and the PLLCR.PLLSTP bit is set to 0, supply of the PLL clock in the MCU after 370 cycles of the middle-speed clock are counted. If oscillation by the PLL clock source is not stable when the PLLSTP bit is set to 0, counting of the middle-speed clock cycles continues after the oscillation of the PLL clock source is stabilized.

[Clearing condition]

- When the PLL is operating, it is deactivated when the PLLCR.PLLSTP bit is set to 1.

9.2.13 Oscillation Stop Detection Control Register (OSTDCR)

Address(es): SYSTEM.OSTDCR 4001 E040h

	b7	b6	b5	b4	b3	b2	b1	b0
	OSTDE	—	—	—	—	—	—	OSTDIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	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
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	OSTDE	Oscillation Stop Detection Function Enable	0: Disable oscillation stop detection function 1: Enable oscillation stop detection function.	R/W

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 notified to the POEG.

If the Oscillation Stop Detection Flag in the Oscillation Stop Detection Status Register (OSTDSR.OSTDF) requires clearing, clear the OSTDF after the OSTDIE bit is set to 0. Wait for at least 2 PCLKB cycles before setting this bit to 1. A longer wait time might be required depending on the number of cycles required to read out a given I/O register.

OSTDE bit (Oscillation Stop Detection Function Enable)

The OSTDE bit enables the oscillation stop detection function.

When this bit is 1 (enable), 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.

This 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 and 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, 4, 8 for ICLK, FCLK, BCLK, PCLKA, PCLKB, PCLKC, PCLKD is prohibited
- In low-voltage mode, selecting division by 1, 2 for ICLK, FCLK, BCLK, PCLKA, PCLKB, PCLKC, PCLKD is prohibited.

9.2.14 Oscillation Stop Detection Status Register (OSTDSR)

Address(es): SYSTEM.OSTDSR 4001 E041h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	OSTDF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	OSTDF	Oscillation Stop Detection Flag	0: Main clock oscillation stop has not been detected 1: Main clock oscillation stop has been detected.	R(/W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. This bit can only be set to 0.

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 the main clock oscillation stop is detected, this flag is not set to 0 even when the main clock oscillation is restarted. The OSTDF flag is set 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 OSTDF 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.

OSTDSR.OSTDF 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 and PLL.

[Setting condition]

- The main clock oscillation is stopped when OSTDCR.OSTDE is 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 (system clock is PLL).

9.2.15 Main Clock Oscillator Wait Control Register (MOSCWTCR)

Address(es): SYSTEM.MOSCWTCR 4001 E0A2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	MSTS[3:0]			
Value after reset:	0	0	0	0	0	1	0	1

Bit	Symbol	Bit name	Description	R/W
b3 to b0	MSTS[3:0]	Main Clock Oscillator Wait Time Setting	b3 b0 0 0 0 0: Wait time = 2 cycles (0.25 µs) 0 0 0 1: Wait time = 1024 cycles (128 µs) 0 0 1 0: Wait time = 2048 cycles (256 µs) 0 0 1 1: Wait time = 4096 cycles (512 µs) 0 1 0 0: Wait time = 8192 cycles (1024 µs) 0 1 0 1: Wait time = 16384 cycles (2048 µs) (value after reset) 0 1 1 0: Wait time = 32768 cycles (4096 µs) 0 1 1 1: Wait time = 65536 cycles (8192 µs) 1 0 0 0: Wait time = 131072 cycles (16384 µs) 1 0 0 1: Wait time = 262144 cycles (32768 µs). Other settings are prohibited. Wait time is calculated at MOCO = 8 MHz (typically 0.125 µs).	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

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

Set the MSTS[3:0] bits to select the oscillation stabilization wait time of 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 0000b because the oscillation stabilization time is not required.

The wait time set in the MSTS[3:0] bits is counted using the MOCO clock. The MOCO automatically oscillates when necessary, regardless of the value of the MOCOCR.MCSTP bit.

After the set wait time elapses, supply of the main clock is started internally in the MCU, and the OSCSF.MOSCSF flag becomes 1. If the specified wait time is short, supply of the main clock is started 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.

9.2.16 High-Speed On-Chip Oscillator Wait Control Register (HOCOWTCR)

Address(es): SYSTEM.HOCOWTCR 4001 E0A5h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	HSTS[2:0]		
Value after reset:	0	0	0	0	0	1	0	1

Bit	Symbol	Bit name	Description	R/W
b2 to b0	HSTS[2:0]	HOCO wait time setting	b2 b0 1 0 1: • Wait time = 245 cycles (29.125 µs) when HOCO operating frequency is 24 MHz or 32 MHz and operation power control mode is other than low-voltage mode. • Wait time = 287 cycles (35.875 µs) when HOCO operating frequency is 48 MHz and operation power control mode is other than low-voltage mode. • Wait time = 679 cycles (84.88 µs) (value after reset) and operation power control mode is low-voltage mode. 1 1 0: • Wait time = 541 cycles (67.63 µs) when HOCO operating frequency is 64 MHz. Other settings are prohibited. Wait time calculated at MOCO is 8 MHz (typically 0.125 µs).	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

HOCOWTCR controls the wait time until output of the signal from the high speed clock oscillator to the internal circuits starts. HOCOWTCR can be written to only when the HOCOCR.HCSTP bit is 1 or the OSCSF.HOCOSF flag is 1. Do not write to HOCOWTCR 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 clock oscillator starts, the oscillation stabilization wait circuit starts counting cycles of the middle-speed clock associated with the setting 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 to 1.

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

Note: When the HOCO clock operating frequency is 64 MHz, HOCOWTCR.HSTS[2:0] must always be set to 110b.

9.2.17 Main Clock Oscillator Mode Oscillation Control Register (MOMCR)

Address(es): SYSTEM.MOMCR 4001 E413h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MOSEL	—	—	MODRV1	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	MODRV1	Main Clock Oscillator Drive Capability 1 Switching	0: 10 MHz to 20 MHz 1: 1 MHz to 10 MHz.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	MOSEL	Main Clock Oscillator Switching	0: Resonator 1: External clock input.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The EXTAL/XTAL pin is also used as a port. In the initial setting state, the pin is set as a port.

The MOSTP bit must be 1 (MOSC is stopped) before modifying 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.

9.2.18 Sub-Clock Oscillator Mode Control Register (SOMCR)

Address(es): SYSTEM.SOMCR 4001 E481h

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

Bit	Symbol	Bit name	Description	R/W
b1, b0	SODRV[1:0]	Sub-Clock Oscillator Drive Capability Switching	b1 b0 0 0: Normal mode 0 1: Low power mode 1 1 0: Low power mode 2 1 1: Low power mode 3.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SOSCCR.SOSTP must be 1 (SOSC is stopped) before modifying this register.

SODRV[1:0] bits (Sub-Clock Oscillator Drive Capability Switching)

The SODRV[1:0] bits switch the drive capability of the sub-clock oscillator.

9.2.19 Segment LCD Source Clock Control Register (SLCDSCKCR)

Address(es): SYSTEM.SLCDSCKCR 4001 E050h

	b7	b6	b5	b4	b3	b2	b1	b0
	LCDSCKEN	—	—	—	—	LCDSCKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	LCDSCKSEL[2:0]	LCD Source Clock (LCDSRCCLK) Select	b2 b0 0 0 0: LOCO 0 0 1: SOSC 0 1 0: MOSC 1 0 0: HOCO. Other settings are prohibited.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	LCDSCKEN	LCD Source Clock Out Enable	0: Disable LCD source clock out 1: Enable LCD source clock out.	R/W

Setting the LCDSCKEN bit and LCDSCKSEL[2:0] bits at the same time is prohibited.

LCDSCKSEL[2:0] bits (LCD Source Clock (LCDSRCCLK) Select)

The LCDSCKSEL[2:0] bits select the LOCO, SOSC, MOSC, or HOCO clock as the LCD clock source. Clear the LCDSCKEN bit to 0 when changing the LCD source clock.

When changing these bits, use the following steps:

1. Set LCDSCKEN to 0 (LCD source clock out is disabled).
2. Wait for 3 LCD source clock cycles and 2 ICLK cycles before making the change.
3. Write the changed value to LCDSCKSEL[2:0] bits.
4. Read LCDSCKSEL[2:0] bits to confirm that the bits changed.

LCDSCKEN bit (LCD Source Clock Out Enable)

Set the LCDSCKEN bit to enable output of the LCD source clock to LCD module.

When this bit is set to 1, the selected clock is output. Confirm that the LCD source clock selected by LCDSCLKSEL[2:0] bits is stable before changing this bit. When transitioning to Software Standby mode after changing this bit, use the following steps:

1. Change this bit.
2. Wait for at least 2 cycles of the source clock selected in LCDSCKSEL[2:0] bits.
3. Execute the WFI instruction.

When stopping the source clock selected by LCDSCKSEL[2:0] bits after clearing this bit to 0, use the following steps:

1. Clear this bit to 0 (LCD source clock output is disabled).
2. Wait for at least 2 cycles of the source clock selected by LCDSCKSEL[2:0] bits.
3. Stop the source clock selected by LCDSCKSEL[2:0] bits.

9.2.20 Clock Out Control Register (CKOCR)

Address(es): SYSTEM.CKOCR 4001 E03Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	CKOEN	CKODIV[2:0]			—	CKOSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CKOSEL[2:0]	Clock Out Source Select	b2 b0 0 0: HOCO 0 1: MOCO 1 0: LOCO 1 1: MOSC 1 0: SOSC. Other settings are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	CKODIV[2:0]	Clock Out input frequency Division Select	b6 b4 0 0: ×1 0 1: /2 1 0: /4 1 1: /8 0 0: /16 0 1: /32 1 0: /64 1 1: /128.	R/W
b7	CKOEN	Clock Out enable	0: Disable clock out 1: Enable clock out.	R/W

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

The CKOSEL[2:0] bits select the HOCO, MOCO, LOCO, MOSC, or SOSC clock as the source of the clock to be output from the CLKOUT pin.

When changing the CLKOUT source clock, clear the CKOEN bit to 0.

CKODIV[2:0] bits (Clock Out input frequency Division Select)

The CKODIV[2:0] bits select the clock division ratio.

When changing the division ratio, clear the CKOEN bit to 0. The division ratio of the output clock frequency should be set to a value no higher than the characteristics of the CLKOUT pin output frequency. For details on the CLKOUT pin characteristics, see section 52, 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 selected clock out source clock is stopped in that mode.

9.2.21 External Bus Clock Output Control Register (EBCKOCR)

Address(es): SYSTEM.EBCKOCR 4001 E052h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	EBCKOEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	EBCKOEN	EBCLK Pin Output Control	0: Disable EBCLK pin output (fixed high) 1: Enable EBCLK pin output.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

9.2.22 LOCO User Trimming Control Register (LOCOUTCR)

Address(es): SYSTEM.LOCOUTCR 4001 E492h

	b7	b6	b5	b4	b3	b2	b1	b0
	LOCOUTRM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

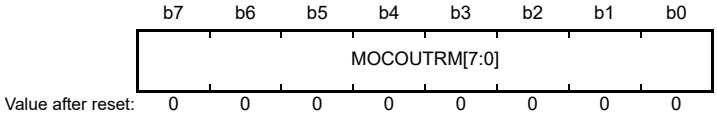
Bit	Symbol	Bit name	Description	R/W
b7 to b0	LOCOUTRM[7:0]	LOCO User Trimming	b7 b0 1 0 0 0 0 0 0 0: -128 1 0 0 0 0 0 0 1: -127 1 0 0 0 0 0 1 0: -126 … 1 1 1 1 1 1 1 1: -1 0 0 0 0 0 0 0 0: Center Code 0 0 0 0 0 0 0 1: +1 … 0 1 1 1 1 1 0 1: +125 0 1 1 1 1 1 1 0: +126 0 1 1 1 1 1 1 1: +127 These bits are added to the original LOCO trimming bits.	R/W

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 required corresponds to the frequency stabilization time at the start of the MCU operation. When the ratio of the LOCO clock frequency and the other oscillation frequency is an integer value, changing the LOCOUTCR value is prohibited.

9.2.23 MOCO User Trimming Control Register (MOCOUTCR)

Address(es): SYSTEM.MOCOUTCR 4001 E061h



Bit	Symbol	Bit name	Description	R/W
b7 to b0	MOCOUTRM[7:0]	MOCO User Trimming	b7 b0 1 0 0 0 0 0 0 0: -128 1 0 0 0 0 0 0 1: -127 1 0 0 0 0 0 1 0: -126 … 1 1 1 1 1 1 1 1: -1 0 0 0 0 0 0 0 0: Center Code 0 0 0 0 0 0 0 1: +1 … 0 1 1 1 1 1 0 1: +125 0 1 1 1 1 1 1 0: +126 0 1 1 1 1 1 1 1: +127 These bits are added to the original MOCO trimming bits.	R/W

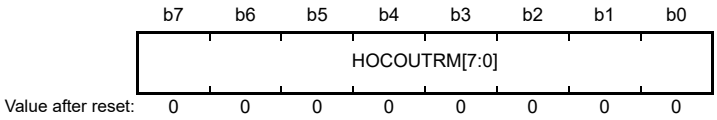
MCU operation is not guaranteed when MOCOUTCR is set to a value that causes the MOCO frequency to be outside of the specification range.

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.

9.2.24 HOCO User Trimming Control Register (HOCOUTCR)

Address(es): SYSTEM.HOCOUTCR 4001 E062h



Bit	Symbol	Bit name	Description	R/W
b7 to b0	HOCOUTRM[7:0]	HOCO User Trimming	b7 b0 1 0 0 0 0 0 0 0: -128 1 0 0 0 0 0 0 1: -127 1 0 0 0 0 0 1 0: -126 … 1 1 1 1 1 1 1 1: -1 0 0 0 0 0 0 0 0: Center Code 0 0 0 0 0 0 0 1: +1 … 0 1 1 1 1 1 0 1: +125 0 1 1 1 1 1 1 0: +126 0 1 1 1 1 1 1 1: +127 These bits are added to the original HOCO trimming bits.	R/W

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 frequency stabilization wait time at the start of the MCU operation.

9.2.25 Trace Clock Control Register (TRCKCR)

Address(es): SYSTEM.TRCKCR 4001 E03Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	TRCKEN	—	—	—	TRCK[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit name	Description	R/W
b3 to b0	TRCK[3:0]	Trace Clock operation frequency select	b3 b0 0 0 0 0: /1 0 0 0 1: /2 (value after reset) 0 0 1 0: /4 Other settings are prohibited.	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TRCKEN	Trace Clock operating enable	0: Disable operation 1: Enable operation.	R/W

The Trace Clock Control Register controls the switching of the trace clock. Set TRCKEN to 0 before changing the TRCLK frequency.

Note: TRCKCR register can be initialized by all resets except VBATT_POR.

9.3 Main Clock Oscillator

To supply the clock signal to the main clock oscillator use one of the 2 following ways:

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

9.3.1 Connecting Crystal Resonator

Figure 9.4 shows an example of connection to 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 resonator manufacturer recommends the use of 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 9.1.

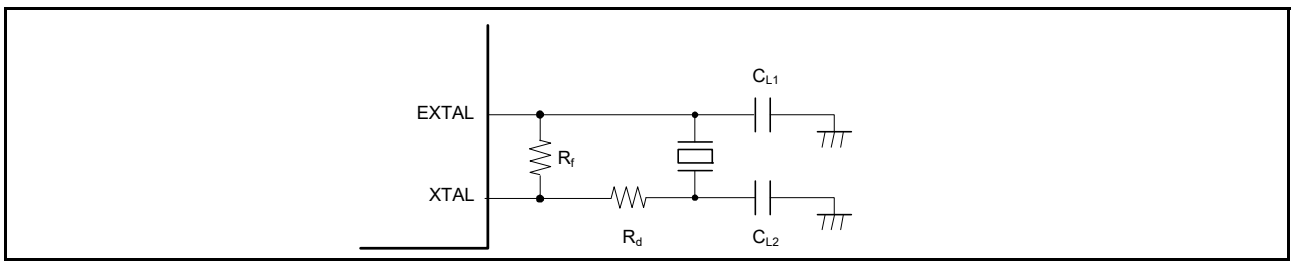


Figure 9.4 Example of crystal resonator connection

9.3.2 External Clock Input

Figure 9.5 shows an example connection of an external clock input. To operate the oscillator with an external clock signal, set the MOMCR.MOSEL bit to 1. The XTAL pin goes to high-impedance.

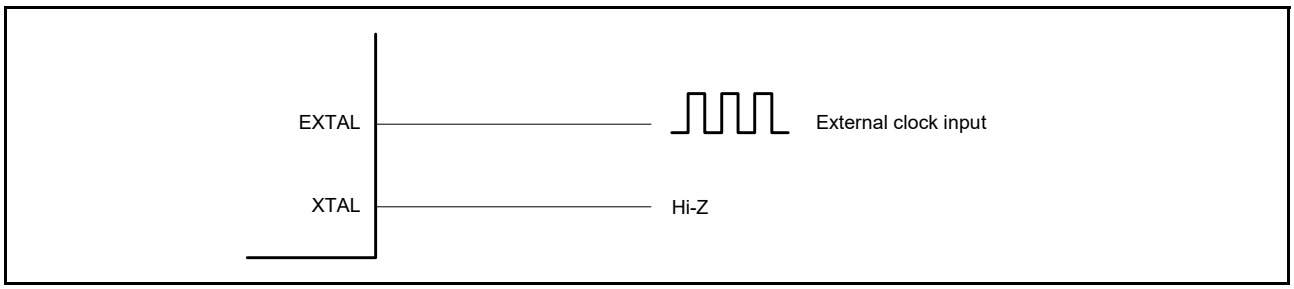


Figure 9.5 Equivalent circuit for external clock

9.3.3 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.

9.4 Sub-Clock Oscillator

The only way of supplying a clock signal to the sub-clock oscillator is by connecting a crystal oscillator.

9.4.1 Connecting a 32.768-kHz Crystal Resonator

To supply a clock to the sub-clock oscillator, connect a 32.768-kHz crystal resonator, as shown in Figure 9.6. A damping resistor (Rd) can be added, if necessary. Because the resistor values vary according to the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If the resonator manufacturer recommends the use of an external feedback resistor (Rf), insert an Rf between XCIN and XCOUT 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 sub-clock oscillator as described in Table 9.1.

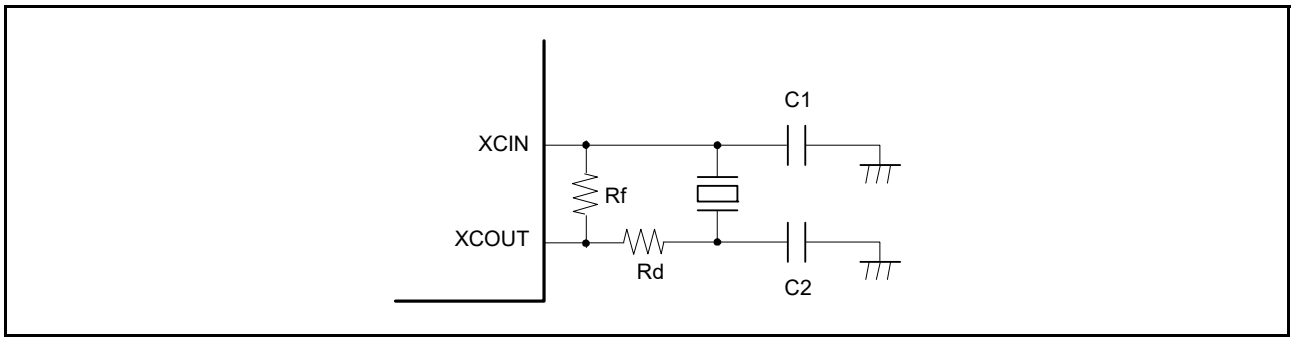


Figure 9.6 Connection example of 32.768-kHz crystal resonator

9.5 Oscillation Stop Detection Function

9.5.1 Oscillation Stop Detection and Operation after Detection

The oscillation stop detection function detects the main clock oscillator stop. When an 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
- If an oscillation stop is detected with SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL), the PLL clock remains as the system clock source. The frequency becomes a free-running oscillation frequency and the setting of the SCKSCR.CKSEL[2:0] bits does not change.

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 52, Electrical Characteristics.

Switching between the main clock oscillator and the MOCO clock or between the PLL clock and the PLL free-running clock is controlled by the oscillation stop detection flag (OSTDSR.OSTDF).

OSTDF controls the switched clock as follows:

- 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 to MOSC again.
- SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL):
  When OSTDF changes from 0 to 1, the clock source switches to the PLL free-running oscillation clock.
  When OSTDF changes from 1 to 0, the clock source switches to PLL again.

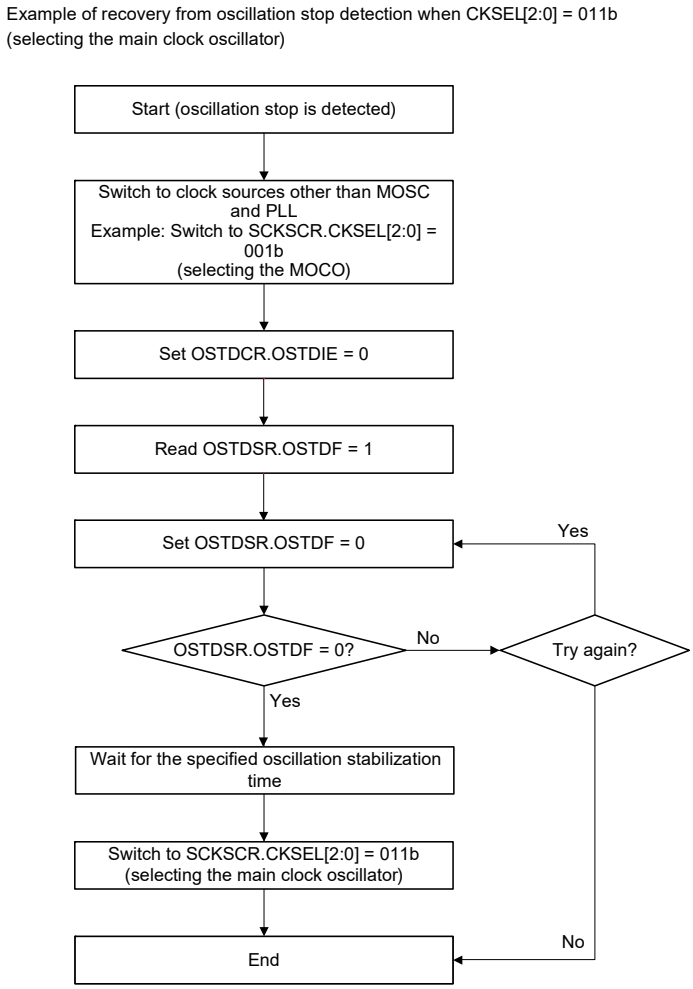
To switch the clock source to the main clock or PLL clock again after oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock or PLL clock, and clear the OSTDF flag to 0. Also, check that the OSTDF flag is not 1, then set the CKSEL[2:0] bits to the main clock or PLL 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. So, the oscillation stop detection function must be disabled before the main clock oscillator is stopped by the software or a transition is made to Software Standby mode.

The oscillation stop detection function switches the following clocks to the MOCO clock (when system clock is MOSC) or the PLL free-running clock (when system clock is PLL):

- All clocks that can be selected as the MOSC or PLL except CLKOUT
- The system clock (ICLK) frequency during the MOCO operation (when system clock is MOSC) or PLL free-running (when system clock is PLL) operation is specified in the MOCO oscillation frequency and the division ratio set by the system clock select bits (SCKDIVCR.ICK[2:0]).



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 9.7 Flow of recovery on detection of oscillator stop

9.5.2 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 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 the oscillation stop detection interrupts. For details, see section 14, Interrupt Controller Unit (ICU).

9.6 PLL Circuit

The PLL circuit provides a function for multiplying the frequency from the oscillator.

9.7 Internal Clock

Clock sources for the internal clock signals include:

- Main clock oscillator
- Sub-clock oscillator
- HOCO clock
- MOCO clock
- LOCO clock
- PLL clock
- Dedicated clock for the IWDT
- External clock for JTAG.

The following internal clocks are produced from these sources:

- Operating clock for the CPU, DMAC, DTC, Flash, and SRAM — system clock (ICLK)
- Operating clocks for peripheral modules — PCLKA, PCLKB, PCLKC, and PCLKD
- Operating clock for the flash interface— FCLK
- Clock for the external bus controller and external pin output — EBCLK
- Operating clock for the USBFS — UCLK
- Operating clock for the CAN — CANMCLK
- Operating clocks for the CAC — CACCLK
- Operating clock for the RTC LOCO clock — RTCLCLK
- Operating clock for the RTC sub clock — RTCSCLK
- Operating clock for the IWDT — IWDTCLK
- Operating clock for the AGT LOCO clock — AGTLCLK
- Operating clock for the AGT sub clock — AGTSCLK
- Operating clock for the SysTick timer — SYSTICCLK
- Source clock for the SLCDC — LCDSRCCLK
- Clock for external pin output — CLKOUT
- Operating clock for the JTAG — JTAGTCK.

For details of the registers used to set the frequencies of the internal clocks, see section 9.7.1, System Clock (ICLK) to section 9.7.14, JTAG Clock (JTAGTCK).

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

9.7.1 System Clock (ICLK)

The system clock, ICLK, is the operating clock for the CPU, DMAC, DTC, flash memory, and SRAM.

The ICLK frequency is specified in the following bits:

- ICK[2:0] in SCKDIVCR
- CKSEL[2:0] in SCKSCR
- PLLMUL[4:0] and PLODIV[1:0] in PLLCCR2
- HOCOFRQ1[2:0] in OFS1.

9.7.2 Peripheral Module Clock (PCLKA, PCLKB, PCLKC, PCLKD)

The peripheral module clocks, PCLKA, PCLKB, PCLKC, and PCLKD, are the operating clocks for the peripheral modules.

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

- PCKA[2:0], PCKB[2:0], PCKC[2:0], and PCKD[2:0] in SCKDIVCR
- CKSEL[2:0] in SCKSCR
- PLLMUL[4:0] and PLODIV[1:0] in PLLCCR2
- HOCOFRQ1[2:0] in OFS1.

9.7.3 Flash Interface Clock (FCLK)

The flash interface clock, FCLK, is the operating clock for the flash memory interface. In addition to reading from the data flash, FCLK is used for the programming and erasure of the code flash and data flash.

The FCLK frequency is specified in the following bits:

- FCK[2:0] in SCKDIVCR
- CKSEL[2:0] in SCKSCR
- PLLMUL[4:0] and PLODIV[1:0] in PLLCCR2
- HOCOFRQ1[2:0] in OFS1.

9.7.4 External Bus Clock (BCLK)

The external bus clock, BCLK, is an operating clock for the external bus controller. It is also output externally from the EBCLK pin for the external connection bus.

BCLK can be output from the EBCLK pin by setting the EBCKOCR.EBCKOEN bit to 1 and setting the PmnPFS.PSEL[4:0] bits to 01011b. Only change the PmnPFS.PSEL[4:0] bits to 01011b when the EBCKOCR.EBCKOEN bit is 0. When the BCKCR.BCLKDIV bit is set to 1, the BCLK clock divided by 2 is output from the EBCLK pin.

The BCLK frequency is specified in the following bits:

- BCK[2:0] in SCKDIVCR
- CKSEL[2:0] in SCKSCR
- PLLMUL[4:0], and PLODIV[1:0] in PLLCCR2
- HOCOFRQ1[2:0] in OFS1.

BCLK should not be set to a frequency higher than the system clock (ICLK) frequency.

9.7.5 USB Clock (UCLK)

The USB clock, UCLK, is an operating clock for the USBFS module. A 48-MHz clock must be supplied to the USBFS module. When the USBFS module is used, the UCLK clock must be specified at 48 MHz.

The UCLK frequency is specified in the following bits:

- CKSEL[2:0] in SCKSCR
- PLLMUL[4:0] and PLODIV[1:0] in PLLCCR2
- HOCOFRQ1[2:0] in OFS1.

9.7.6 CAN Clock (CANMCLK)

The CAN clock, CANMCLK, is an operating clock for the CAN module. CANMCLK is generated by the main clock oscillator.

9.7.7 CAC Clock (CACCLK)

The CAC clock, CACCLK, is an operating clock for the CAC.

CACCLK is generated by the following:

- Main clock oscillator
- Sub-clock oscillator
- High-speed clock oscillator
- Middle-speed clock oscillator
- Low-speed on-chip oscillator
- IWDT-dedicated on-chip oscillator.

9.7.8 RTC-Dedicated Clock (RTCSCLK, RTCLCLK)

The RTC-dedicated clocks, RTCSCLK and RTCLCLK, are the operating clock for the RTC.

RTCSCLK is generated by the sub-clock oscillator, and RTCLCLK is generated by the LOCO clock.

9.7.9 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.

9.7.10 AGT-Dedicated Clock (AGTSCLK, AGTLCLK)

The AGT-dedicated clock, AGTSCLK and AGTLCLK, is the operating clock for the AGT.

AGTSCLK is generated by the sub-clock oscillator, and AGTLCLK is generated by the LOCO clock.

9.7.11 SysTick Timer-Dedicated Clock (SYSTICCLK)

The SysTick timer-dedicated clock, SYSTICKCLK, is the operating clock for the SYSTICCLK.

SYSTICCLK is generated by the LOCO clock.

9.7.12 Segment LCDC Source Clock (LCDSRCCLK)

The Segment LCDC source clock, LCDSRCCLK, is used as the operating clock of the SLCDC.

The LCDSRCCLK is specified by the LCDSCKSEL[2:0] bits in SLCDSCKCR.

LCDSRCCLK is output when SLCDSCKCR.LCDSCKEN is set to 1. When changing the value of SLCDSCKCR.LCDSCKSEL[2:0], make sure that the value of SLCDSCKCR.LCDSCKEN is 0.

9.7.13 Clock/Buzzer Output Clock (CLKOUT)

The CLKOUT is output externally from the CLKOUT pin for the clock or buzzer output.

CLKOUT is output to the CLKOUT pin when CKOCR.CKOEN is set to 1. Only change the value of CKODIV[2:0] or CKOSEL[2:0] in CKOCR, when the CKOCR.CKOEN bit is 0.

The CLKOUT clock frequency is specified in the following bits:

- CKODIV[2:0] or CKOSEL[2:0] in CKOCR
- PLLMUL[4:0] and PLODIV[1:0] in PLLCCR2
- HOCOFRQ1[2:0] in OFS1.

9.7.14 JTAG Clock (JTAGTCK)

JTAGTCK, is the dedicated operating clock for the JTAG.

JTAGTCK is generated by the external clock for JTAG (TCK).

9.8 Usage Notes

9.8.1 Notes on Clock Generation Circuit

The frequencies of the system clock (ICLK), peripheral module clock (PCLKA to PCLKD), flash interface clock (FCLK), and external bus clock (BCLK) supplied to each module change according to the settings of SCKDIVCR. Each frequency must meet the following conditions:

- Select each frequency that is within the operation-guaranteed range of the clock cycle time (t_{cyc}) specified in the AC electrical characteristics. See section 52, Electrical Characteristics
- The frequencies must not exceed the ranges listed in Table 9.2.
- The peripheral modules operate on the PCLKB and PCLKA. The operating speed of modules such as the timer and SCI is different before and after the frequency is changed.
- The system clock (ICLK), peripheral module clock (PCLKA to PCLKD), flash interface clock (FCLK), external bus clock (BCLK) must be set according to Table 9.2.

Do not change the clock frequency during external bus access. In addition, when external bus access starts after a change to the clock frequency, confirm that the frequency changes are complete before accessing the bus. To ensure correct processing after the clock frequency changes, first modify the relevant Clock Control register to change the frequency, next read the value from the register, and finally perform the subsequent processing.

9.8.2 Notes on Resonator

Because various resonator characteristics relate closely to your board design, adequate evaluation is required before use. See the resonator connection example in Figure 9.6. The circuit constants for the resonator depend on the resonator to be used and the stray capacitance of the mounting circuit. Therefore, consult the resonator manufacturer when determining the circuit constants. The voltage to be applied between the resonator pins must be within the absolute maximum rating.

9.8.3 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 9.8,W to prevent electromagnetic induction from interfering with correct oscillation.

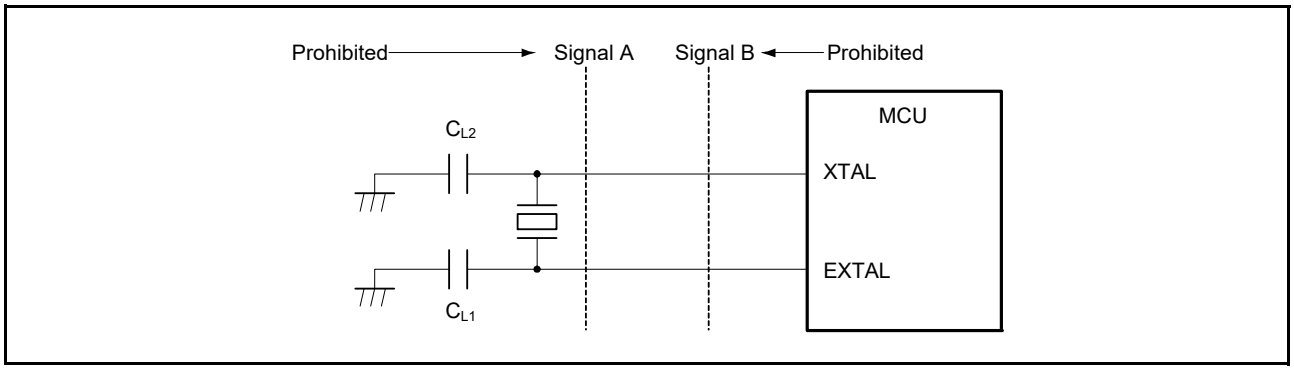


Figure 9.8 Signal routing in board design for oscillation circuit (applies to the sub-clock oscillator for the main clock oscillator)

9.8.4 Notes on Resonator Connect Pin

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports P212 and P213. When these pins are used as general ports, the main clock must be stopped (MOSCCR.MOSTP should be set to 1).

10. Clock Frequency Accuracy Measurement Circuit (CAC)

10.1 Overview

The Clock Frequency Accuracy Measurement Circuit (CAC) counts the number of pulses of the clock to be measured (measurement target clock) within the time generated by the clock to be used as a measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range.

When measurement is completed 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 10.1 lists the CAC specifications, Figure 10.1 shows the block diagram, and Table 10.2 describes the I/O pins.

Table 10.1 CAC specifications

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

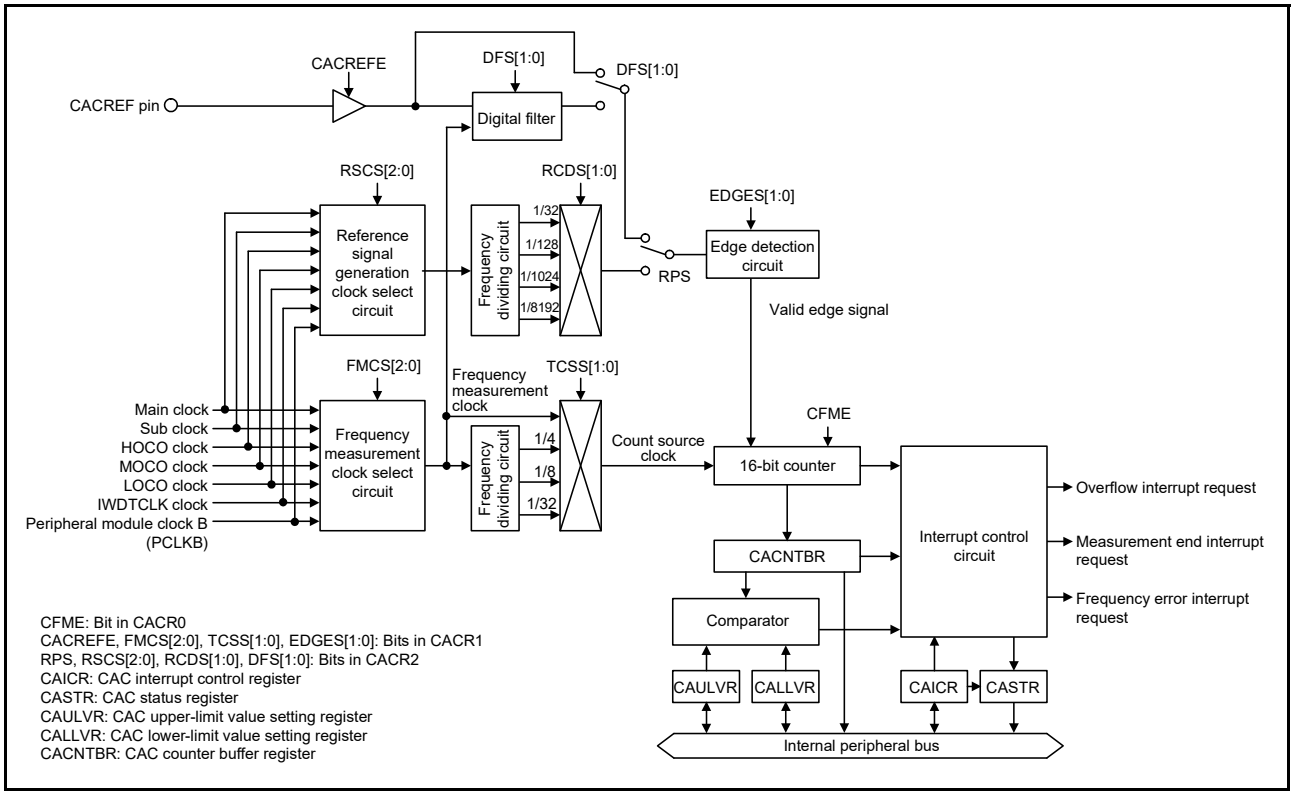


Figure 10.1 CAC block diagram

Table 10.2 CAC pin configuration

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

10.2 Register Descriptions

10.2.1 CAC Control Register 0 (CACR0)

Address(es): CAC.CACR0 4004 4600h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFME
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CFME	Clock Frequency Measurement Enable	0: Disable 1: Enable.	R/W
b7 to b1	—	Reserved	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. Read the CFME bit to confirm that the bit value has changed. Additional write accesses are ignored before the change is complete.

10.2.2 CAC Control Register 1 (CACR1)

Address(es): CAC.CACR1 4004 4601h

	b7	b6	b5	b4	b3	b2	b1	b0
	EDGES[1:0]		TCSS[1:0]		FMCS[2:0]			CACREFE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CACREFE	CACREF Pin Input Enable	0: Disable 1: Enable.	R/W
b3 to b1	FMCS[2:0]	Measurement Target Clock Select	b3 b1 0 0 0: Main clock oscillator 0 0 1: Sub-clock oscillator 0 1 0: HOCO clock 0 1 1: MOCO clock 1 0 0: LOCO clock 1 0 1: Peripheral module clock (PCLKB) 1 1 0: IWDTCLK clock 1 1 1: Setting prohibited.	R/W
b5, b4	TCSS[1:0]	Measurement Target Clock Frequency Division Ratio Select	b5 b4 0 0: No division 0 1: ×1/4 clock 1 0: ×1/8 clock 1 1: ×1/32 clock.	R/W
b7, b6	EDGES[1:0]	Valid Edge Select	b7 b6 0 0: Rising edge 0 1: Falling edge 1 0: Both rising and falling edges 1 1: Setting prohibited.	R/W

Note 1. 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 (Measurement Target Clock Frequency Division Ratio 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.

10.2.3 CAC Control Register 2 (CACR2)

Address(es): CAC.CACR2 4004 4602h

	b7	b6	b5	b4	b3	b2	b1	b0
	DFS[1:0]		RCDS[1:0]		RSCS[2:0]			RPS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RPS	Reference Signal Select	0: CACREF pin input 1: Internal clock (internally generated signal).	R/W
b3 to b1	RSCS[2:0]	Measurement Reference Clock Select	b3 b1 0 0 0: Main clock oscillator 0 0 1: Sub-clock oscillator 0 1 0: HOCO clock 0 1 1: MOCO clock 1 0 0: LOCO clock 1 0 1: Peripheral module clock (PCLKB) 1 1 0: IWDTCLK clock 1 1 1: Setting prohibited.	R/W
b5, b4	RCDS[1:0]	Measurement Reference Clock Frequency Division Ratio Select	b5 b4 0 0: ×1/32 clock 0 1: ×1/128 clock 1 0: ×1/1024 clock 1 1: ×1/8192 clock.	R/W
b7, b6	DFS[1:0]	Digital Filter Select	b7 b6 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 1. 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 division ratio of the reference clock when an internal reference clock is selected (RPS = 1). 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 select its sampling clock.

10.2.4 CAC Interrupt Control Register (CAICR)

Address(es): CAC.CAICR 4004 4603h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	OVFFCL	MENDFCL	FERRFCL	—	OVFIE	MENDIE	FERRIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	FERRIE	Frequency Error Interrupt Request Enable	0: Disable 1: Enable.	R/W
b1	MENDIE	Measurement End Interrupt Request Enable	0: Disable 1: Enable.	R/W
b2	OVFIE	Overflow Interrupt Request Enable	0: Disable 1: Enable.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	FERRFCL	FERRF Clear	When 1 is written to this bit, the FERRF flag is cleared. This bit is read as 0.	R/W
b5	MENDFCL	MENDF Clear	When 1 is written to this bit, the MENDF flag is cleared. This bit is read as 0.	R/W
b6	OVFFCL	OVFF Clear	When 1 is written to this bit, the OVFF flag is cleared. This bit is read as 0.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

FERRIE bit (Frequency Error Interrupt Request Enable)

The FERRIE bit specifies whether the frequency error interrupt request is enabled.

MENDIE bit (Measurement End Interrupt Request Enable)

The MENDIE bit specifies whether the measurement end interrupt request is enabled.

OVFIE bit (Overflow Interrupt Request Enable)

The OVFIE bit specifies whether the overflow interrupt request is enabled.

FERRFCL bit (FERRF Clear)

Setting the FERRFCL bit to 1 clears the FERRF flag.

MENDFCL bit (MENDF Clear)

Setting the MENDFCL bit to 1 clears the MENDF flag.

OVFFCL bit (OVFF Clear)

Setting the OVFFCL bit to 1 clears the OVFF flag.

10.2.5 CAC Status Register (CASTR)

Address(es): CAC.CASTR 4004 4604h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	OVFF	MENDF	FERRF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	FERRF	Frequency Error Flag	0: The clock frequency is within the allowable range 1: The clock frequency has deviated beyond the allowable range (frequency error).	R
b1	MENDF	Measurement End Flag	0: Measurement is in progress 1: Measurement ended.	R
b2	OVFF	Overflow Flag	0: The counter has not overflowed 1: The counter overflowed.	R
b7 to b3	—	Reserved	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)

This flag indicates the end of measurement.

[Setting condition]

- Measurement ends.

[Clearing condition]

- 1 is written to the MENDFCL bit.

OVFF flag (Overflow Flag)

This flag indicates that the counter has overflowed.

[Setting condition]

- The counter overflows.

[Clearing condition]

- 1 is written to the OVFFCL bit.

10.2.6 CAC Upper-Limit Value Setting Register (CAULVR)

Address(es): CAC.CAULVR 4004 4606h

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

CAULVR is a 16-bit read/write register that specifies the upper value of the allowable range. When the counter value rises above 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.

10.2.7 CAC Lower-Limit Value Setting Register (CALLVR)

Address(es): CAC.CALLVR 4004 4608h

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

CALLVR 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.

10.2.8 CAC Counter Buffer Register (CACNTBR)

Address(es): CAC.CACNTBR 4004 460Ah

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

CACNTBR is a 16-bit read-only register that retains the measurement result.

10.3 Operation

10.3.1 Measuring Clock Frequency

The CAC measures the clock frequency with the CACREF pin input or the internal clock as a reference. Figure 10.2 shows an operating example of the CAC.

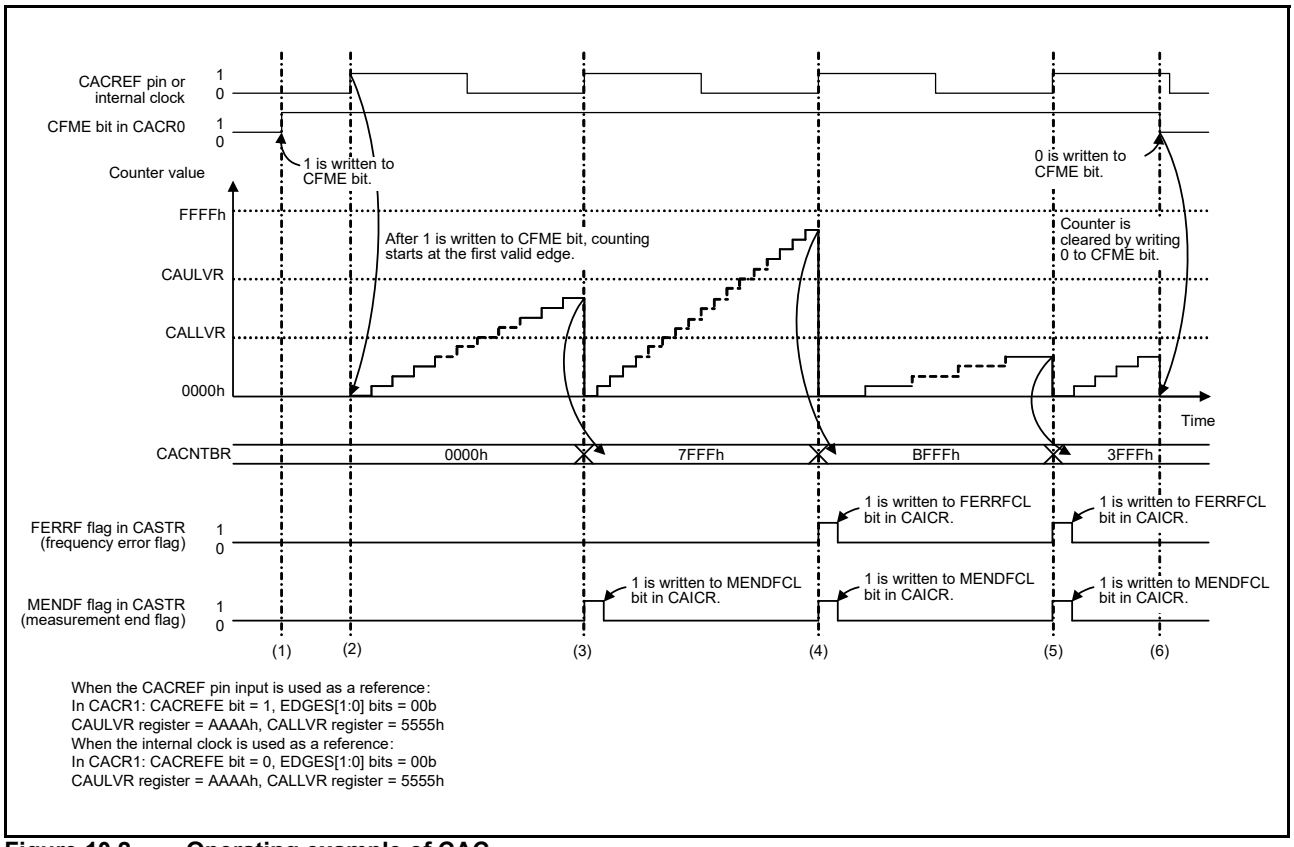


Figure 10.2 Operating example of CAC

1. Before writing 1 to CACR0.CFME, set CACR1 and CACR2 to define the measurement target clock and measurement reference clock. Writing 1 to the CACR0.CFME bit enables clock frequency measurement.
2. The timer starts counting up if the valid edge selected by the CACR1.EDGES[1:0] bits is input from the measurement reference clock. The valid edge is a rising edge (CACR1.EDGES[1:0] = 00b), as shown in Figure 10.2.
3. When the next valid edge is input, the counter value is transferred in CACNTBR and compared with the values of 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 counting up.

10.3.2 Digital Filtering of Signals on CACREF Pin

The CACREF pin has a digital filter, and levels on CACREF pin are transmitted to the internal circuitry after three consecutive matches at 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 in 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 by the following formula:

Counter value error = (1 cycle of the count source clock) / (1 cycle of the sampling clock)

10.4 Interrupt Requests

The CAC generates three interrupt requests:

- Frequency error interrupt
- Measurement end interrupt
- Overflow interrupt.

When an interrupt source is generated, the associated status flag becomes 1. Table 10.3 provides information on the CAC interrupt requests.

Table 10.3 CAC interrupt requests

Interrupt request	Interrupt enable bit	Status flag	Interrupt source
Frequency error interrupt	CAICR.FERRIE	CASTR.FERRF	The result of comparing CACNTBR to 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	The counter overflows

10.5 Usage Note

10.5.1 Module Stop Function Setting

The CAC operation can be disabled or enabled with the Module Stop Control Register C (MSTPCRC). The CAC is initially stopped after reset. Releasing the module-stop state enables access to registers. For details, see section 11, Low Power Modes.

11. Low Power Modes

11.1 Overview

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

Table 11.1 lists the specifications of functions to reduce power consumption. Table 11.2 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 DMAC, DTC, and SRAM operate.

Table 11.1 Specifications of low power mode functions

Parameter	Specification
Reducing power consumption by switching clock signals	Frequency division ratio can be selected independently for the system clock (ICLK), peripheral module clock (PCLKA, PCLKB, PCLKC, PCLKD), external bus clock (BCLK), and flash interface clock (FCLK).*1
EBCLK output control	Selectable to BCLK output or high-level output
Module-stop	Peripheral module functions can be stopped independently
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. • Five operating power control modes are available: - High-speed mode - Middle-speed mode - Low-speed mode - Low-voltage mode - Subosc-speed mode.

Note 1. For details, see section 9, Clock Generation Circuit.

Table 11.2 Operating conditions of each low power mode (1 of 2)

Parameter	Sleep mode	Software Standby mode	Snooze mode*1
Transition condition	WFI instruction while SBYCR.SSBY = 0	WFI instruction while SBYCR.SSBY = 1	Snooze request in Software Standby mode. SNZCR.SNZE = 1
Canceling method	All interrupts. Any reset available in the mode.	Interrupts shown in Table 11.3. Any reset available in the mode.	Interrupts shown in Table 11.3. Any reset available in the mode.
State after cancellation by an interrupt	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)
State after cancellation by a reset	Reset state	Reset state	Reset state
Main clock oscillator	Selectable	Stop	Selectable*2
Sub-clock oscillator	Selectable	Selectable	Selectable
High-speed on-chip oscillator	Selectable	Stop	Selectable
Middle-speed on-chip oscillator	Selectable	Stop	Selectable
Low-speed on-chip oscillator	Selectable	Selectable	Selectable
IWDT-dedicated on-chip oscillator	Selectable*4	Selectable*4	Selectable*4
PLL	Selectable	Stop	Selectable*2
Oscillation stop detection function	Selectable	Operation prohibited	Operation prohibited
Clock/buzzer output function	Selectable	Selectable*3	Selectable
External Bus (EBCLK)	Selectable	Stop (Retained)	Operation prohibited
CPU	Stop (Retained)	Stop (Retained)	Stop (Retained)
SRAM (ECC SRAM included)	Selectable	Stop (Retained)	Selectable
Flash memory	Operating	Stop (Retained)	Stop (Retained)

Table 11.2 Operating conditions of each low power mode (2 of 2)

Parameter	Sleep mode	Software Standby mode	Snooze mode*1
DMA Controller (DMAC)	Selectable	Stop (Retained)	Operation prohibited
Data Transfer Controller (DTC)	Selectable	Stop (Retained)	Selectable
USB 2.0 Full-Speed Module (USBFS)	Selectable	Stop (Retained)*5	Operation prohibited*5
Watchdog Timer (WDT)	Selectable*4	Stop (Retained)	Stop (Retained)
Independent Watchdog Timer (IWDT)	Selectable*4	Selectable*4	Selectable*4
Realtime clock (RTC)	Selectable	Selectable	Selectable
Asynchronous General Purpose Timer (AGTn, n = 0, 1)	Selectable	Selectable*6	Selectable*6
14-Bit A/D Converter (ADC14)	Selectable	Stop (Retained)	Selectable*12
12-Bit D/A Converter (DAC12)	Selectable	Stop (Retained)	Selectable
Capacitive Touch Sensing Unit (CTSU)	Selectable	Stop (Retained)	Selectable
Segment LCD Controller (SLCDC)	Selectable	Selectable*7	Selectable
Data Operation Circuit (DOC)	Selectable	Stop (Retained)	Selectable
Serial Communications Interface (SCI0)	Selectable	Stop (Retained)	Selectable*10
Serial Communications Interface (SCIn, n = 1 to 4, 9)	Selectable	Stop (Retained)	Operation prohibited
I^2C Bus Interface (IIC0)	Selectable	Selectable	Operation prohibited
I^2C Bus Interface (IICn, n = 1, 2)	Selectable	Stop (Retained)	Operation prohibited
Event Link Controller (ELC)	Selectable	Stop (Retained)	Selectable*8
High-Speed Analog Comparator (ACMPHSn, n = 0, 1)	Selectable	Selectable*9	Selectable*9
Low Power Analog Comparator (ACMPLP0)	Selectable	Selectable*9	Selectable*9
Low Power Analog Comparator (ACMPLP1)	Selectable	Selectable*9	Selectable*9
Operational Amplifier (OPAMP)	Selectable	Selectable	Selectable
NMI, IRQn (n = 0 to 15) 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*11	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.

Note 1. All modules whose module-stop bits are 0 start as soon as PCLKs are supplied after entering Snooze mode. To avoid an increase in 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. When using SCI0 in Snooze mode, MOSCCR.MOSTP and PLLCR.PLLSTP bits must be 1.

Note 3. Stopped when the clock output source select bits (CKOCR.CKOSEL[2:0]) are set to a value other than 010b (LOCO) and 100b (SOSC).

Note 4. 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.

Note 5. Detection of USBFS resumption is possible.

Note 6. AGT0 operation is possible when 100b (LOCO) or 110b (SOSC) is selected in the AGT0.AGTMR1.TCK[2:0] bits. AGT1 operation is possible when 100b (LOCO), 110b (SOSC), or 101 (Underflow event signal from AGT0) is selected in the AGT1.AGTMR1.TCK[2:0] bits.

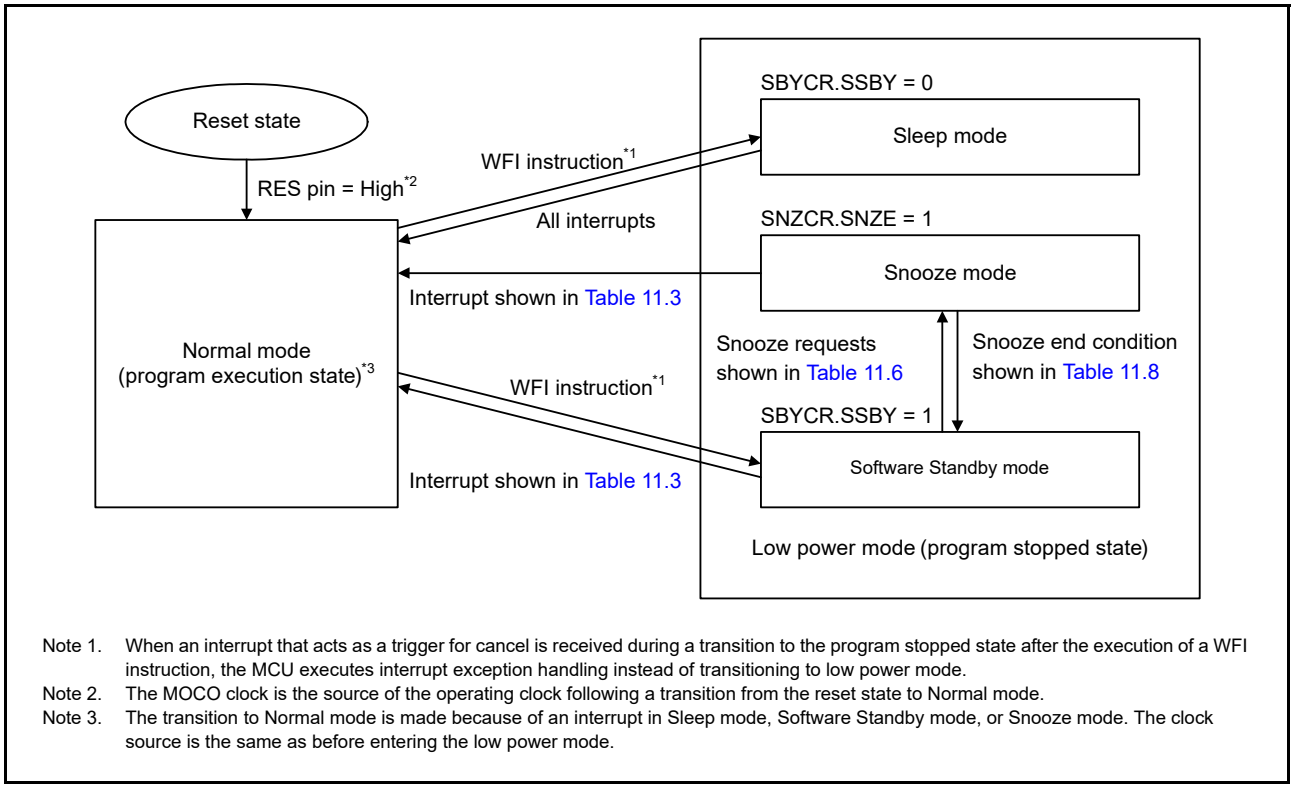
Note 7. Operation is possible when 000b (LOCO) or 001b (SOSC) is selected in the SLCDSCKCR.LCDSCKSEL[2:0] bits. Stopping is selected when the SLCDSCKCR.LCDSCKSEL[2:0] bits are set to a value other than 000b or 001b.

Note 8. Event lists the restrictions described in section 11.9.13, ELC Event in Snooze Mode.
Note 9. Only VCOUT function is permitted. The VCOUT pin operates when ACMPHS or ACMPLP uses no digital filter. For details on digital filter, see section 43, High-Speed Analog Comparator (ACMPHS) and section 44, Low Power Analog Comparator (ACMPLP).
Note 10. Serial communication modes of SCI0 is only in asynchronous mode.
Note 11. For the address bus and bus control signals (CS0 to CS3, RD, WR0 to WR1, WR, and BC0 to BC1), keeping the output state or changing to the high-impedance state can be selected by the SBYCR.OPE bit.
Note 12. When using the 14-Bit A/D Converter in Snooze mode, the ADCMPCR.CMPAE or ADCMPCR.CMPBE bit must be 1.

Table 11.3 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
VBATT	VBATT_LVD	Yes	Yes
Port	PORT_IRQn (n = 0 to 15)	Yes	Yes
LVD	LVD_LVD1	Yes	Yes
	LVD_LVD2	Yes	Yes
IWDT	IWDT_NMIUNDF	Yes	Yes
USBFS	USBFS_USBR	Yes	Yes
RTC	RTC_ALM	Yes	Yes
	RTC_PRD	Yes	Yes
KINT	KEY_INTKR	Yes	Yes
AGT1	AGT1_AGTI	Yes	Yes*3
	AGT1_AGTCMAI	Yes	Yes
	AGT1_AGTCMBI	Yes	Yes
ACMPLP	ACMP_LP0	Yes	Yes
IIC0	IIC0_WUI	Yes	No
ADC140	ADC140_WCMPM	No	Yes with SELSR0*1*3
	ADC140_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*3
DOC	DOC_DOPCI	No	Yes with SELSR0*1
CTSU	CTSU_CTSUFN	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 14, 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, whether the request can be accepted depends on the timing of the occurrence.
Note 2. Only one of either SCI0_AM or SCI0_RXI_OR_ERI can be selected.
Note 3. The event that is enabled by SNZEDCR must not be used.



Note 1. When an interrupt that acts as a trigger for cancel is received during a transition to the program stopped state after the execution of a WFI instruction, the MCU executes interrupt exception handling instead of transitioning to low power mode.
Note 2. The MOCO clock 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 because of an interrupt in Sleep mode, Software Standby mode, or Snooze mode. The clock source is the same as before entering the low power mode.

Figure 11.1 Mode transitions

11.2 Register Descriptions

11.2.1 Standby Control Register (SBYCR)

Address(es): SYSTEM.SBYCR 4001 E00Ch

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

Bit	Symbol	Bit name	Description	R/W
b13 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	OPE	Output Port Enable	0: In Software Standby mode, the address bus and bus control signals are set to the high-impedance state. In Snooze mode, the address bus and bus control signals are the same as before entering Software Standby mode. 1: In Software Standby mode, the address bus and bus control signals retain the output state.	R/W
b15	SSBY	Software Standby	0: Sleep mode 1: Software Standby mode.	R/W

OPE bit (Output Port Enable)

The OPE bit specifies whether to set to the high-impedance state or to retain the output of the address bus and bus control signals (CS0 to CS3, RD, WR0, WR1, WR, BC0, and BC1) in Software Standby or Snooze mode.

SSBY bit (Software Standby)

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 due to an interrupt, the SSBY bit remains 1. The SSBY bit can be cleared by writing 0 to it.

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

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

11.2.2 Module Stop Control Register A (MSTPCRA)

Address(es): SYSTEM.MSTPCRA 4001 E01Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	MSTPA22	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	MSTPA6	—	—	—	—	MSTPA1	MSTPA0
Value after reset:	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MSTPA0	SRAM0 Module Stop*1	Target module: SRAM0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b1	MSTPA1	SRAM1 Module Stop	Target module: SRAM1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b5 to b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b6	MSTPA6	ECCSRAM Module Stop*1	Target module: ECCSRAM 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b21 to b7	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b22	MSTPA22	DMA Controller/Data Transfer Controller Module Stop*2	Target module: DMAC, DTC 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b31 to b23	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. The MSTPA0 and MSTPA6 bit settings must be the same.
Note 2. When rewriting the MSTPA22 bit from 0 to 1, disable the DMAC and DTC before setting the MSTPA22 bit.

11.2.3 Module Stop Control Register B (MSTPCRB)

Address(es): MSTP.MSTPCRB 4004 7000h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPB31	MSTPB30	MSTPB29	MSTPB28	MSTPB27	—	—	—	—	MSTPB22	—	—	MSTPB19	MSTPB18	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	MSTPB11	—	MSTPB9	MSTPB8	MSTPB7	MSTPB6	MSTPB5	—	—	MSTPB2	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b2	MSTPB2	Controller Area Network Module Stop*1	Target module: CAN0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b4, b3	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Bit	Symbol	Bit name	Description	R/W
b5	MSTPB5	IrDA Module Stop	Target module: IrDA 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b6	MSTPB6	Quad Serial Peripheral Interface Module Stop	Target Module: QSPI 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b7	MSTPB7	I2C Bus Interface 2 Module Stop	Target module: IIC2 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b8	MSTPB8	I2C Bus Interface 1 Module Stop	Target module: IIC1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b9	MSTPB9	I2C Bus Interface 0 Module Stop	Target module: IIC0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b10	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b11	MSTPB11	Universal Serial Bus 2.0 FS Interface Module Stop*2	Target module: USBFS 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b17 to b12	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b18	MSTPB18	Serial Peripheral Interface 1 Module Stop	Target module: SPI1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b19	MSTPB19	Serial Peripheral Interface 0 Module Stop	Target module: SPI0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b21, b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b22	MSTPB22	Serial Communication Interface 9 Module Stop	Target module: SCI9 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b26 to b23	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b27	MSTPB27	Serial Communication Interface 4 Module Stop	Target module: SCI4 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b28	MSTPB28	Serial Communication Interface 3 Module Stop	Target module: SCI3 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b29	MSTPB29	Serial Communication Interface 2 Module Stop	Target module: SCI2 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b30	MSTPB30	Serial Communication Interface 1 Module Stop	Target module: SCI1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b31	MSTPB31	Serial Communication Interface 0 Module Stop	Target module: SCI0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W

Note 1. The MSTPB2 bit must be written while the oscillation of the clock controlled by this bit is stable. To enter Software Standby mode after writing to this bit, wait for 2 CAN clock (CANMCLK) cycles after writing, and then execute a WFI instruction.

Note 2. To enter Software Standby mode after writing to the MSTPB11 bit, wait for 2 USB clock (UCLK) cycles after writing, and then execute a WFI instruction.

11.2.4 Module Stop Control Register C (MSTPCRC)

Address(es): MSTP.MSTPCRC 4004 7004h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPC31	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	MSTPC14	MSTPC13	MSTPC12	—	—	—	MSTPC8	MSTPC7	—	—	MSTPC4	MSTPC3	—	MSTPC1	MSTPC0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b0	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
b1	MSTPC1	Cyclic Redundancy Check Calculator Module Stop	Target module: CRC 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b2	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b3	MSTPC3	Capacitive Touch Sensing Unit Module Stop	Target module: CTSU 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b4	MSTPC4	Segment LCD Controller Module Stop	Target module: SLCDC 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b6, b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	MSTPC7	Synchronous Serial Interface 1 Module Stop	Target module: SSI1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b8	MSTPC8	Synchronous Serial Interface 0 Module Stop	Target module: SSI0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b11 to b9	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b12	MSTPC12	Secure Digital Host Interface/Multi Media Card Interface Module Stop	Target module: SDHI/MMC 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b13	MSTPC13	Data Operation Circuit Module Stop	Target module: DOC 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b14	MSTPC14	Event Link Controller Module Stop	Target module: ELC 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b30 to b15	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b31	MSTPC31	SCE5 Module Stop*2	Target module: SCE5 0: Cancel the module-stop state 1: Enter the module-stop state.	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 among the clocks output by the oscillators, and then execute a WFI instruction.

Note 2. Set the MSTPC31 bit once to 0 at the beginning of the program to initialize the unused circuit even if the SCE5 is not used in this MCU. See section 11.9.15, Module-Stop Function for an Unused Circuit.

11.2.5 Module Stop Control Register D (MSTPCRD)

Address(es): MSTP.MSTPCRD 4004 7008h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPD31	MSTPD30	MSTPD29	MSTPD28	—	—	—	—	—	—	—	MSTPD20	—	—	—	MSTPD16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	MSTPD14	—	—	—	—	—	—	—	MSTPD6	MSTPD5	—	MSTPD3	MSTPD2	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b2	MSTPD2	Asynchronous General Purpose Timer 1 Module Stop*1	Target module: AGT1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b3	MSTPD3	Asynchronous General Purpose Timer 0 Module Stop*2	Target module: AGT0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b4	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b5	MSTPD5	General PWM Timer 323 to 320 Module Stop	Target module: GPT323 to GPT320 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b6	MSTPD6	General PWM Timer 329 to 324 Module Stop	Target module: GPT329 to GPT324 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b13 to b7	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b14	MSTPD14	Port Output Enable for GPT Module Stop	Target module: POEG 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b15	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b16	MSTPD16	14-Bit A/D Converter Module Stop	Target module: ADC140 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b19 to b17	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b20	MSTPD20	12-Bit D/A Converter Module Stop	Target module: DAC12 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b27 to b21	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b28	MSTPD28	High-Speed Analog Comparator 0 Module Stop	Target module: ACMPHS0 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b29	MSTPD29	Low Power Analog Comparator Module Stop	Target module: ACMPLP 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b30	MSTPD30	High-Speed Analog Comparator 1 Module Stop	Target module: ACMPHS1 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W
b31	MSTPD31	Operational Amplifier Module Stop	Target module: OPAMP 0: Cancel the module-stop state 1: Enter the module-stop state.	R/W

Note 1. When the count source is sub-clock oscillator or LOCO, AGT1 counting does not stop even if MSTPD2 is set to 1. If the count source is the sub-clock oscillator or LOCO, this bit must be set to 1 except when accessing the AGT1 registers.

Note 2. When the count source is sub-clock oscillator or LOCO, AGT0 counting does not stop even if MSTPD3 is set to 1. If the count source is the sub-clock oscillator or LOCO, this bit must be set to 1 except when accessing the AGT0 registers.

11.2.6 Operating Power Control Register (OPCCR)

Address(es): SYSTEM.OPCCR 4001 E0A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OPCMTSF	—	—	OPCM[1:0]	
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	OPCM[1:0]	Operating Power Control Mode Select	b1 b0 0 0: High-speed mode 0 1: Middle-speed mode 1 0: Low-voltage mode*1 1 1: Low-speed mode.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	OPCMTSF	Operating Power Control Mode Transition Status Flag	0: Transition complete 1: Transition in progress.	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. HOCOCR.HCSTP must always be 0.

The OPCCR register is used to reduce power consumption in Normal mode, Sleep mode and Snooze mode by specifying a lower operating frequency and operating voltage in the OPCCR setting.

For the procedure to change the operating power control modes, see section 11.5, 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 11.4 shows the relationship between the operating power control modes, and the OPCM[1:0] and SOPCM bits settings.

Writing to OPCCR.OPCM[1:0] is prohibited while HOCOCR.HCSTP and OSCSF.HOCOSF are 0 as the oscillation of the HOCO clock has not yet become stable.

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 is set to 1 when the OPCM bit is written, and to 0 when mode transition completes. Confirm that this flag is 0 before proceeding.

11.2.7 Sub Operating Power Control Register (SOPCCR)

Address(es): SYSTEM.SOPCCR 4001 E0AAh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SOPCMTSF	—	—	—	SOPCM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SOPCM	Sub Operating Power Control Mode Select	0: Not Subosc-speed mode 1: Subosc-speed mode.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SOPCMTSF	Sub Operating Power Control Mode Transition Status Flag	0: Transition complete 1: Transition in progress.	R
b7 to b5	—	Reserved	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 by initiating entry to and exit from Subosc-speed mode. Subosc-speed mode is available only when using the sub-clock oscillator or LOCO without dividing the frequency.

For the procedure to change operating power control modes, see section 11.5, 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 (operating mode set by OPCCR.OPCM[1:0]) before the transition to Subosc-speed mode.

Table 11.4 shows the relationship between the operating power control modes, the OPCM[1:0] and SOPCM bits settings.

SOPCMTSF flag (Sub Operating Power Control Mode Transition Status Flag)

The SOPCMTSF flag indicates the switching control state when the subosc operating power control mode is switched. This flag is set to 1 when the SOPCM bit is written, and to 0 when mode transition completes. Confirm that this flag is 0 before proceeding.

Table 11.4 shows each operating power control mode.

Table 11.4 Relationship between the power control mode and the OPCM[1:0] and SOPCM bit settings

Operating power control mode	OPCM[1:0] bits	SOPCM bit	Power consumption
High-speed mode	00b	0	High
Middle-speed mode	01b	0	↓
Low-voltage mode	10b	0	
Low-speed mode	11b	0	
Subosc-speed mode	xxb	1	Low

11.2.8 Snooze Control Register (SNZCR)

Address(es): SYSTEM.SNZCR 4001 E092h

	b7	b6	b5	b4	b3	b2	b1	b0
	SNZE	—	—	—	—	—	SNZDTCEN	RXDREQEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RXDREQEN	RXD0 Snooze Request Enable	0: Ignore the RXD0 falling edge in Software Standby mode 1: Detect the RXD0 falling edge in Software Standby mode.	R/W
b1	SNZDTCEN	DTC Enable in Snooze mode	0: Disable DTC operation in Snooze mode 1: Enable DTC operation in Snooze mode.	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SNZE	Snooze mode Enable	0: Disable Snooze mode 1: Enable Snooze mode.	R/W

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 is only available 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 (ICU Event Link Setting Register n).

SNZE bit (Snooze mode Enable)

The SNZE bit enables or disables 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, one of the triggers shown in Table 11.6 occurring 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, clear the SNZE bit once and then set it before re-entering Software Standby mode. For details, see section 11.8, Snooze Mode.

11.2.9 Snooze End Control Register (SNZEDCR)

Address(es): SYSTEM.SNZEDCR 4001 E094h

	b7	b6	b5	b4	b3	b2	b1	b0
	SCI0UMTED	—	—	AD0UMTED	AD0MATED	DTCNZRED	DTCZRED	AGTUNFED
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	AGTUNFED	AGT1 Underflow Snooze End Enable	0: Disable the Snooze end request 1: Enable the Snooze end request.	R/W
b1	DTCZRED	Last DTC Transmission Completion Snooze End Enable	0: Disable the Snooze end request 1: Enable the Snooze end request.	R/W
b2	DTCNZRED	Not Last DTC Transmission Completion Snooze End Enable	0: Disable the Snooze end request 1: Enable the Snooze end request.	R/W
b3	AD0MATED	ADC140 Compare Match Snooze End Enable	0: Disable the Snooze end request 1: Enable the Snooze end request.	R/W
b4	AD0UMTED	ADC140 Compare Mismatch Snooze End Enable	0: Disable the Snooze end request 1: Enable the Snooze end request.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SCI0UMTED	SCI0 Address Mismatch Snooze End Enable	0: Disable the Snooze end request 1: Enable the Snooze end request.	R/W

To use one of the triggers shown in Table 11.8 as a condition to switch from Snooze mode to Software Standby mode, set the associated bit in the SNZEDCR register to 1. The event that is used fore returning to Normal mode from Snooze mode listed in Table 11.3 must not be enabled in SNZEDCR.

AGTUNFED bit (AGT1 Underflow Snooze End Enable)

The AGTUNFED bit enables or disables a transition from Snooze mode to Software Standby mode on an AGT1 underflow. For details on the trigger conditions, see section 24, Asynchronous General Purpose Timer (AGT).

DTCZRED bit (Last DTC Transmission Completion Snooze End Enable)

The DTCZRED bit enables or disables a transition from Snooze mode to Software Standby mode on completion of the last DTC transmission, signaled when the CRA or CRB register in the DTC is 0. For details on the trigger conditions, see section 18, Data Transfer Controller (DTC).

DTCNZRED bit (Not Last DTC Transmission Completion Snooze End Enable)

The DTCNZRED bit enable or disables a transition from Snooze mode to Software Standby mode on completion of each DTC transmission, signaled when the CRA or CRB register in the DTC is not 0. For details on the trigger conditions, see section 18, Data Transfer Controller (DTC).

AD0MATED bit (ADC140 Compare Match Snooze End Enable)

The AD0MATED bit enables or disables a transition from Snooze mode to Software Standby mode on an ADC140 event when the conversion result matches the expected data. For details on the trigger conditions, see section 39, 14-Bit A/D Converter (ADC14).

AD0UMTED bit (ADC140 Compare Mismatch Snooze End Enable)

The AD0UMTED bit enables or disables a transition from Snooze mode to Software Standby mode on an ADC140 event when the conversion result does not match the expected data. For details on the trigger conditions, see section 39, 14-Bit A/D Converter (ADC14).

SCI0UMTED bit (SCI0 Address Mismatch Snooze End Enable)

The SCI0UMTED bit enables or disables 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 29, Serial Communications Interface (SCI). Only set this bit to 1 when SCI0 is operating in asynchronous mode.

11.2.10 Snooze Request Control Register (SNZREQCR)

Address(es): SYSTEM.SNZREQCR 4001 E098h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	SNZREQEN30	SNZREQEN29	SNZREQEN28	—	—	SNZREQEN25	SNZREQEN24	SNZREQEN23	—	—	—	—	—	SNZREQEN17	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SNZREQEN15	SNZREQEN14	SNZREQEN13	SNZREQEN12	SNZREQEN11	SNZREQEN10	SNZREQEN9	SNZREQEN8	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	Bit name	Description	R/W
b0	SNZREQEN0	Snooze Request Enable 0	Enables IRQ0 pin Snooze request: 0: Disable 1: Enable.	R/W
b1	SNZREQEN1	Snooze Request Enable 1	Enables IRQ1 pin Snooze request: 0: Disable 1: Enable.	R/W
b2	SNZREQEN2	Snooze Request Enable 2	Enables IRQ2 pin Snooze request: 0: Disable 1: Enable.	R/W
b3	SNZREQEN3	Snooze Request Enable 3	Enables IRQ3 pin Snooze request: 0: Disable 1: Enable.	R/W
b4	SNZREQEN4	Snooze Request Enable 4	Enables IRQ4 pin Snooze request: 0: Disable 1: Enable.	R/W
b5	SNZREQEN5	Snooze Request Enable 5	Enables IRQ5 pin Snooze request: 0: Disable 1: Enable.	R/W
b6	SNZREQEN6	Snooze Request Enable 6	Enables IRQ6 pin Snooze request: 0: Disable 1: Enable.	R/W
b7	SNZREQEN7	Snooze Request Enable 7	Enables IRQ7 pin snooze request: 0: Disable 1: Enable.	R/W
b8	SNZREQEN8	Snooze Request Enable 8	Enables IRQ8 pin snooze request: 0: Disable 1: Enable.	R/W
b9	SNZREQEN9	Snooze Request Enable 9	Enables IRQ9 pin snooze request: 0: Disable 1: Enable.	R/W
b10	SNZREQEN10	Snooze Request Enable 10	Enables IRQ10 pin snooze request: 0: Disable 1: Enable.	R/W
b11	SNZREQEN11	Snooze Request Enable 11	Enables IRQ11 pin snooze request: 0: Disable 1: Enable.	R/W
b12	SNZREQEN12	Snooze Request Enable 12	Enables IRQ12 pin snooze request: 0: Disable 1: Enable.	R/W
b13	SNZREQEN13	Snooze Request Enable 13	Enables IRQ13 pin snooze request: 0: Disable 1: Enable.	R/W

Bit	Symbol	Bit name	Description	R/W
b14	SNZREQEN14	Snooze Request Enable 14	Enables IRQ14 pin snooze request: 0: Disable 1: Enable.	R/W
b15	SNZREQEN15	Snooze Request Enable 15	Enables IRQ15 pin snooze request: 0: Disable 1: Enable.	R/W
b16	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b17	SNZREQEN17	Snooze Request Enable 17	Enables Key Interrupt snooze request: 0: Disable 1: Enable.	R/W
b22 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23	SNZREQEN23	Snooze Request Enable 23	Enables ACMPLP0 snooze request: 0: Disable 1: Enable.	R/W
b24	SNZREQEN24	Snooze Request Enable 24	Enables RTC alarm snooze request: 0: Disable 1: Enable.	R/W
b25	SNZREQEN25	Snooze Request Enable 25	Enables RTC period snooze request: 0: Disable 1: Enable.	R/W
b27, b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	SNZREQEN28	Snooze Request Enable 28	Enables AGT1 underflow snooze request: 0: Disable 1: Enable.	R/W
b29	SNZREQEN29	Snooze Request Enable 29	Enables AGT1 compare match A snooze request: 0: Disable 1: Enable.	R/W
b30	SNZREQEN30	Snooze Request Enable 30	Enables AGT1 compare match B snooze request: 0: Disable 1: Enable.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The SNZREQCR 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 14, Interrupt Controller Unit (ICU), the MCU enters Normal mode when the trigger is generated while the associated bit of the SNZREQCR register is 1. The WUPEN register settings always have a higher priority than the SNZREQCR register settings. For details, see section 11.8, Snooze Mode and section 14, Interrupt Controller Unit (ICU).

11.2.11 Flash Operation Control Register (FLSTOP)

Address(es): SYSTEM.FLSTOP 4001 E09Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	FLSTPF	—	—	—	FLSTOP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	FLSTOP	Selecting ON/OFF of the Flash Memory Operation	0: Operate code flash and data flash memory 1: Stop code flash and data flash memory.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	FLSTPF	Flash Memory Operation Status Flag	0: Transition completes 1: Transition in progress (from the flash-stop-status to flash-operating-status or flash-operating-status to flash-stop-status).	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FLSTOP bit (Selecting ON/OFF of the Flash Memory Operation)

The FLSTOP bit enables or disables flash memory. The FLSTOP bit must be written in a program executing in the SRAM. To use an interrupt when the FLSTOP bit is 1, be sure to place the interrupt vector in the SRAM. Set this bit to 0 when low-voltage mode is not selected.

Note 1. When changing the value of the FLSTOP bit from 1 to 0 to start flash memory operation, confirm that the FLSTPF flag is 0 and OSCSF.HOCOSF is 1 before restarting access to the flash memory. Instructions can then be executed in the code flash memory.

Note 2. Writing to FLSTOP.FLSTOP is prohibited while HOCOCR.HCSTP and OSCSF.HOCOSF are 0 (HOCO is in stabilization wait counting).

FLSTPF flag (Flash Memory Operation Status Flag)

The FLSTPF flag indicates the status of the transition from the flash-stop-status to flash-operating-status or from the flash-operating-status to the flash-stop-status. When the transition completes, the flag is read as 0. When using flash memory again after stopping it once, make sure that the FLSTPF flag is 0 before proceeding.

11.2.12 Power Save Memory Control Register (PSMCR)

Address(es): SYSTEM.PSMCR 4001 E09Fh

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

Bit	Symbol	Bit name	Description	R/W
b1, b0	PSMC[1:0]	Power Save Memory Control	b1 b0 0 0: All SRAMs are on in Software Standby mode 0 1: 48 KB SRAM (2000 0000h to 2000 BFFFh) is on in Software Standby mode 1 0: Setting prohibited 1 1: Setting prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PSMC[1:0] bits (Power Save Memory Control)

The PSMC[1:0] bits select the SRAM retention area in Software Standby mode. Setting these bits to 01b (48 KB SRAM in Software Standby mode) reduces the supply current. Set the PSMC register before executing a WFI instruction.

11.2.13 System Control OCD Control Register (SYOCDCR)

Address(es): SYSTEM.SYOCDCR 4001 E40Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	DBGEN	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DBGEN	Debugger Enable bit	0: On-chip debugger is disabled 1: On-chip debugger is enabled. Set to 1 first in on-chip debug mode.	R/W

DBGEN bit (Debugger Enable bit)

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.

11.3 Reducing Power Consumption by Switching Clock Signals

The clock frequency changes when the SCKDIVCR.FCK[2:0], ICK[2:0], BCK[2:0], PCKA[2:0], PCKB[2:0], PCKC[2:0], and PCKD[2:0] bits are set. The CPU, DMAC, DTC, flash memory, and SRAM use the operating clock specified by the ICK[2:0] bits.

Peripheral modules use the operating clock specified in the PCKA[2:0], PCKB[2:0], PCKC[2:0], and PCKD[2:0] bits.

The flash memory interface uses the operating clock specified in the FCK[2:0] bits.

The external bus uses the operating clock specified in the BCK[2:0] bits.

For details, see section 9, Clock Generation Circuit.

11.4 Module-Stop Function

The module-stop function can be set for each on-chip peripheral module. When the MSTPmi bit (m = A to D; i = 31 to 0) in MSTPCRA to MSTPCRD is set to 1, the specified module stops operating and enters the module-stop state, but the CPU continues to operate independently. Clearing the MSTPmi bit to 0 cancels the module-stop state, allowing the module to resume operation at the end of the bus cycle. The internal states of the modules are retained in the module-stop state.

After a reset is canceled, all modules other than the DMAC, DTC, and SRAMs are placed in the module-stop state. Do not access the module while the corresponding MSTPmi bit is 1. Otherwise, the read/write data and the operation of the module is not guaranteed. Do not set the MSTPmi bit to 1 while the corresponding module is accessed.

11.5 Function for Lower Operating Power Consumption

Power consumption can be reduced in Normal mode, Sleep mode and Snooze mode by selecting an appropriate operating power mode for the given operating frequency and operating voltage.

11.5.1 Setting Operating Power Control Mode

Make sure that the operating conditions such as the voltage range and the frequency ranges are always within the specified ranges before and after switching the operating power control modes. This section provides example procedures for switching operating power control modes.

Table 11.5 Oscillators available in each mode

	Oscillator						
Mode	PLL*1	High-speed on-chip oscillator	Middle-speed on-chip oscillator	Low-speed on-chip oscillator	Main clock oscillator	Sub-clock oscillator	IWDT-dedicated on-chip oscillator
High-speed	Available	Available	Available	Available	Available	Available	Available
Middle-speed	Available	Available	Available	Available	Available	Available	Available
Low-voltage	N/A	Available	Available	Available	Available	Available	Available
Low-speed	N/A	Available	Available	Available	Available	Available	Available
Subosc-speed	N/A	N/A	N/A	Available	N/A	Available	Available

Note 1. The VCC range for the PLL is 2.4 to 5.5 V.

(1) Switching from a higher to a lower power mode

Example 1: To switch from High-speed mode to Low-speed mode:

Operation begins in High-speed mode.

1. Disable the flash cache by resetting FCACHEE.FCACHEEN when the flash cache is cacheable in High-speed mode.
2. Change the oscillator to what is used in Low-speed mode. Set the frequency of each clock lower than the maximum operating frequency in Low-speed mode.
3. Turn off the oscillator that is not required in Low-speed mode.
4. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
5. Set the OPCCR.OPCM bit to 11b (Low-speed mode).
6. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
7. Perform the following steps when the flash cache is cacheable in Low-speed mode:
   a. Invalidate the flash cache by setting FCACHEIV.FCACHEIV.
   b. Check that FCACHEIV.FCACHEIV is 0.
   c. Enable the flash cache by setting FCACHEE.FCACHEEN.

Operation is now in Low-speed mode.

Example 2: To switch from High-speed mode to Subosc-speed mode:

Operation begins in High-speed mode.

1. Disable the flash cache by resetting FCACHEE.FCACHEEN when the flash cache is cacheable in High-speed mode.
2. Switch the clock source to sub-clock oscillator. Turn off HOCO, MOCO, main oscillator and PLL.
3. Confirm that all clock sources other than the sub-clock oscillator are stopped.
4. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).

5. Set the SOPCCR.SOPCM bit to 1 (Subosc-speed mode).
6. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).
7. Perform the following steps when the flash cache is cacheable in Subosc-speed mode:
   a. Invalidate the flash cache by setting FCACHEIV.FCACHEIV.
   b. Check that FCACHEIV.FCACHEIV is 0.
   c. Enable the flash cache by setting FCACHEE.FCACHEEN.

Operation is now in Subosc-speed mode.

(2) Switching from a lower power mode to a higher power mode

Example 1: To switch from Subosc-speed mode to High-speed mode:

Operation begins in Subosc-speed mode.

1. Disable the flash cache by resetting FCACHEE.FCACHEEN when the flash cache is cacheable in Subosc-speed mode.
2. Confirm that the SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).
3. Set SOPCCR.SOPCM bit to 0 (High-speed mode).
4. Confirm that SOPCCR.SOPCMTSF flag is 0 (indicates transition completed).
5. Turn on the oscillator needed in High-speed mode.
6. Set the frequency of each clock to lower than the maximum operating frequency for High-speed mode.
7. Perform the following steps when the flash cache is cacheable in High-speed mode:
   a. Invalidate the flash cache by setting FCACHEIV. FCACHEIV.
   b. Check that FCACHEIV.FCACHEIV is 0.
   c. Enable the flash cache by setting FCACHEE.FCACHEEN.

Operation is now in High-speed mode.

Example 2: To switch from Low-speed mode to High-speed mode:

Operation begins in Low-speed mode.

1. Disable the flash cache by resetting FCACHEE.FCACHEEN when the flash cache is cacheable in Low-speed mode.
2. Confirm that OPCCR.OPCMTSF flag is 0 (indicates transition completed).
3. Set the OPCCR.OPCM bit to 00b (High-speed mode).
4. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
5. Turn on any oscillator needed in High-speed mode.
6. Set the frequency of each clock to lower than the maximum operating frequency for High-speed mode.
7. Perform the following steps when the flash cache is cacheable in High-speed mode:
   a. Invalidate the flash cache by setting FCACHEIV.FCACHEIV.
   b. Check that FCACHEIV.FCACHEIV is 0.
   c. Enable the flash cache by setting FCACHEE.FCACHEEN.

Operation is now in High-speed mode.

11.5.2 Operating range

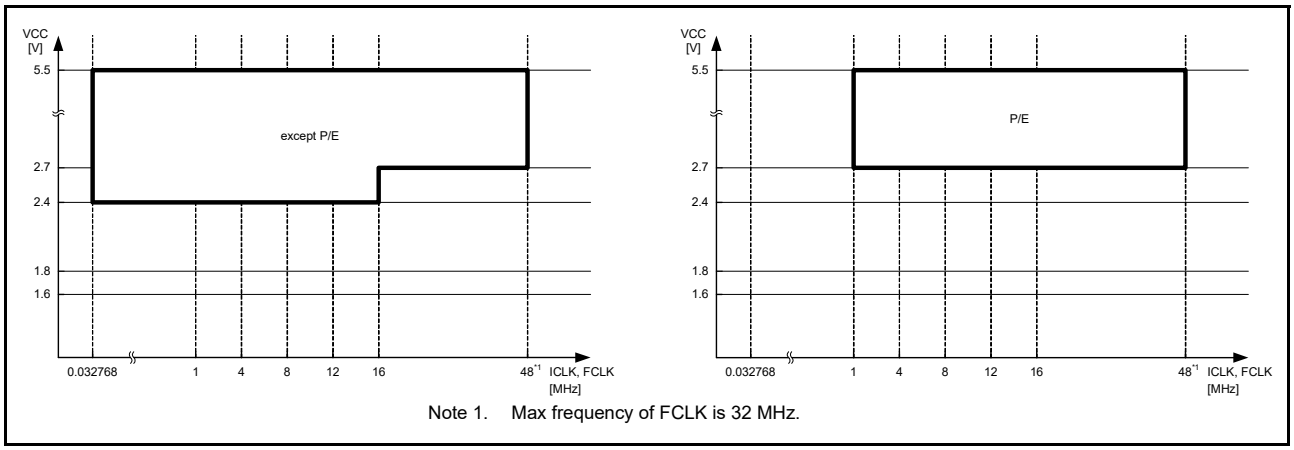
High-speed mode

During flash read, the maximum operating frequency is 48 MHz for ICLK and 32 MHz for FCLK, and the operating voltage range is 2.4 to 5.5 V. However, the maximum operating frequency for ICLK and FCLK during flash read is 16 MHz when the operating voltage is 2.4 V or larger and smaller than 2.7 V.

During flash programming/erasure, the operating frequency range is 1 to 48 MHz and the operating voltage range is 2.7 to 5.5 V.

The PLL can be used when the operating voltage is 2.4 V or above.

Figure 11.2 shows the operating voltages and frequencies in High-speed mode.



Note 1. Max frequency of FCLK is 32 MHz.

Figure 11.2 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 flash read is 12 MHz for ICLK and FCLK. The operating voltage range is 1.8 to 5.5 V during flash read. However, for ICLK and FCLK, the maximum operating frequency during flash read is 8 MHz when the operating voltage is 1.8 V or larger and smaller than 2.4 V.

During flash programming/erasure, the operating frequency range is 1 to 12 MHz and the operating voltage range is 1.8 to 5.5 V. The maximum operating frequency during flash programming/erasure is 8 MHz when the operating voltage is 1.8 V or larger and smaller than 2.4 V.

The PLL can be used when the operating voltage is 2.4 V or above.

Figure 11.3 shows the operating voltages and frequencies in Middle-speed mode.

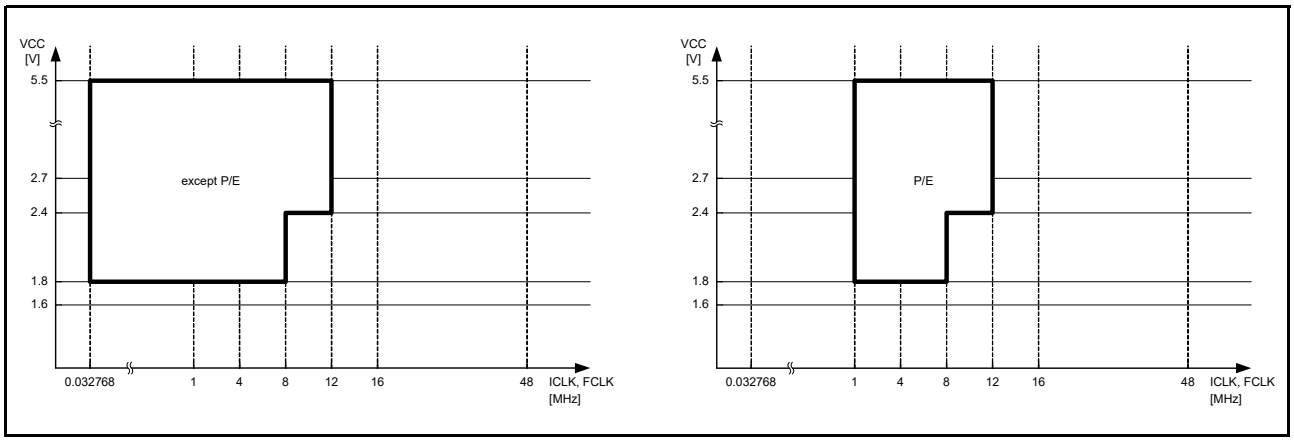


Figure 11.3 Operating voltages and frequencies in Middle-speed mode

Low-voltage mode

After a reset is canceled, operation is started in the low-voltage mode. Using the PLL is prohibited in this mode. The maximum operating frequency during flash read is 4 MHz for ICLK and FCLK. The operating voltage range is 1.6 to 5.5 V during flash read. During flash programming/erasure, the operating frequency range is 1 to 4 MHz and the operating voltage range is 1.8 to 5.5 V.

Figure 11.4 shows the operating voltages and frequencies in Low-voltage mode.

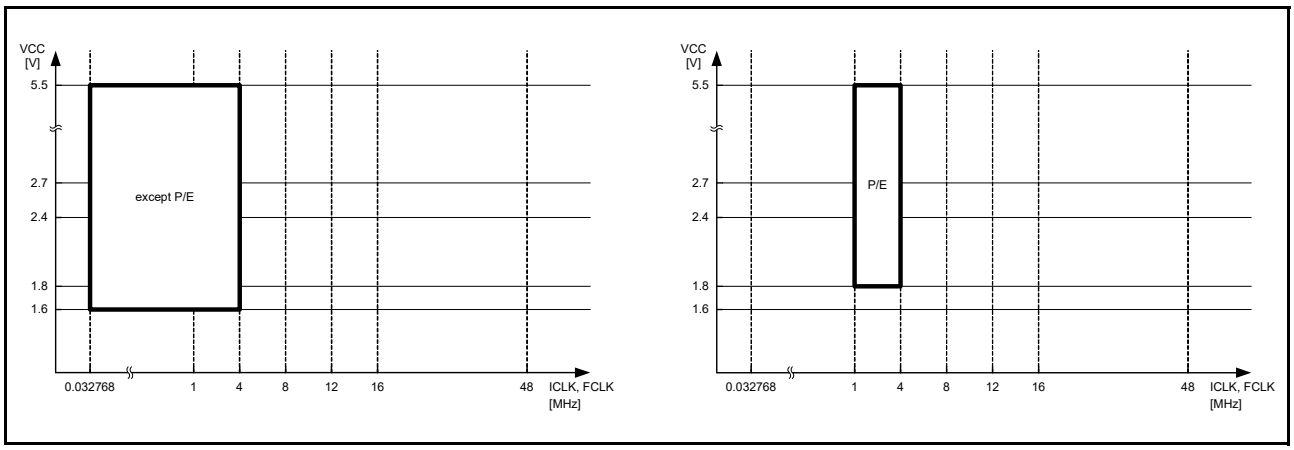


Figure 11.4 Operating voltages and frequencies in Low-voltage mode

Low-speed mode

The maximum operating frequency during flash read is 1 MHz for ICLK and FCLK. The operating voltage range is 1.8 to 5.5 V during flash read. P/E operations for flash memory are prohibited. Using the PLL is also prohibited.

Figure 11.5 shows the operating voltages and frequencies in Low-speed mode.

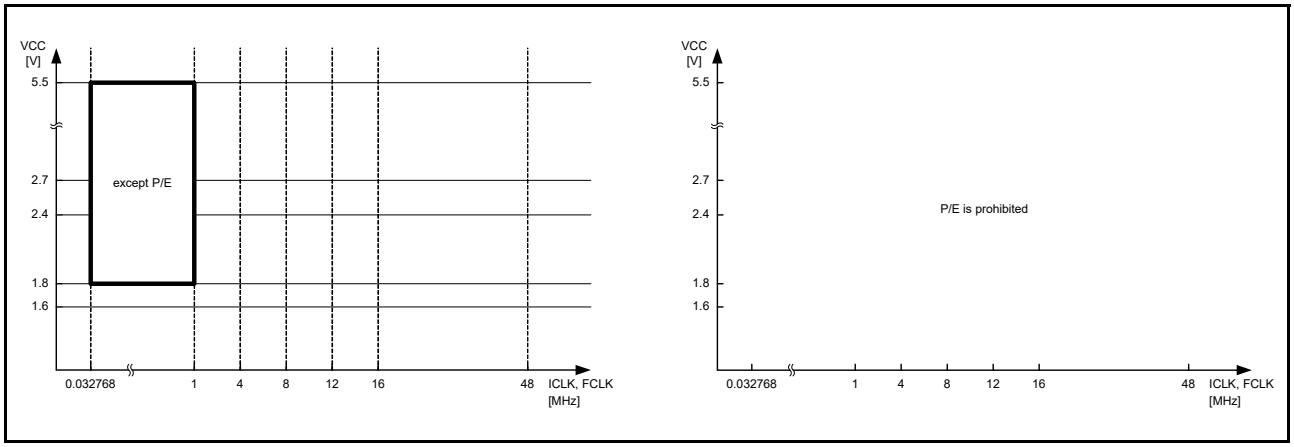


Figure 11.5 Operating voltages and frequencies in Low-speed mode

Subosc-speed mode

The maximum operating frequency during flash read is 37.6832 kHz for ICLK and FCLK. The operating voltage range is 1.8 to 5.5 V during flash read.

P/E operations for flash memory are prohibited. Using the oscillators other than the sub-clock oscillator or low-speed on-chip oscillator is also prohibited.

Figure 11.6 shows the operating voltages and frequencies in Subosc-speed mode.

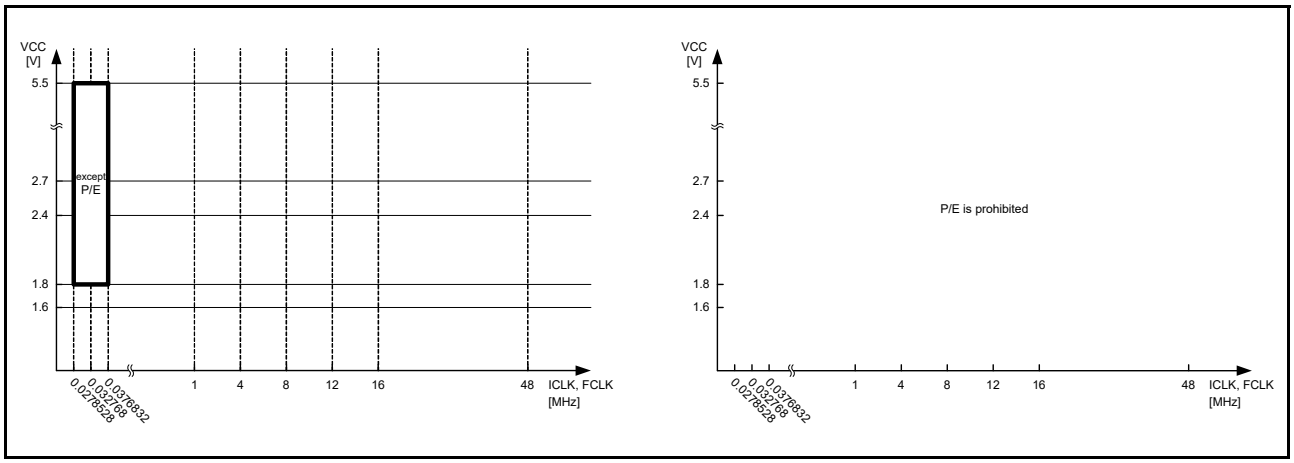


Figure 11.6 Operating voltages and frequencies in Subosc-speed mode

11.6 Sleep Mode

11.6.1 Transition 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 14, 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). In the same way, counting by WDT stops when the MCU enters Sleep mode while the WDT is in register start mode and the WDTCSTPR.SLCSTP bit in 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). In the same way, 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).

11.6.2 Canceling Sleep Mode

Sleep mode is canceled by any interrupt, RES pin reset, power-on reset, voltage monitor reset, SRAM parity error reset, SRAM ECC error reset, Bus master MPU error reset, Bus slave MPU error reset, or reset caused by an IWDT or a WDT underflow.

You can cancel the Sleep mode in any of the following ways:

1. Canceling by an interrupt
   When an available interrupt request is generated, Sleep mode is canceled and the MCU starts the interrupt handling.
2. Canceling by RES pin reset
   When RES pin is driven low, the MCU enters the reset state. You must keep the RES pin low for the time period specified in section 52, Electrical Characteristics. When 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 condition:
   - 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 the correct interrupt settings, see section 14, Interrupt Controller Unit (ICU).

11.7 Software Standby Mode

11.7.1 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 11.2 shows the status of the on-chip peripheral functions and oscillators. Available resets or interrupts in Software Standby mode cause the MCU to cancel Software Standby mode. See Table 11.3 for available interrupt sources and section 14.2.9, Wake Up Interrupt Enable Register (WUPEN) 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 14, Interrupt Controller Unit (ICU).

Clear the DMAST.DMST and DTCST.DTCST bits 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.

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 or Snooze mode).

WDT stops counting when the MCU enters Software Standby mode.

Do not enter Software Standby mode while OSTDCR.OSTDE is 1 (oscillation stop detection function is enabled). Before entering Software Standby mode, execute a WFI instruction after disabling the oscillation stop detection function (OSTDCR.OSTDE = 0). If a WFI instruction is executed while OSTDCR.OSTDE is 1, the MCU enters Sleep mode even when SBYCR.SSBY is 1. Also, do not enter Software Standby mode while the flash memory performs a programming or erasing procedure. Before entering Software Standby mode, execute a WFI instruction after the programming or erasing procedure completes.

11.7.2 Canceling Software Standby Mode

Software Standby mode is canceled by an available interrupt shown in Table 11.3, RES pin reset, power-on reset, voltage monitor reset, or reset caused by an IWDT underflow.

The oscillators that operate before the transition to Software Standby mode restart. After all the oscillators are stabilized, the MCU returns to Normal mode from Software Standby mode. See section 14.2.9, Wake Up Interrupt Enable Register (WUPEN) 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 (for available interrupts, see Table 11.3) is generated, an oscillator that was operating 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 RES pin is driven low, the MCU enters the reset state, and the oscillators in default status start operating. You must keep the RES pin low for the time period specified in section 52, 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.

11.7.3 Software Standby Mode Operation Example

Figure 11.7 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 on a rising edge of the IRQn pin.

In this example, an IRQn pin interrupt is accepted when the IRQCRi.IRQMD[1:0] bits of the ICU are set to 01b (falling edge) in Normal mode, and then set to 10b (rising edge). Next, the SBYCR.SSBY bit is set to 1 and a WFI instruction is executed. Entry to Software Standby mode completes, and then 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 14, Interrupt Controller Unit (ICU). The oscillation stabilization time in Figure 11.7 is specified in section 52, Electrical Characteristics.

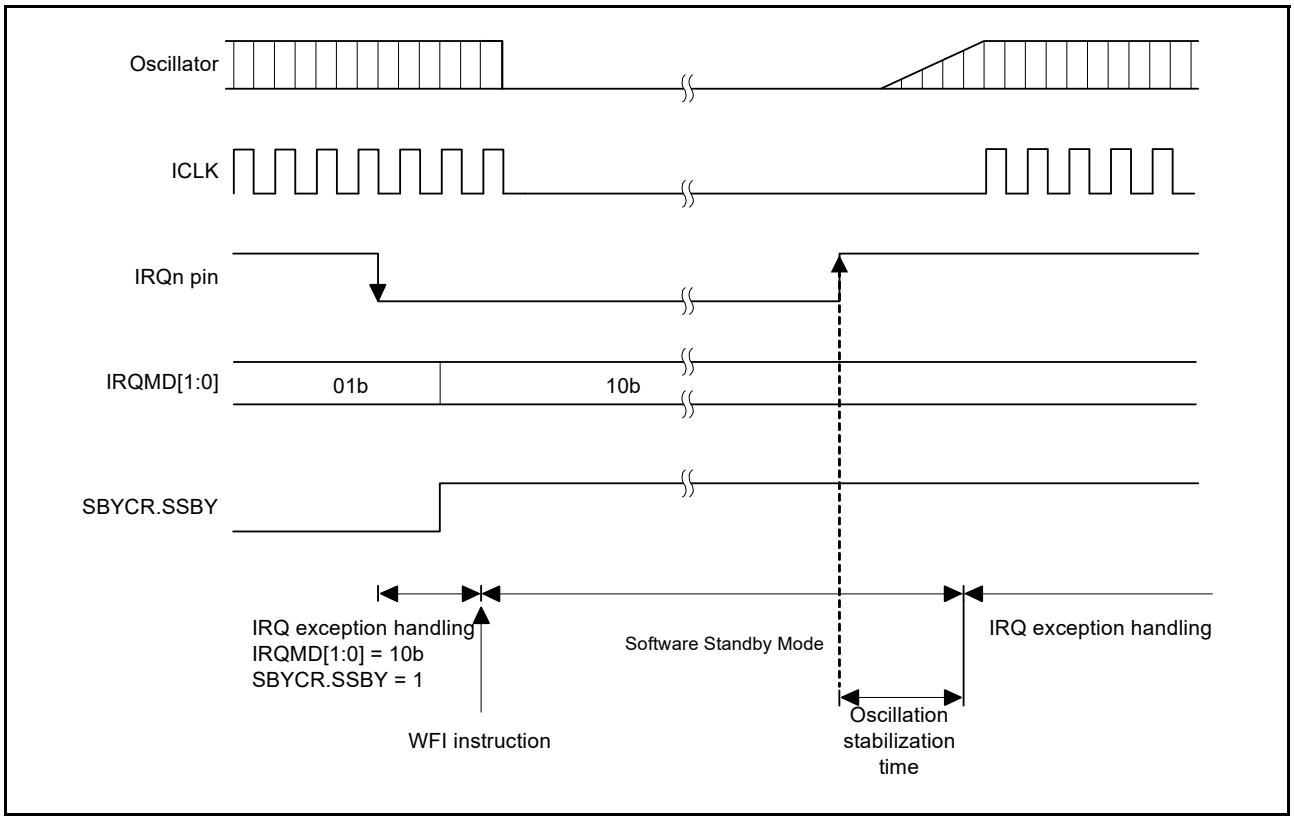


Figure 11.7 Example of Software Standby mode application

11.8 Snooze Mode

11.8.1 Transition to Snooze Mode

Figure 11.8 shows Snooze mode entry configuration. When the Snooze control circuit receives a Snooze request in Software Standby mode, the MCU transitions to Snooze mode. In this mode, some peripheral modules operate without waking up the CPU. The peripheral modules that can operate in Snooze mode are shown in Table 11.2, Operating conditions of each low power mode. Also, DTC operation can be selected in Snooze mode by setting the SNZCR.SNZDTCEN bit.

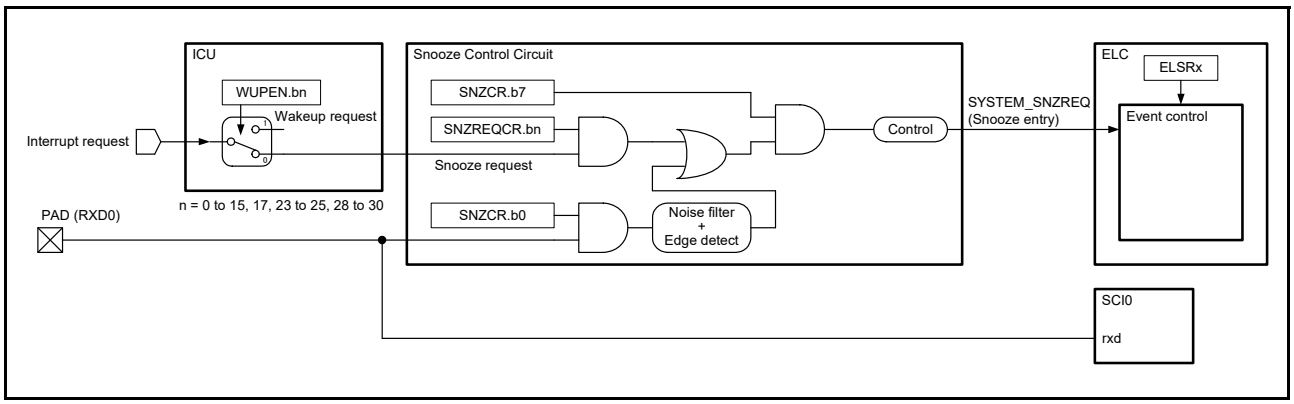


Figure 11.8 Snooze mode entry configuration

Table 11.6 shows the Snooze requests that switch the MCU from Software Standby mode to Snooze mode. To use a listed Snooze request as a trigger to switch to Snooze mode, the associated SNZREQENn bit in the SNZREQCR register or RXDREQEN bit in the SNZCR register must be set before entering Software Standby mode. Do not enable multiple Snooze requests at the same time.

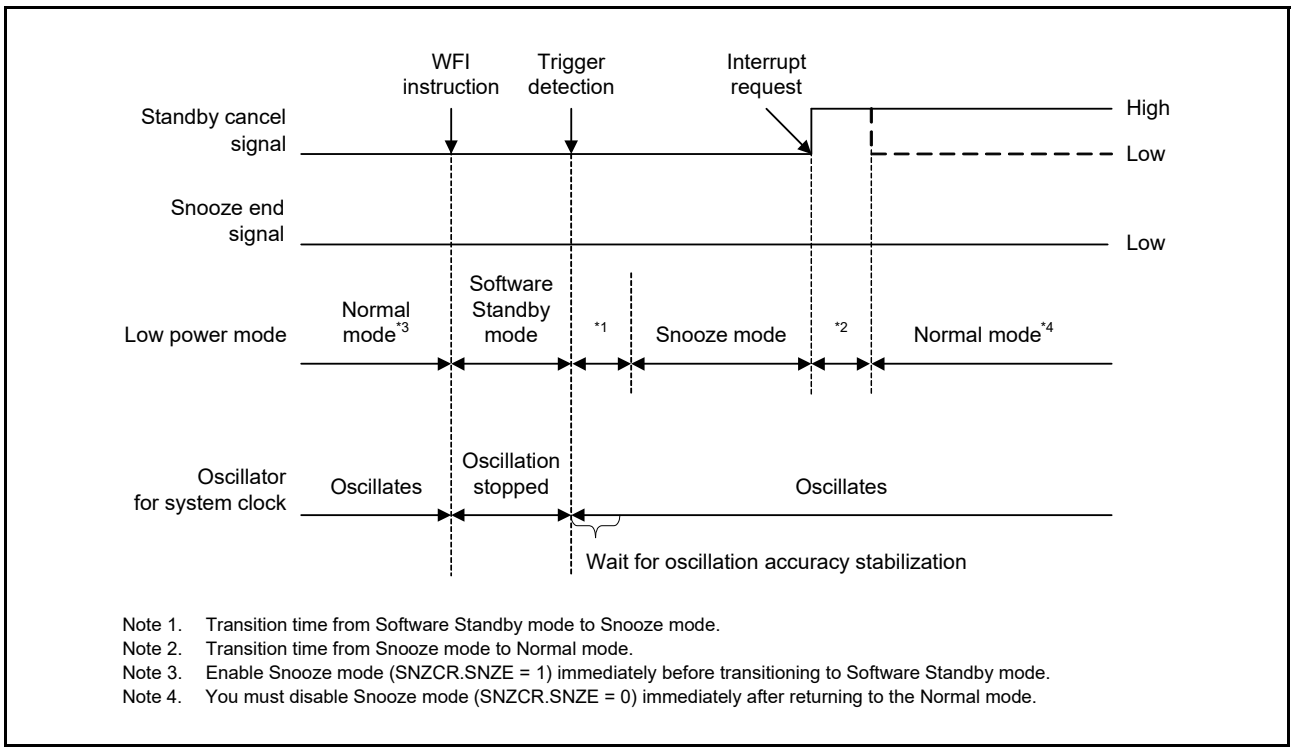
Table 11.6 Available Snooze requests to switch to Snooze mode

	Control Register	
Snooze request	**Register**	**Bit**
PORT_IRQn (n = 0 to 15)	SNZREQCR	SNZREQENn (n = 0 to 15)
KEY_INTKR	SNZREQCR	SNZREQEN17
ACMP_LP0	SNZREQCR	SNZREQEN23
RTC_ALM	SNZREQCR	SNZREQEN24
RTC_PRD	SNZREQCR	SNZREQEN25
AGT1_AGTI	SNZREQCR	SNZREQEN28
AGT1_AGTCMAI	SNZREQCR	SNZREQEN29
AGT1_AGTCMBI	SNZREQCR	SNZREQEN30
RXD0 falling edge	SNZCR	RXDREQEN*1

Note 1. Do not set the RXDREQEN bit to 1 except in asynchronous mode.

11.8.2 Canceling Snooze Mode

Snooze mode is canceled by any interrupt request that is available in Software Standby mode or any reset. Table 11.3 shows the requests that can be used to exit each mode. On exiting the Snooze mode, the MCU transitions to Normal mode and proceeds with exception processing for the given interrupt or reset. An action triggered by the interrupt requests selected in SELSR0 cancels Snooze mode. Interrupt canceling Snooze mode must be selected by IELSRn (n = 0 to 63) to link to the NVIC for the corresponding interrupt handling. See section 14, Interrupt Controller Unit (ICU) for information on setting SELSR0 and IELSRn.



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 transitioning to Software Standby mode.
Note 4. You must disable Snooze mode (SNZCR.SNZE = 0) immediately after returning to the Normal mode.

Figure 11.9 Canceling of Snooze mode when an interrupt request signal is generated

11.8.3 Return to Software Standby Mode

Table 11.7 shows the Snooze end request that can be used as a trigger 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 one of the requests invokes transition to Software Standby mode from Snooze mode.

Table 11.8 shows the Snooze end conditions that consist of the Snooze end requests and the conditions of the peripheral modules. The CTSU, SCI0, ADC140, and DTC can keep the MCU in Snooze mode until they complete operation. However, an AGT1 underflow as a trigger to return to Software Standby mode cancels Snooze mode without waiting for the completion of SCI0 operation.

Figure 11.10 shows the timing chart for the transition from Snooze mode to Software Standby mode. The mode transition occurs depending on which Snooze end requests are set in the SNZEDCR register. A Snooze request is cleared automatically after the MCU operation returns to Software Standby mode.

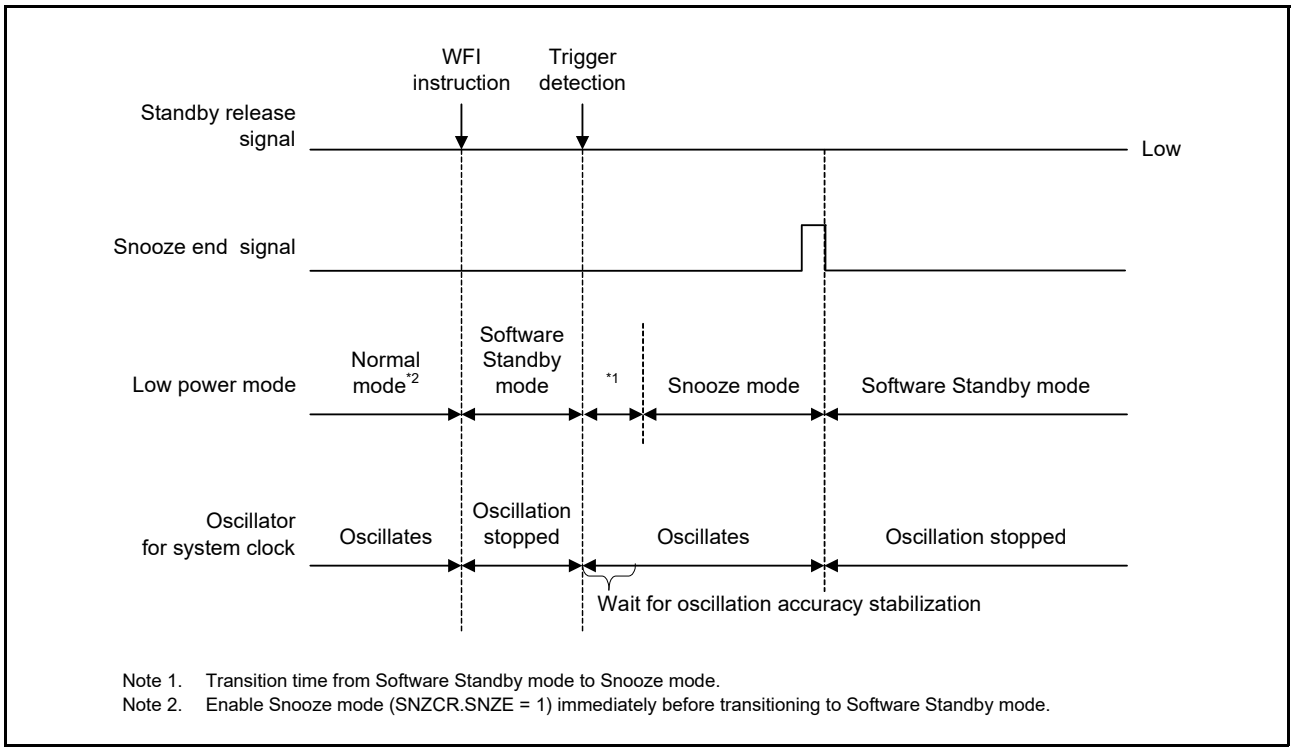
Table 11.7 Available snooze end requests (triggers to return to Software Standby mode)

Snooze end request	Enable/disable control	
	Register	Bit
AGT1 underflow or measurement complete (AGT1_AGTI)	SNZEDCR	b0
DTC transfer completion (DTC_COMPLETE)	SNZEDCR	b1
Not DTC transfer completion (DTC_TRANSFER)	SNZEDCR	b2
ADC140 window A/B compare match (ADC140_WCMPM)	SNZEDCR	b3
ADC140 window A/B compare mismatch (ADC140_WCMPUM)	SNZEDCR	b4
SCI0 address mismatch (SCI0_DCUF)	SNZEDCR	b7

Table 11.8 Snooze end conditions

Module operating when a snooze end request occurs	Snooze end request	
	AGT1 underflow	Other than AGT1 underflow
DTC	The MCU transitions to the Software Standby mode after all of these modules complete operation.	The MCU transfers to the Software Standby mode after all of the modules listed to the left complete operation.
ADC140		
CTSU		
SCI0	The MCU transitions to the Software Standby mode immediately after a Snooze end request is generated.	
Other than above	The MCU transitions to the Software Standby mode immediately after a Snooze end request is generated.	

Note: If the DTC is used to activate the ADC140, CTSU, or SCI, 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 transitioning to Software Standby mode.

Figure 11.10 Canceling of Snooze mode when interrupt request signal is not generated

11.8.4 Snooze Operation Example

Figure 11.11 shows an example setting for using ELC in Snooze mode.

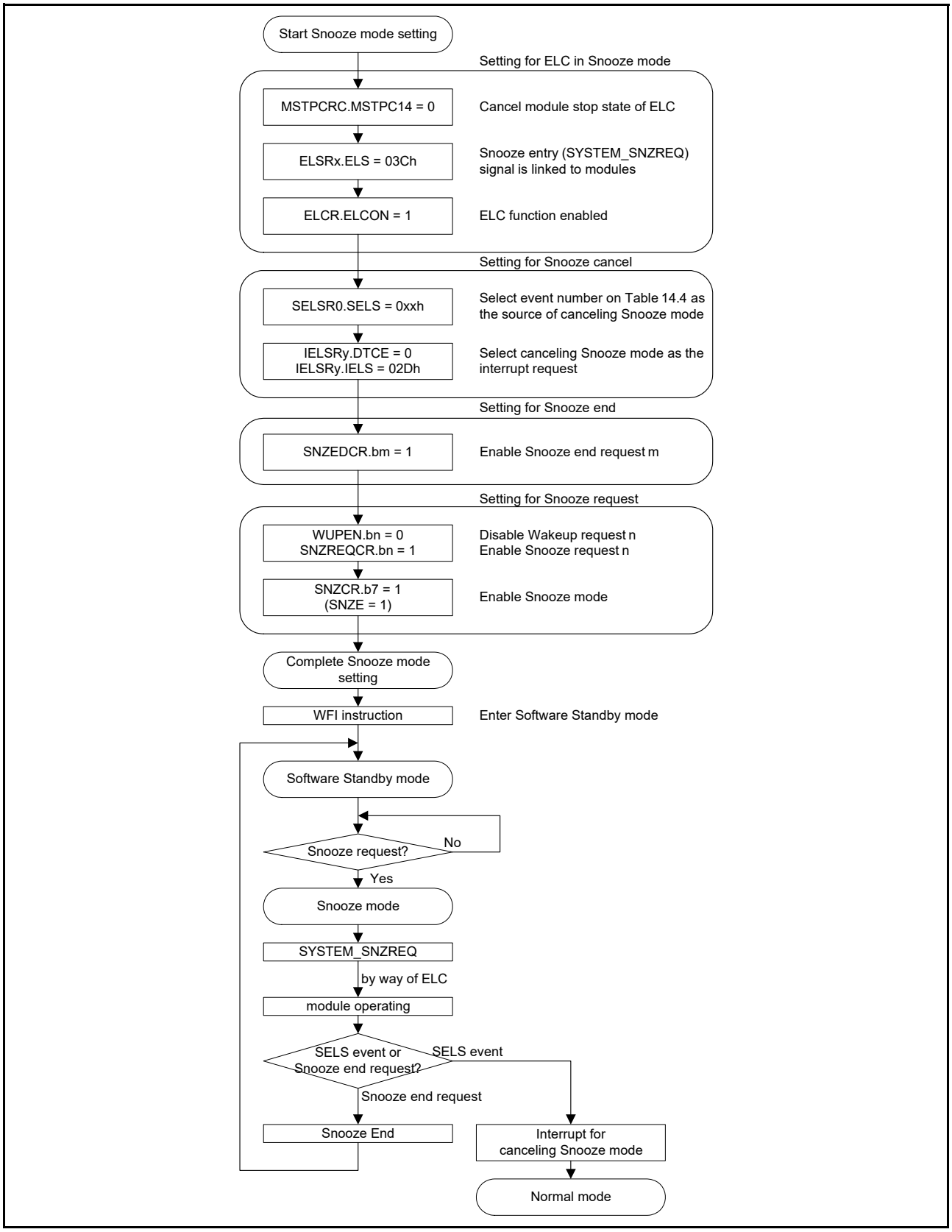


Figure 11.11 Example setting for using ELC in Snooze mode

The MCU is capable of data transmission/reception in SCI0 asynchronous mode without CPU intervention. Table 11.9 and Table 11.10 show the maximum transfer rate of the SCI0 in Snooze mode. When using the SCI0 in Snooze mode, use one of the following operating modes: High-speed mode, Middle-speed mode, or Low-speed mode. Do not use Low-voltage mode or Subosc-speed mode.

Table 11.9 and Table 11.10 show the max transfer rate of SCI0 in Snooze mode. When SCI0 is used in the Snooze mode, the following settings must be used: BGDM = 0, ABCS = 0, ABSCE = 0. See section 29, Serial Communications Interface (SCI) for information on these bits.

High-speed mode, Middle-speed mode, Low-speed mode

Table 11.9 HOCO: ± 1.0% (Ta = -20 to 85°C) **(Unit: bps)**

Maximum division ratio of ICLK, PCLKA, PCLKB, PCLKC, PCLKD, FCLK, BCLK, and TRCLK	HOCO frequency			
	24 MHz	32 MHz	48 MHz	64 MHz
1	9600*1	-	-	-
2	9600*2	9600*4	4800	-
4	9600*3	9600*5	4800	2400
8	4800	4800	4800	2400
16	4800	4800	4800	2400
32	2400	2400	2400	2400
64	2400	2400	2400	2400

Note 1. SCI0.SMR.CKS[1:0]=00b, SCI0.SEMR.BRME=1, SCI0.BRR=3Dh, SCI0.MDDR=CEh must be used for 9600bps.
Note 2. SCI0.SMR.CKS[1:0]=00b, SCI0.SEMR.BRME=1, SCI0.BRR=1Eh, SCI0.MDDR=CEh must be used for 9600bps.
Note 3. SCI0.SMR.CKS[1:0]=00b, SCI0.SEMR.BRME=1, SCI0.BRR=0Dh, SCI0.MDDR=BAh must be used for 9600bps.
Note 4. SCI0.SMR.CKS[1:0]=00b, SCI0.SEMR.BRME=1, SCI0.BRR=32h, SCI0.MDDR=FEh must be used for 9600bps.
Note 5. SCI0.SMR.CKS[1:0]=00b, SCI0.SEMR.BRME=1, SCI0.BRR=18h, SCI0.MDDR=F9h must be used for 9600bps.

High-speed mode, Middle-speed mode, Low-speed mode

Table 11.10 HOCO: ± 2.0% (Ta = -40 to -20°C, 85 to 105°C) **(Unit: bps)**

Maximum division ratio of ICLK, PCLKA, PCLKB, PCLKC, PCLKD, FCLK, BCLK, and TRCLK	HOCO frequency			
	24 MHz	32 MHz	48 MHz	64 MHz
1	2400	-	-	-
2	2400	2400	2400	-
4	2400	2400	2400	1200
8	2400	2400	2400	1200
16	2400	2400	2400	1200
32	1200	1200	1200	1200
64	1200	1200	1200	1200

Figure 11.12 shows an example setting for using the SCI0 in Snooze mode entry.

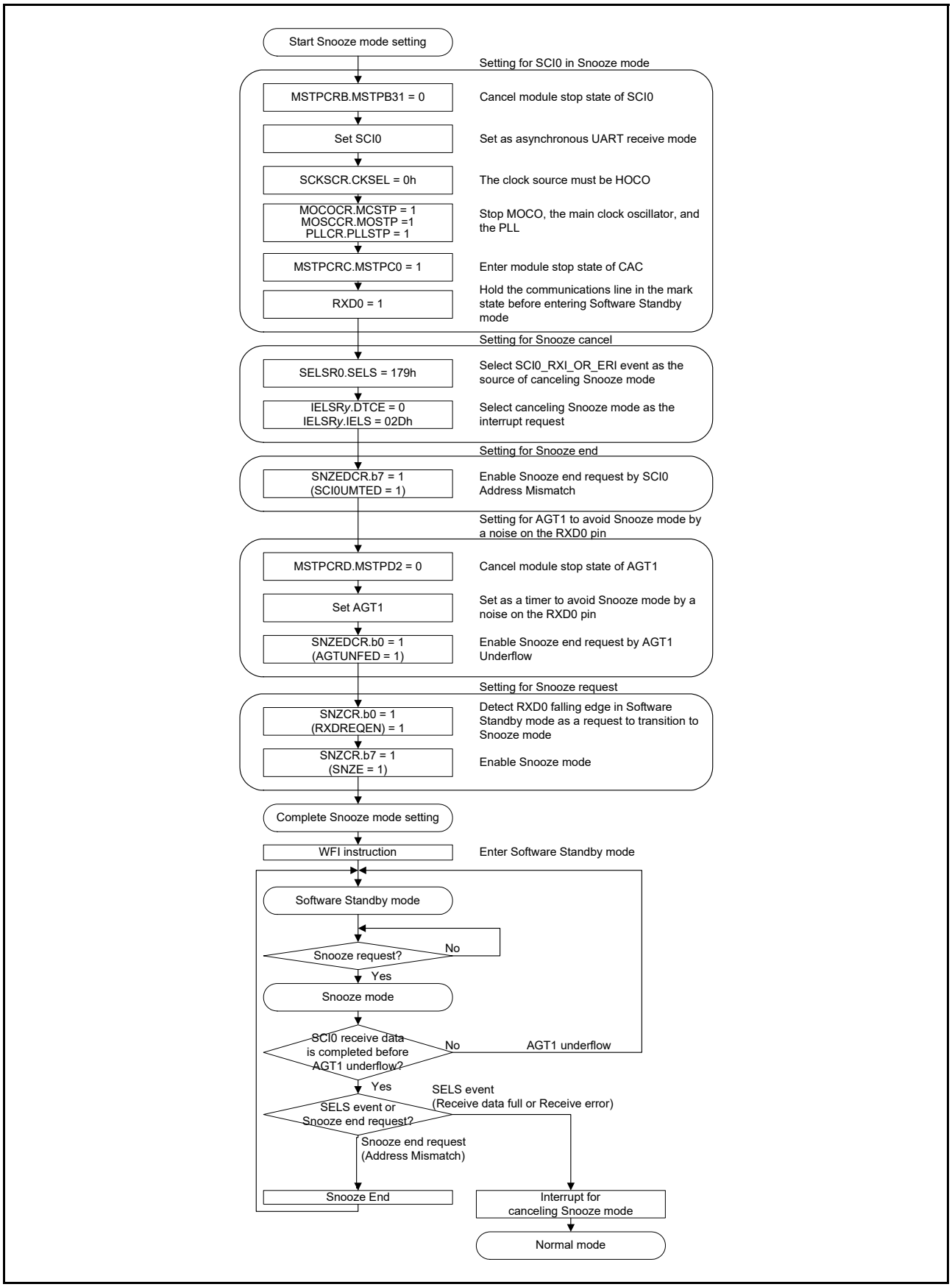


Figure 11.12 Example setting for using SCI0 in Snooze mode entry

11.9 Usage Notes

11.9.1 Register Access

(1) Invalid register write accesses during specific modes or transitions

Do not write to registers listed in this section under any of the listed conditions.

[Registers]

- All registers with a peripheral name of "SYSTEM".

[Conditions]

- When OPCCR.OPCMTSF = 1 or SOPCCR.SOPCMTSF = 1 (during transition of the operating power control mode)
- During time period from executing a WFI instruction to returning to Normal mode
- When FENTRYR.FENTRY0 = 1 or FENTRYR.FENTRYD = 1 (flash P/E mode, data flash P/E mode)
- When FLSTOP.FLSTPF = 1 (during transition).

(2) Valid settings for the clock-related registers

Table 11.11 and Table 11.12 show the valid setting of the clock-related registers in each operating power control mode. Do not write any value other than the valid setting. Any other written value is ignored. Each register has certain prohibited settings under conditions other than those related to the operating power control modes. See section 9, Clock Generation Circuit for these other conditions for each register.

Table 11.11 Valid settings for the clock-related registers (1)

Mode	Valid setting								
	SCKSCR.CKSEL[2:0] CKOCR.CKOSEL[2:0]	SCKDIVCR.FCK[2:0] ICK[2:0]	SLCDSCKCR.LCDSCKSEL[2:0]	PLLCR.PLLSTP	HOCOCR.HCSTP	MOCOCR.MCSTP	LOCOCR.LCSTP	MOSCCR.MOSTP	SOSCCR.SOSTP
High-speed, Middle-speed	000b (HOCO) 001b (MOCO) 010b (LOCO) 011b (MOSC) 100b (SOSC) 101b (PLL)*1	000b (1/1) 001b (1/2) 010b (1/4) 011b (1/8) 100b (1/16) 101b (1/32) 110b (1/64)	000b (LOCO) 001b (SOSC) 010b (MOSC) 100b (HOCO)	0 (operating) 1 (stop)	0 (operating) 1 (stop)	0 (operating) 1 (stop)	0 (operating) 1 (stop)	0 (operating) 1 (stop)	0 (operating) 1 (stop)
Low-speed, Low-voltage	000b (HOCO) 001b (MOCO) 010b (LOCO) 011b (MOSC) 100b (SOSC)			1 (stop)					
Subosc-speed	010b (LOCO) 100b (SOSC)	000b (1/1)	000b (LOCO) 001b (SOSC)	1 (stop)	1 (stop)	1 (stop)	0 (operating) 1 (stop)	1 (stop)	0(operating) 1 (stop)

Note 1. SCKSCR.CKSEL[2:0] only.

Table 11.12 Valid setting for the clock-related registers (2)

Operating oscillator	Valid setting	
	SOPCCR.SOPCM	OPCCR.OPCM[1:0]
PLL	0	00b, 01b
High-speed on-chip oscillator	0	00b, 01b, 10b, 11b
Middle-speed on-chip oscillator		
Main clock oscillator		
Low-speed on-chip oscillator	0, 1	00b, 01b, 10b, 11b
Sub-clock oscillator		
IWDT-dedicated on-chip oscillator		

(3) Invalid register write access in Subosc-speed mode

Do not write to registers listed in this section under the listed condition.

[Registers]

- SCKSCR, OPCCR.

[Condition]

- SOPCCR.SOPCM = 1 (Subosc-speed mode).

(4) Invalid register write accesses by DTC or DMAC

Do not write to registers listed in this section by DTC or DMAC.

[Registers]

- MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD.

(5) Invalid register write accesses in Snooze mode

Do not write to registers listed in this section in Snooze mode. They must be set before entering Software Standby mode.

[Registers]

- SNZCR, SNZEDCR, SNZREQCR.

(6) Invalid write access to FLSTOP.FLSTOP

Do not write the FLSTOP.FLSTOP bit to 1 under the listed conditions.

[Conditions]

- SOPCCR.SOPCM = 0, OPCCR.OPCM[1:0] = 00b (High-speed mode)
- SOPCCR.SOPCM = 0, OPCCR.OPCM[1:0] = 01b (Middle-speed mode)
- SOPCCR.SOPCM = 0, OPCCR.OPCM[1:0] = 11b (Low-speed mode)
- SOPCCR.SOPCM = 1 (Subosc-speed mode).

(7) Invalid write access to MEMWAIT.MEMWAIT

Do not set the MEMWAIT.MEMWAIT bit to 1 under the listed conditions.

[Conditions]

- SOPCCR.SOPCM = 0, OPCCR.OPCM[1:0] = 01 (Middle-speed mode)
- SOPCCR.SOPCM = 0, OPCCR.OPCM[1:0] = 10 (Low-voltage mode)
- SOPCCR.SOPCM = 0, OPCCR.OPCM[1:0] = 11 (Low-speed mode)
- SOPCCR.SOPCM = 1 (Subosc-speed mode).

(8) Invalid write access when PRCR.PRC1 is 0

Do not write to the registers in this section when the PRCR.PRC1 bit is 0.

[Registers]

- SBYCR, SNZCR, SNZEDCR, SNZREQCR, FLSTOP, PSMCR, OPCCR, SOPCCR.

11.9.2 I/O Port States

The I/O port states in Software Standby mode and Snooze mode, except when modifying in Snooze mode, are the same before entering the modes. Therefore, the power consumption is not reduced while the output signals are held high.

11.9.3 Module-Stop State of DMAC and DTC

Before writing 1 to MSTPCRA.MSTPA22, clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0. For details, see section 17, DMA Controller (DMAC) and section 18, Data Transfer Controller (DTC).

11.9.4 Internal Interrupt Sources

Interrupts do not operate in the module-stop state. If the module-stop bit is set while an interrupt request is generated, a CPU interrupt source, or a DMAC or DTC startup source cannot be cleared. Always disable the corresponding interrupts before setting the module-stop bits.

11.9.5 Transition to Low Power Modes

Because the MCU does not support wakeup by event, do not enter low power modes (Sleep mode or Software Standby mode) by executing a WFE instruction. Also, do not set the SLEEPDEEP bit of the System Control Register in the Cortex®-M4 core because the MCU does not support low power modes by SLEEPDEEP.

11.9.6 Timing of WFI Instruction

It is possible for the WFI instruction to be executed before I/O register and CS area writes are complete, in which case operation might not proceed as intended. This can happen if the WFI is placed immediately after a write to an I/O register or CS area. To avoid this problem, read back the register and CS area that was written to confirm that the write completed. For example, reading the MSTPCRB register before execution of the WFI instruction can secure the period to complete writing to the I/O register.

11.9.7 Writing WDT/IWDT Registers by DMAC or DTC in Sleep Mode or Snooze Mode

Do not write to the WDT or IWDT registers by the DMAC or DTC while WDT or IWDT is stopped after entering Sleep mode or Snooze mode.

11.9.8 Oscillators in Snooze Mode

Oscillators that stop on entering Software Standby mode automatically restart when a trigger for transitioning to Snooze mode is generated. The MCU does not enter Snooze mode until all of 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.

11.9.9 Snooze Mode Entry by RXD0 Falling Edge

When the SNZCR.RXDREQEN bit is 1, noise on the RXD0 pin might cause the MCU to transition from Software Standby mode to Snooze mode. Any subsequent RXD0 data can be received in Snooze mode by noise on the RXD0 pin. If the MCU does not receive RXD0 data after the noise, an interrupt such as SCI0_ERI or SCI0_RXI, or an address non-match event is not generated, and the MCU stays in Snooze mode. To avoid this, an AGT1 underflow interrupt must be used to return to Software Standby mode or Normal mode when using SCI0 in Snooze mode. However, do not use the AGT1 underflow as a source to return to Software Standby mode during an SCI communication. This causes the SCI0 to stop the operation in a half-finished state.

11.9.10 Using SCI0 in Snooze Mode

When using SCI0 in Snooze mode, a wakeup request other than an AGT1 underflow must not be used.

When using SCI0 in Snooze mode, the following conditions must be satisfied:

- The clock source must be HOCO
- MOCO, the main clock oscillator, and the PLL must stop before entering Software Standby mode
- The RXD0 pin must be kept at high level before entering Software Standby mode
- A transition to Software Standby mode must not occur during an SCI communication
- The MSTPCRC.MSTPC0 bit must be 1 before entering Software Standby mode.

11.9.11 Conditions of A/D Conversion Start in Snooze Mode

The ADC14 can be triggered only by the ELC in Snooze mode. Do not use a software trigger or the ADTRG0 pin.

11.9.12 Conditions of CTSU in Snooze Mode

The CTSU can only be started by the ELC in Snooze mode.

11.9.13 ELC Event in Snooze Mode

The ELC events available in Snooze mode are listed in this section. 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)
- ADC140 window A/B compare match (ADC140_WCMPM)
- ADC140 window A/B compare mismatch (ADC140_WCMPUM)
- Data operation circuit interrupt (DOC_DOPCI).

11.9.14 Module-Stop Function for ADC140

When entering the Software Standby mode, it is recommended to set the ADC140 module-stop state to reduce power consumption. In this case, the ADC140 can be available in the Snooze mode with releasing the ADC140 module-stop using the DTC. Similarly, set the module-stop using the DTC before returning to the Software Standby mode from the Snooze mode.

11.9.15 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 could increase to a value greater than the specified value (as provided in the User's Manual), by up to 600 µA. To avoid this, initialize the unused circuit using the steps shown in Figure 11.13, Initial setting flow example for an unused circuit.

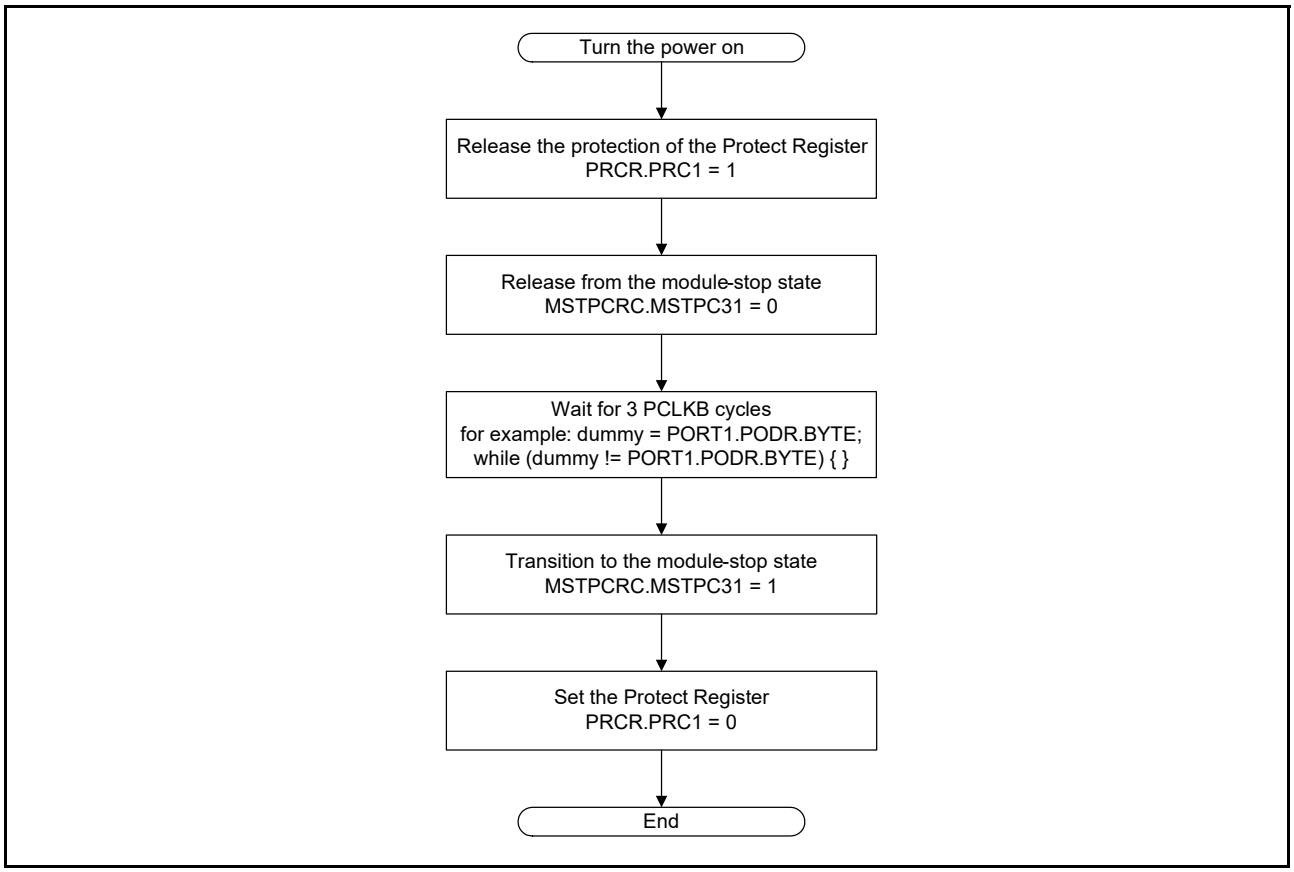


Figure 11.13 Initial setting flow example for an unused circuit

12. Battery Backup Function

12.1 Overview

The MCU provides a battery backup function that maintains partial battery powering in the event of a power loss. Switching between VCC and VBATT, battery backup power is provided for the RTC, AGT, SOSC, LOCO, Wakeup Control/Backup Memory, and VBATT_R Low Voltage Detection.

During normal operation, the battery-powered area is powered by the main power supply, the VCC pin. When a VCC voltage drop is detected, the power source switches to the dedicated battery backup power pin, the VBATT pin. When the voltage rises again, the power source switches back from VBATT to VCC. Table 12.1 lists the VBATT wakeup I/O pin configuration.

Table 12.1 VBATT wakeup I/O pin configuration

Pin Name	I/O	Function
VBATWIOn	Input/Output	Output wakeup signal for the VBATT Wakeup Control function. External event input for the VBATT Wakeup Control function.

Note: n = 0 to 2.

12.1.1 Features of Battery Backup Function

The features include:

- Battery power supply switch
- VBATT pin low voltage detection
- VBATT_R low voltage detection
- Backup registers
- VBATT wakeup control function
- Time capture pin detection.

12.1.2 Battery Power Supply Switch

When the voltage applied to the VCC pin drops, this feature switches the power supply from the VCC pin to the VBATT pin. When the voltage rises, it switches the power supply from the VBATT pin back to the VCC pin. The switch is controlled by the VBTCR1.BPWSWSTP bit. By default, switching is enabled and can be disabled by setting the VBTCR1.BPWSWSTP bit to 1.

12.1.3 VBATT Pin Low Voltage Detection

The VBATT low voltage detection function supports the battery-powered area. This function monitors whether power is supplied to the VBATT pin. The low voltage condition of the power supply can be detected using a flag provided in the VBATT Status Register.

12.1.4 VBATT_R Low Voltage Detection

VBATT_R low voltage detection function supports the battery-powered area. VBATT_R is the output voltage of the battery power supply switch. This function monitors the VBATT_R voltage level. A low voltage detection causes VBATT_POR reset and initializes the battery-powered area. See details in the description for each register. The VBATT Status Register includes a flag to check for this low voltage detection.

12.1.5 Backup Registers

The battery-powered area provides 512 one-byte backup registers. These registers retain data only when VBATT is supplied and VCC is powered off. This memory is checked by the VBATT pin low voltage detection.

12.1.6 VBATT Wakeup Control Function

The VBATT wakeup control function can toggle the VBATWIO[2:0] pins when the RTC alarm, periodic signal, AGT1 underflow signal, or VBATWIOn (n = 0 to 2) input signal is asserted, when VBATT_R is powered by the VBATT pin.

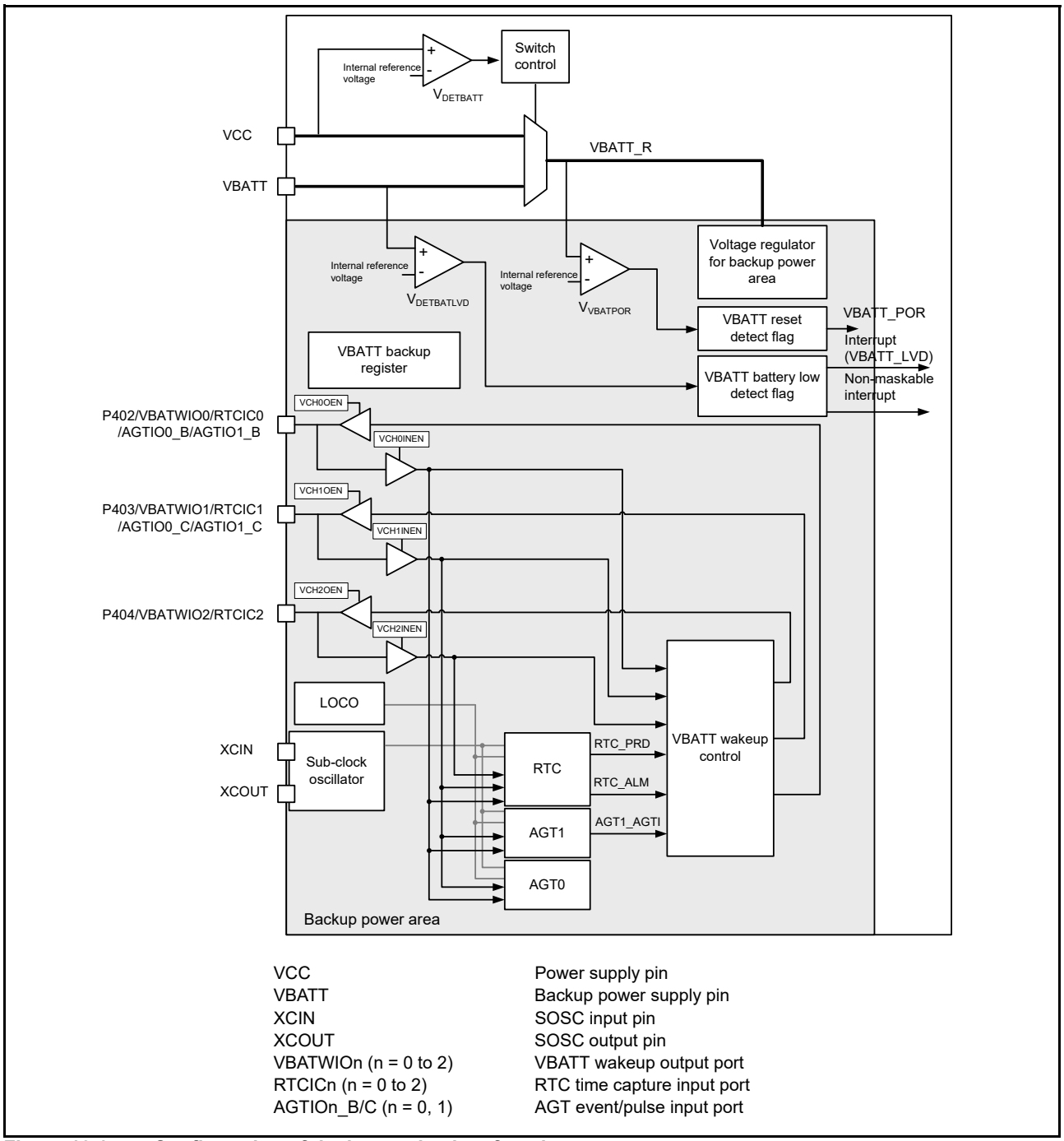
Note: The toggle triggered in the wakeup control function does not generate an interrupt at the ICU or a reset to the reset module. For example, the output toggle triggers other devices on board to control the VCC power supply. For details, see section 12.3.5, VBATT Wakeup Control Function Usage.

12.1.7 Time Capture Pin Detection

The RTC detects input level changes on the time capture pins, RTCICn (n = 0 to 2).
For details on the RTCICn pins, see section 25, Realtime Clock (RTC). To use RTCICn pins, set the VBTICTLR register as described in section 12.2, Register Descriptions.

Note: When the battery backup function is not used, the VBATT pin must be connected to the VCC pin.
Note: When the VBATT function is not used, power is not supplied to RTC, SOSC (including multiplexed port), AGT, or LOCO before setting the VBTCR1.BPWSWSTP bit to 1. After setting the VBTCR1.BPWSWSTP bit, a VBATT_POR reset time, tVBATPOR, as described in section 52, Electrical Characteristics, is required before supplying power to the modules. The VBTCR1.BPWSWSTP bit must be set to 1 after a power-on reset, regardless of whether the VBATT function is used. See section 12.2.1, VBATT Control Register 1 (VBTCR1) for details.

Figure 12.1 shows the configuration of the battery backup function.



VCC	Power supply pin
VBATT	Backup power supply pin
XCIN	SOSC input pin
XCOUT	SOSC output pin
VBATWIOn (n = 0 to 2)	VBATT wakeup output port
RTCICn (n = 0 to 2)	RTC time capture input port
AGTIOn_B/C (n = 0, 1)	AGT event/pulse input port

Figure 12.1 Configuration of the battery backup function

12.2 Register Descriptions

12.2.1 VBATT Control Register 1 (VBTCR1)

Address(es): SYSTEM.VBTCR1 4001 E41Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	BPWSWSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	BPWSWSTP	Battery Power Supply Switch Stop	0: Enable battery power supply switch 1: Stop battery power supply switch.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

BPWSWSTP bit (Battery Power Supply Switch Stop)

The BPWSWSTP bit can enable the battery power supply switch to change the battery backup module supply voltage from VCC to VBATT when the voltage applied to the VCC pin drops. When the battery power supply switch is stopped, the battery backup module power supply is always from VCC. To disable the battery backup function, write 1 to this bit. This bit is initialized only on power-on reset.

Note: This bit can be set without checking the VBATSR.VBTRVLD bit status.

Note: The VBTCR1.BPWSWSTP bit must be set to 1 after a power-on reset, regardless of whether the VBATT function is used. The setting flow of the VBTCR1.BPWSWSTP bit is shown in Figure 12.2. Also, the VBTCR1.BPWSWSTP bit must be cleared after the other related registers are set when the VBATT function is used.

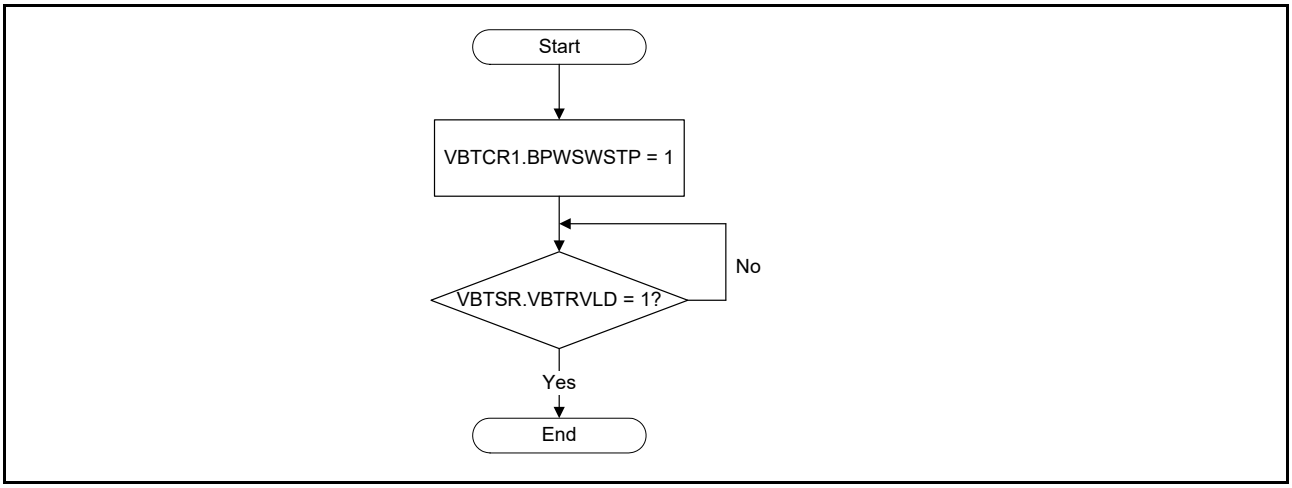


Figure 12.2 Setting flow of the VBTCR1.BPWSWSTP bit

Note: In Figure 12.2, if the VBTSR.VBTRVLD bit is not 1, it takes the VBATT_POR reset time, tVBATPOR, as described in section 52, Electrical Characteristics, to exit the loop.
When VBTSR.VBTRVLD bit is 0, the following registers cannot be accessed. Other registers can be accessed regardless of this condition:

- LOCOCR, LOCOUTCR, SOSCCR, and SOMCR described in section 9, Clock Generation Circuit.
- All registers described in this section except for VBTCR1 and the VBTSR.VBTRVLD bit.
- All registers described in section 24, Asynchronous General Purpose Timer (AGT).
- All registers described in section 25, Realtime Clock (RTC).

12.2.2 VBATT Control Register 2 (VBTCR2)

Address(es): SYSTEM.VBTCR2 4001 E4B0h

	b7	b6	b5	b4	b3	b2	b1	b0
	VBTLVDLVL[1:0]		—	VBTLVDEN	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	VBTLVDEN	VBATT Pin Low Voltage Detect Enable	0: Disable VBATT pin low voltage detection 1: Enable VBATT pin low voltage detection.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	VBTLVDLVL[1:0]	VBATT Pin Low Voltage Detect Level Select	b7 b6 0 0: Reserved 0 1: Setting prohibited 1 0: 2.3 V 1 1: 2.1 V.	R/W

The VBTCR2 register controls the VBATT pin low voltage detection function. VBTCR2 is reset by the VBATT_POR signal.

VBTLVDEN bit (VBATT Pin Low Voltage Detect Enable)

The VBTLVDEN bit controls the VBATT pin low voltage detection.

VBTLVDLVL[1:0] bits (VBATT Pin Low Voltage Detect Level Select)

The VBTLVDLVL[1:0] bits select the VBATT_R low voltage detection level.

12.2.3 VBATT Status Register (VBTSR)

Address(es): SYSTEM.VBTSR 4001 E4B1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	VBTRVLD	—	—	VBTBLDF	VBTRDF
Value after reset:	0	0	0	0*5	0	0	0*1	1*2

Bit	Symbol	Bit name	Description	R/W
b0	VBTRDF	VBATT_R Reset Detect Flag	0: VBATT_R voltage power-on reset not detected 1: VBATT_R selected voltage power-on reset detected.	R/(W)*3
b1	VBTBLDF	VBATT Battery Low Detect Flag*4	0: VBATT pin low voltage not detected 1: VBATT pin low voltage detected.	R/(W)*3
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	VBTRVLD	VBATT_R Valid	0: VBATT_R area not valid 1: VBATT_R area valid.	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This flag is only reset by the VBATT_POR reset.
Note 2. This flag is only set by the VBATT_POR reset.
Note 3. Only 0 can be written after reading 1.
Note 4. This flag is only valid when VBTLVDEN is 1. If VBTLVDEN is 0, this flag is read as 0.
Note 5. Depends on the VBATT_R voltage level.

VBTRDF flag (VBATT_R Reset Detect Flag)

The VBTRDF flag indicates that a VBATT_R (selected voltage of VCC or VBATT) power-on reset occurs.

[Setting condition]

- When a VBATT_R voltage power-on reset occurs.

[Clearing condition]

- When 1 is read from and then 0 is written to VBTRDF.

VBTBLDF flag (VBATT Battery Low Detect Flag)

The VBTBLDF flag indicates that a VBATT battery low voltage detection occurs.

[Setting condition]

- When VBATT battery low voltage detection occurs.

[Clearing condition]

- When 1 is read from and then 0 is written to VBTBLDF.

VBTRVLD bit (VBATT_R Valid)

The VBTRVLD checks whether the VBATT_R area is valid.

Confirm that VBTRVLD bit is 1 before writing to or reading from the following registers:

- LOCOCR, LOCOUTCR, SOSCCR, and SOMCR described in section 9, Clock Generation Circuit
- All registers described in this section except for VBTCR1 and the VBTSR.VBTRVLD bit
- All registers described in section 24, Asynchronous General Purpose Timer (AGT)
- All registers described in section 25, Realtime Clock (RTC).

12.2.4 VBATT Comparator Control register (VBTCMPCR)

Address(es): SYSTEM.VBTCMPCR 4001 E4B2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	VBTCMPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	VBTCMPE	VBATT Pin Low Voltage Detect Circuit Output Enable	0: Disable VBATT pin low voltage detect circuit output 1: Enable VBATT pin low voltage detect circuit output.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

VBTCMPE bit (VBATT Pin Low Voltage Detect Circuit Output Enable)

The VBTCMPE controls the VBATT pin low voltage detection circuit output. This bit is initialized by the VBATT_POR signal.

12.2.5 VBATT Pin Low Voltage Detect Interrupt Control Register (VBTLVDICR)

Address(es): SYSTEM.VBTLVDICR 4001 E4B4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	VBTLVDISEL	VBTLVDIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	VBTLVDIE	VBATT Pin Low Voltage Detect Interrupt Enable	0: Disable VBATT pin low voltage detection interrupt 1: Enable VBATT pin low voltage detection interrupt.	R/W
b1	VBTLVDISEL	Pin Low Voltage Detect Interrupt Select	0: Non-maskable Interrupt 1: Maskable Interrupt.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The VBTLVDICR is reset by the VBATT_POR signal.

12.2.6 VBATT Backup Register (VBTBKR[n]) (n = 0 to 511)

Address(es): SYSTEM.VBTBKR[0] 4001 E500h to SYSTEM.VBTBKR[511] 4001 E6FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	VBTBKR[0:7]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

VBTKBKRn is an 8-bit access read/write register to store data powered by VBATT. The value of this register is retained even when VCC is not powered but VBATT is powered. This register is not initialized by any reset.

Note: When accessing the VBATT backup registers, the VCC level must be over $V_{_BKBATT}$ as described in section 52, Electrical Characteristics.

Note: The system using VBATT backup registers and Software Standby mode must constantly activate LOCO.

12.2.7 VBATT Wakeup Control Register (VBTWCTLR)

Address(es): SYSTEM.VBTWCTLR 4001 E4B6h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	VWEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	VWEN	VBATT Wakeup Enable	0: Disable wakeup function 1: Enable wakeup function.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The VBTWCTLR controls the VBATT Wakeup Control Register. VBTWCTLR is reset by the VBATT_POR signal.

VWEN bit (VBATT Wakeup Enable)

The VWEN bit enables the VBATT wakeup control function. When the VWEN bit is set to 0 and the VBTOCTLR.VCHnOEN (n = 0 to 2) bit is set to 1, the VBATWIOn (n = 0 to 2) pin output is low level. When the VWEN bit is set to 1, the output from VBATWIOn pin changes to the level specified by the VBTOCTLR.VOUTnLSEL (n = 0 to 2) bit.

Set the VWEN bit to 1 only after setting of the following registers is complete. Set VWEN to 0 first before modifying these registers:

- VBTWCHnOTSR
- VBTICTLR
- VBTOCTLR
- VBTWTER
- VBTWEGR (n = 0 to 2)

12.2.8 VBATT Wakeup I/O 0 Output Trigger Select Register (VBTWCH0OTSR)

Address(es): SYSTEM.VBTWCH0OTSR 4001 E4B8h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CH0VAGTUTE	CH0VRTCATE	CH0VRTCTE	CH0VCH2TE	CH0VCH1TE	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	CH0VCH1TE	VBATWIO0 Output VBATWIO1 Trigger Enable	0: Disable VBATT wakeup I/O 0 output trigger by the VBATWIO1 pin 1: Enable VBATT wakeup I/O 0 output trigger by the VBATWIO1 pin.	R/W
b2	CH0VCH2TE	VBATWIO0 Output VBATWIO2 Trigger Enable	0: Disable VBATT wakeup I/O 0 output trigger by the VBATWIO2 pin 1: Enable VBATT wakeup I/O 0 output trigger by the VBATWIO2 pin.	R/W
b3	CH0VRTCTE	VBATWIO0 Output RTC Periodic Signal Enable	0: Disable VBATT wakeup I/O 0 output trigger by the RTC periodic signal 1: Enable VBATT wakeup I/O 0 output trigger by the RTC periodic signal.	R/W
b4	CH0VRTCATE	VBATWIO0 Output RTC Alarm Signal Enable	0: Disable VBATT wakeup I/O 0 output trigger by the RTC alarm signal 1: Enable VBATT wakeup I/O 0 output trigger by the RTC alarm signal.	R/W
b5	CH0VAGTUTE	VBATWIO0 Output AGT Underflow Signal Enable	0: Disable VBATT wakeup I/O 0 output trigger by the AGT1 underflow signal 1: Enable VBATT wakeup I/O 0 output trigger by the AGT1 underflow signal.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The VBTWCH0OTSR controls the VBATT wakeup I/O 0 output trigger source.

If a bit in this register is set to 1 and the associated wakeup trigger flag in the VBTWFR register is set, the VBATWIO0 pin output signal is based on the VOUT0LSEL bit in the VBTOCTLR register.

The VBTWCH0OTSR register is initialized by the VBATT_POR signal.

12.2.9 VBATT Wakeup I/O 1 Output Trigger Select Register (VBTWCH1OTSR)

Address(es): SYSTEM.VBTWCH1OTSR 4001 E4B9h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CH1VAGTUTE	CH1VRTCATE	CH1VRTCTE	CH1VCH2TE	—	CH1VCH0TE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CH1VCH0TE	VBATWIO1 Output VBATWIO0 Trigger Enable	0: Disable VBATT wakeup I/O 1 output trigger by the VBATWIO0 pin 1: Enable VBATT wakeup I/O 1 output trigger by the VBATWIO0 pin.	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	CH1VCH2TE	VBATWIO1 Output VBATWIO2 Trigger Enable	0: Disable VBATT wakeup I/O 1 output trigger by the VBATWIO2 pin 1: Enable VBATT wakeup I/O 1 output trigger by the VBATWIO2 pin.	R/W
b3	CH1VRTCTE	VBATWIO1 Output RTC Periodic Signal Enable	0: Disable VBATT wakeup I/O 1 output trigger by the RTC periodic signal 1: Enable VBATT wakeup I/O 1 output trigger by the RTC periodic signal.	R/W
b4	CH1VRTCATE	VBATWIO1 Output RTC Alarm Signal Enable	0: Disable VBATT wakeup I/O 1 output trigger by the RTC alarm signal 1: Enable VBATT wakeup I/O 1 output trigger by the RTC alarm signal.	R/W
b5	CH1VAGTUTE	VBATWIO1 Output AGT Underflow Signal Enable	0: Disable VBATT wakeup I/O 1 output trigger by the AGT1 underflow signal 1: Enable VBATT wakeup I/O 1 output trigger by the AGT1 underflow signal.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The VBTWCH1OTSR controls the VBATT wakeup I/O 1 output trigger source.

If a bit in this register is set to 1 and the associated wakeup trigger flag in the VBTWFR register is set, VBATWIO1 pin output signal is based on the VOUT1LSEL bit in the VBTOCTLR register.

The VBTWCH1OTSR register is initialized by the VBATT_POR signal.

12.2.10 VBATT Wakeup I/O 2 Output Trigger Select Register (VBTWCH2OTSR)

Address(es): SYSTEM.VBTWCH2OTSR 4001 E4BAh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CH2VAGTUTE	CH2VRTCATE	CH2VRTCTE	—	CH2VCH1TE	CH2VCH0TE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CH2VCH0TE	VBATWIO2 Output VBATWIO0 Trigger Enable	0: Disable VBATT wakeup I/O 2 output trigger by the VBATWIO0 pin 1: Enable VBATT wakeup I/O 2 output trigger by the VBATWIO0 pin.	R/W
b1	CH2VCH1TE	VBATWIO2 Output VBATWIO1 Trigger Enable	0: Disable VBATT wakeup I/O 2 output trigger by the VBATWIO1 pin 1: Enable VBATT wakeup I/O 2 output trigger by the VBATWIO1 pin.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	CH2VRTCTE	VBATWIO2 Output RTC Periodic Signal Enable	0: Disable VBATT wakeup I/O 2 output trigger by the RTC periodic signal 1: Enable VBATT wakeup I/O 2 output trigger by the RTC periodic signal.	R/W
b4	CH2VRTCATE	VBATWIO2 Output RTC Alarm Signal Enable	0: Disable VBATT wakeup I/O 2 output trigger by the RTC alarm signal 1: Enable VBATT wakeup I/O 2 output trigger by the RTC alarm signal.	R/W
b5	CH2VAGTUTE	VBATWIO2 Output AGT Underflow Signal Enable	0: Disable VBATT wakeup I/O 2 output trigger by the AGT1 underflow signal 1: Enable VBATT wakeup I/O 2 output trigger by the AGT1 underflow signal.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The VBTWCH2OTSR controls the VBATT wakeup I/O 2 output trigger source.

When a bit in this register is set to 1 and the associated wakeup trigger flag in the VBTWFR register is set, VBATWIO2 pin output signal is based on the VOUT2LSEL bit in the VBTOCTLR register.

The VBTWCH2OTSR register is initialized by the VBATT_POR signal.

12.2.11 VBATT Input Control Register (VBTICTLR)

Address(es): SYSTEM.VBTICTLR 4001 E4BBh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	VCH2INEN	VCH1INEN	VCH0INEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	VCH0INEN	VBATT Wakeup I/O 0 Input Enable	0: Disable VBATWIO0, RTCIC0, AGTIO0_B, and AGTIO1_B inputs 1: Enable VBATWIO0, RTCIC0, AGTIO0_B, and AGTIO1_B inputs.	R/W
b1	VCH1INEN	VBATT Wakeup I/O 1 Input Enable	0: Disable VBATWIO1, RTCIC1, AGTIO0_C, and AGTIO1_C inputs 1: Enable VBATWIO1, RTCIC1, AGTIO0_C, and AGTIO1_C inputs.	R/W
b2	VCH2INEN	VBATT Wakeup I/O 2 Input Enable	0: Disable VBATWIO2 and RTCIC2 inputs 1: Enable VBATWIO2 and RTCIC2 inputs.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The VBTICTLR enables or disables the VBATT wakeup I/O pins. VBTICTLR is reset by the VBATT_POR signal.

VCHnINEN bit (VBATT Wakeup I/O n Input Enable Bit) (n = 0 to 2)

The VCHnINEN bit enables or disables the VBATT wakeup I/O pin inputs.

When using the VBATT wakeup control function along with the external event input function of AGT(AGTIO0_B, AGTIO0_C, AGTIO1_B, and AGTIO1_C), and the time capture function of RTC (RTCICn (n = 0 to 2)), set the VBTICTLR register. To enable the external event input of AGT, also set the AGTn.AGTIOSEL.TIES bit to 1. For information on these functions, see section 24, Asynchronous General Purpose Timer (AGT) and section 25, Realtime Clock (RTC).

12.2.12 VBATT Output Control Register (VBTOCTLR)

Address(es): SYSTEM.VBTOCTLR 4001 E4BCh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	VOUT2 LSEL	VOUT1 LSEL	VOUT0 LSEL	VCH2O EN	VCH1O EN	VCH0O EN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	VCH0OEN	VBATT Wakeup I/O 0 Output Enable	0: Disable VBATWIO0 output 1: Enable VBATWIO0 output.*1, *2	R/W
b1	VCH1OEN	VBATT Wakeup I/O 1 Output Enable	0: Disable VBATWIO1 output 1: Enable VBATWIO1 output.*1, *2	R/W
b2	VCH2OEN	VBATT Wakeup I/O 2 Output Enable	0: Disable VBATWIO2 output 1: Enable VBATWIO2 output.*1, *2	R/W
b3	VOUT0LSEL	VBATT Wakeup I/O 0 Output Level Selection	0: Output L before VBATT wakeup trigger 1: Output H before VBATT wakeup trigger.	R/W
b4	VOUT1LSEL	VBATT Wakeup I/O 1 Output Level Selection	0: Output L before VBATT wakeup trigger 1: Output H before VBATT wakeup trigger.	R/W
b5	VOUT2LSEL	VBATT Wakeup I/O 2 Output Level Selection	0: Output L before VBATT wakeup trigger 1: Output H before VBATT wakeup trigger.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The VBTOCTLR enables the VBATT wakeup I/O (VBATWIOn (n = 0 to 2)) pin outputs and selects the output level. VBTOCTLR is reset by the VBATT_POR signal.

VCHnOEN bit (VBATT Wakeup I/O n Output Enable Bit) (n = 0 to 2)

The VCHnOEN bit enables or disables the VBATT wakeup I/O pin outputs.

Note 1. Only one of these I/O pins can be set as output pin. Therefore, 2 out of the 3 bits must set to 0.
Note 2. When the VCH0OEN bit is set to 1, P402PFS.PMR bit must be 0.
When the VCH1OEN bit is set to 1, P403PFS.PMR bit must be 0.
When the VCH2OEN bit is set to 1, P404PFS.PMR bit must be 0.

VOUTnLSEL bit (VBATT Wakeup I/O n Output Level Selection) (n = 0 to 2)

The VOUTnLSEL bit selects the output level from the VBATT wakeup I/O n pin. When the VOUTnLSEL bit is set to 0, VBATWIOn pin outputs low before receiving VBATT wakeup trigger and high after receiving VBATT wakeup trigger. When the VOUTnLSEL bit is set to 1, the VBATWIOn pin outputs high before the VBATT wakeup trigger and low after receiving the VBATT wakeup trigger.

12.2.13 VBATT Wakeup Trigger Source Enable Register (VBTWTER)

Address(es): SYSTEM.VBTWTER 4001 E4BDh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	VAGTUE	VRTCAE	VRTCIE	VCH2E	VCH1E	VCH0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	VCH0E	VBATWIO0 Pin Enable	0: Disable VBATT wakeup triggered in the VBATWIO0 pin 1: Enable VBATT wakeup triggered in the VBATWIO0 pin.	R/W
b1	VCH1E	VBATWIO1 Pin Enable	0: Disable VBATT wakeup triggered in the VBATWIO1 pin 1: Enable VBATT wakeup triggered in the VBATWIO1 pin.	R/W
b2	VCH2E	VBATWIO2 Pin Enable	0: Disable VBATT wakeup triggered in the VBATWIO2 pin 1: Enable VBATT wakeup triggered in the VBATWIO2 pin.	R/W
b3	VRTCIE	RTC Periodic Signal Enable	0: Disable VBATT wakeup triggered in RTC periodic signal 1: Enable VBATT wakeup triggered in RTC periodic signal.	R/W
b4	VRTCAE	RTC Alarm Signal Enable	0: Disable VBATT wakeup triggered in RTC alarm signal 1: Enable VBATT wakeup triggered in RTC alarm signal.	R/W
b5	VAGTUE	AGT1 Underflow Signal Enable	0: Disable VBATT wakeup triggered in AGT1 underflow signal 1: Enable VBATT wakeup triggered in AGT1 underflow signal.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The VBTWTER enables or disables the VBATT wakeup trigger. VBTWTER is reset by the VBATT_POR signal. Multiple trigger source selection is possible.

12.2.14 VBATT Wakeup Trigger Source Edge Register (VBTWEGR)

Address(es): SYSTEM.VBTWEGR 4001 E4BEh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	VCH2EG	VCH1EG	VCH0EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	VCH0EG	VBATWIO0 Wakeup Trigger Source Edge Select	0: Generate wakeup trigger on a falling edge 1: Generate wakeup trigger on a rising edge.	R/W
b1	VCH1EG	VBATWIO1 Wakeup Trigger Source Edge Select	0: Generate wakeup trigger on a falling edge 1: Generate wakeup trigger on a rising edge.	R/W
b2	VCH2EG	VBATWIO2 Wakeup Trigger Source Edge Select	0: Generate wakeup trigger on a falling edge 1: Generate wakeup trigger on a rising edge.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The VBTWEGR selects the edge of each VBATT wakeup trigger sources. The VBTWEGR register is reset by the VBATT_POR signal.

12.2.15 VBATT Wakeup Trigger Source Flag Register (VBTWFR)

Address(es): SYSTEM.VBTWFR 4001 E4BFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	VAGTUF	VRTCAF	VRTCIF	VCH2F	VCH1F	VCH0F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	VCH0F	VBATWIO0 Wakeup Trigger Flag	0: Wakeup trigger in the VBATWIO0 pin not generated 1: Wakeup trigger in the VBATWIO0 pin is generated.	R/(W)*
b1	VCH1F	VBATWIO1 Wakeup Trigger Flag	0: Wakeup trigger in the VBATWIO1 pin not generated 1: Wakeup trigger in the VBATWIO1 pin is generated.	R/(W)*
b2	VCH2F	VBATWIO2 Wakeup Trigger Flag	0: Wakeup trigger in the VBATWIO2 pin not generated 1: Wakeup trigger in the VBATWIO2 pin is generated.	R/(W)*
b3	VRTCIF	VBATT RTC-Periodic Wakeup Trigger Flag	0: Wakeup trigger in the RTC periodic signal not generated 1: Wakeup trigger in the RTC periodic signal is generated.	R/(W)*
b4	VRTCAF	VBATT RTC-Alarm Wakeup Trigger Flag	0: Wakeup trigger in the RTC alarm signal not generated 1: Wakeup trigger in the RTC alarm signal is generated.	R/(W)*
b5	VAGTUF	VBATT AGT1 Underflow Wakeup Trigger Flag	0: Wakeup trigger in the AGT1 underflow not generated 1: Wakeup trigger in the AGT1 underflow is generated.	R/(W)*
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Only write 0 to clear the flag after reading 1.

The VBTWFR indicates the triggering factor of the VBATT wakeup control function.

The VBTWCTLR.VWEN bit enables or disables the wakeup function. When VWEN is set to 1, 5 PCLKB cycles of wait time is required before access to this register is valid. Similarly, when VWEN bit is set to 0, the VBTWFR register is disabled after 5 PCLKB cycles after writing.

When the trigger request specified by this register is generated, the corresponding flag in this register is set to 1.

The VBTWFR register is initialized by VBATT_POR.

VCHnF flags (VBATT Wakeup I/O n Wakeup Trigger Flag) (n = 0 to 2)

The VCHnF flags indicate that a trigger request by the VBATWIOn pin is generated.

[Setting condition]

- A trigger request by the VBATWIOn pin specified by VBTWEGR is generated.

[Clearing condition]

- Each bit is read as 1 and then written as 0.

VRTCIF flag (VBATT RTC-Periodic Wakeup Trigger Flag)

This flag indicates that a trigger request by the RTC periodic signal is generated.

[Setting condition]

- A cancel request by the RTC periodic signal is generated.

[Clearing condition]

- This bit is read as 1 and then written as 0.

VRTCAF flag (VBATT RTC-Alarm Wakeup Trigger Flag)

This flag indicates that a trigger request by the RTC alarm signal is generated.

[Setting condition]

- A cancel request by the RTC alarm signal is generated.

[Clearing condition]

- This bit is read as 1 and then written as 0.

VAGTUF flag (VBATT AGT1 Underflow Wakeup Trigger Flag)

This flag indicates that a trigger request by the AGT signal is generated.

[Setting condition]

- A cancel request by the AGT1 underflow signal is generated.

[Clearing condition]

- This bit is read as 1 and then written as 0.

12.3 Operation

12.3.1 Battery Backup Function

When the voltage at the VCC pin drops, power can be supplied to the RTC, AGT, LOCO, and sub-clock oscillator from the VBATT pin. When a power supply drop from the VCC pin is detected, the power connection is switched from the power supply to the VBATT pin. The power supply from the VCC pin resumes when voltage at the VCC pin exceeds $V_{DETBATT}$. The power supply change does not affect the RTC operation. When the voltage level at the VBATT pin voltage drops below the operation-guaranteed voltage, the VBTBLDF bit can be monitored in the VBATT Status Register.

The battery backup function can be used after the voltage monitor 0 reset is enabled.

While VBATT supplies the power, the wakeup control function can toggle the output pin of VBATWIOn (n = 0 to 2) by the trigger of the alarm/periodic signal, AGT1 underflow signal, or VBATWIOn (n = 0 to 2) input signal asserted.

The RTC supports time capture detection, triggered by a change in the time capture pin input level.

The VBATT pin supplies power to the following modules.

- RTC
- AGT0 and AGT1
- Sub-clock oscillator (including XCIN and XCOUT pins)
- VBATWIOn pins (including RTCICn, AGTIOm_B, and AGTIOm_C) (n = 0 to 2, m = 0 to 1)
- LOCO
- VBATT Backup Register
- VBATT wakeup controller.

Table 12.2 shows the operating states in VBATT mode.

Table 12.2 Operating states in VBATT mode (1 of 2)

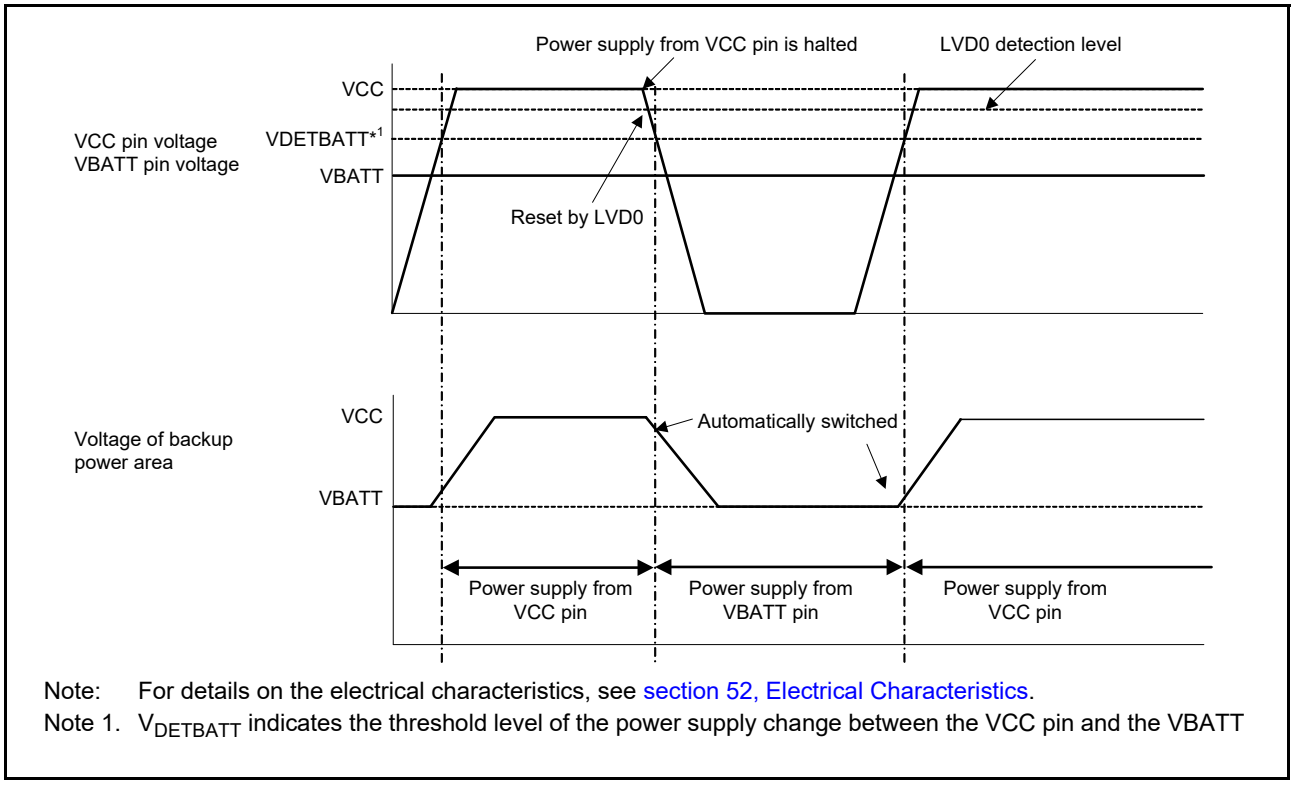
Operating state	VBATT mode
Transition condition	Detection of VCC voltage drop
Canceling method other than reset	Detection of VCC voltage rise
Main clock oscillator	Stopped
Sub-clock oscillator	Operation can be selected by SOSCCR.SOSTP bit. The status of the oscillator is the same as before entering VBATT mode.

Table 12.2 Operating states in VBATT mode (2 of 2)

Operating state	VBATT mode
High-speed on-chip oscillator	Stopped
Middle-speed on-chip oscillator	Stopped
Low-speed on-chip oscillator	Operating
IWDT-dedicated on-chip oscillator	Stopped
PLL	Stopped
CPU	Stopped (undefined)
SRAM (ECC SRAM included)	Stopped (undefined)
VBATT Backup Register	Stopped (retained)
Flash memory	Stopped (retained)
Realtime Clock (RTC)	Selectable when selecting clock that operates as the count source
AGTn (n = 0, 1)	Selectable when selecting clock that operates as the count source. Event count mode without noise filter is available.
Low Voltage Detection (LVD)	Stopped
Power-on reset circuit	Stopped
Battery backup voltage monitor	Operating
Other peripheral modules	Stopped (undefined)
I/O Ports	• AGTIOn_B port (n = 0, 1): Operating (input) • AGTIOn_C port (n = 0, 1): Operating (input) • RTCICn ports (n = 0 to 2): Operating • Other than the specified ports: Undefined • VBATWIOn ports (n = 0 to 2): Operating.

Note: Selectable means that operating can be selected by the control register. Some modules are also controlled by the corresponding module-stop bit.
Stopped (retained) means that the contents of the internal registers are retained but the operations are suspended.
Stopped (undefined) means that the contents of the internal registers are undefined and power to the internal circuit is cut off.

Figure 12.3 shows the switching sequence of the battery backup function.



Note: For details on the electrical characteristics, see section 52, Electrical Characteristics.
Note 1. $V_{DETBATT}$ indicates the threshold level of the power supply change between the VCC pin and the VBATT

Figure 12.3 Switching sequence for the battery backup function

12.3.2 VBATT Battery Power Supply Switch Usage

The battery power supply switch can switch the battery backup module supply from the VCC pin to the VBATT pin, when the voltage being applied to the VCC pin drops. When the voltage rises, this switch changes the power supply from the VBATT pin to the VCC pin. The switch is controlled by the VBTCR1.BPWSWSTP bit.

When the switch is stopped, the battery backup module power supply is always from VCC. If you are not using the battery backup function, you must write 1 to this bit.

Note: You can use the battery backup function after the voltage monitor 0 reset is enabled (OFS1.LVDAS bit is 0). Voltage monitor 0 level should be higher than the VBATT switch level (OFS1.VDSEL1[2:0] bits are 000b, 001b, or 010b).

Note: This bit can be set without verifying the VBATTSR.VBTRVLD bit status.

12.3.3 VBATT Pin Low Voltage Detection Procedures

The VBTSR.VBTBLDF flag and interrupt can be used to monitor VBATT pin low voltage detection using the procedures described in this section.

The following procedure shows how to enable the VBATT pin low voltage detection:

1. Set the voltage monitor 0 reset. See section 8, Low Voltage Detection (LVD).
2. Set the VBTCR1.BPWSWSTP bit to 1 if this bit is being accessed for the first time after a power-on reset.
3. Wait for the VBTSR.VBTRVLD bit to be 1 and confirm that the VBTCR2.VBTLVDEN, VBTLVDICR.VBTLVDIE, and VBTCMPCR.VBTCMPE bits are 0.
4. Specify the detection voltage in the VBTCR2.VBTLVDLVL[1:0] bits (VBATT pin voltage detect level select).
5. Select the type of interrupt in the VBTLVDICR.VBTLVDISEL bit.
6. Set the VBTCR2.VBTLVDEN bit to 1 to enable VBATT pin low voltage detection.
7. Wait for the VBATT comparator operation stabilization time (t_{d_vbat}) as described in section 52, Electrical Characteristics, and then set the VBTCMPCR.VBTCMPE bit to 1 for VBATT pin voltage detect circuit enable.
8. Make sure that the VBTSR.VBTBLDF flag is 0, and then set the VBTLVDICR.VBTLVDIE bit to 1 for the VBATT pin low voltage detection interrupt enable.
9. Clear the VBTCR1.BPWSWSTP bit to 0 to enable the battery power switch. See section 12.3.2, VBATT Battery Power Supply Switch Usage.

When the VBATT low voltage is detected, disable the VBATT low voltage detection as shown in Figure 12.4.

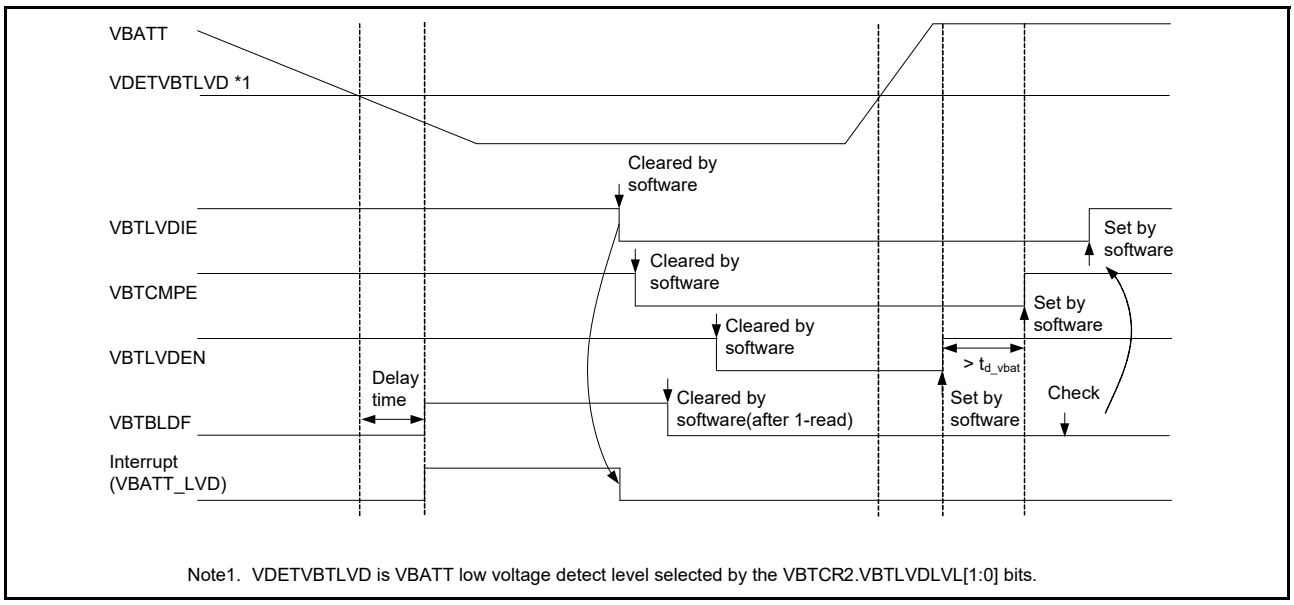


Figure 12.4 Basic operation of VBATT low voltage detection interrupt

The following procedure shows how to disable the VBATT pin low voltage detection:

1. Make sure that the VBTSR.VBTRVLD bit is 1.
2. Set the VBTLVDICR.VBTLVDIE bit to 0 to disable voltage detect interrupt enable.
3. Set the VBTCMPCR.VBTCMPE bit to 0 for VBATT pin voltage detect circuit disable.
4. Set the VBTCR2.VBTLVDEN bit to 0 to disable VBATT pin low voltage detection.
5. Modify the setting of bits related to the VBATT pin low voltage detection registers other than VBTCR2.VBTLVDEN, VBTCMPCR.VBTCMPE, VBTLVDICR.VBTLVDIE.

12.3.4 VBATT Backup Register Usage

The VBATT backup register VBTBKRn where n = 0 to 511, can be used to store or restore data as described in the following procedure:

1. The VBTCR1.BPWSWSTP bit must be set to 1 if this bit is being accessed for the first time after a power-on reset.
2. Wait for the VBTSR.VBTRVLD bit to be 1.
3. VBTBKR[n] where n = 0 to 511 can be accessed by an 8-bit read or write operation.
4. Clear the VBTCR1.BPWSWSTP bit to 0 to enable the battery power switch. See section 12.3.2, VBATT Battery Power Supply Switch Usage.

12.3.5 VBATT Wakeup Control Function Usage

The wakeup control function can toggle the output pin VBATWIOn (n = 0 to 2) when the RTC alarm/periodic signal, AGT1 underflow signal, or VBATWIOn (n = 0 to 2) input signal is asserted, when VBATT_R is powered by the VBATT pin.

Note: The toggle triggered in the wakeup control function does not generate an interrupt at the ICU or a reset at the reset module.

Figure 12.5 shows an example for using the VBATT wakeup control function. In this example, VBATWIO0 is the wakeup output port, the AGTIO1_C port is the external pulse input port, the RTCIC2 port is the external time capture input capture port, and the VBATWIO2 port is the external time capture input trigger port. The VBATWIO0 output toggles from low level to high level when the associated trigger is asserted. The trigger source for the wakeup control function is the AGT1 timer underflow signal, RTC periodic signal, or VBATWIO2 input rising edge.

Use the following steps to set the VBATT wakeup control function:

1. Set the VBTCR1.BPWSWSTP bit to 1 if this bit is being accessed for the first time after a power-on reset.
2. Wait for the VBTSR.VBTRVLD bit to be 1. Then, confirm that the VBTWCTLR.VWEN and VBTSR.VBTRDF bits are 0. If these bits are not 0, clear them to 0.
3. Specify the VBATWIOn port direction in the VBTICTLR.VCHnINEN and VBTOCTLR.VCHnOEN bits. Set the VBTOCTLR.VOUTnLSEL bit (n = 0 to 2) to 0 or 1 to select the output level.
   In this example, use the VBATWIO2/RTCIC2 port as the time capture input, VBATWIO0 port as wakeup output port, and VBATWIO1 port as external pulse input port.
   Set the following bits to 1:
   - VBTOCTLR.VCH0OEN
   - VBTICTLR.VCH1INEN
   - VBTICTLR.VCH2INEN.

   In addition, set VBTOCTLR.VOUT0LSEL to 0 to toggle the output from low level to high level.
4. Set the peripheral module setting as required.
   In this example, specify the time capture function for time capture setting with the RTC setting. See section 25, Realtime Clock (RTC) for details. Also set the AGT timer setting. See section 24, Asynchronous General Purpose Timer (AGT) for details.
5. Select the wakeup trigger source in the VBTWTER register.
   In this example, set the VBTWTER.VAGTUE, VBTWTER.VRTCIE, and VBTWTER.VCH2E bits to 1 to select the trigger source as the AGT1 underflow signal, RTC periodic signal, and VBATWIO2 input trigger.

6. Select the wakeup trigger source edge in the VBTWEGR register.
   For example, set the VBTWEGR.VCH2EG bit to 1 to select the VBATWIO2 port as the rising-edge trigger.
7. Select the VBATT wakeup output trigger source in the VBTWCHnOTSR register (n = 0 to 2).
   In this example, set the VBTWCH0OTSR.CH0VAGTUTE, VBTWCH0OTSR.CH0VRTCTE, and VBTWCH0OTSR.CH0VCH2TE bits to 1.
8. Set the VBTWCTLR.VWEN bit to 1 to activate the VBATT wakeup control function, and then clear the VBTCR1.BPWSWSTP bit to 0 to enable the battery power supply switch. Again, set the VBTWCTLR.VWEN bit to 1 to enable the VBATT wakeup control function.
9. Set the I/O registers to output 0 or 1 to the external power management IC to request stopping the power supply.
   After stopping the power supply, if the AGT1 underflow signal, RTC periodic signal, or VBATWIO2 input trigger is asserted, the VBATT wakeup trigger source flag for the associated event (VBTWFR.VAGTF, VBTWFR.VRTCIF, or VBTWFR.VCH2F) is set to 1, and the output toggles from low level to high level on the VBATWIO0 port. Then the MCU is supplied power, and it starts up from a low voltage monitor 0 reset (LVD0). In this example, the external power management IC stops supplying power when it detects a positive transition on the I/O port powered by VCC pin, and starts to supply power when it detects a positive transition on the VBATWIO0 port.
   The timing chart of VBATT wakeup function is shown in Figure 12.6.

The following procedure shows how to set the registers after the MCU starts up from a low voltage monitor 0 reset (LVD0) by the VBATT wakeup trigger.

1. Set the VBTCR1.BPWSWSTP bit to 1.
2. Wait for the VBTSR.VBTRVLD bit to be 1 and confirm that the VBTSR.VBTRDF bit is 0.
3. Check the VBATT wakeup trigger source by reading the VBTWFR register. In the example shown in Figure 12.6, the VBTWFR.VRTCIF bit is set to 1.
4. Clear the corresponding bit in the VBTWFR register to 0 to toggle the output on the VBATWIOn port (n = 0 to 2).
   In the example shown in Figure 12.6, the output is toggled from high to low on the VBATWIO0 port.
5. Set the I/O registers to output 0 or 1 to the external power management IC as needed.
6. To repeat the VBATT wakeup operation, clear the VBTCR1.BPWSWSTP bit to 0 and set the I/O registers to output 0 or 1 to the external power management IC to request stopping power supply again.
   If you want to change the wakeup trigger conditions, clear the VBTWCTLR.VWEN bit to 0, and clear the all bit in the VBTWTER register before setting other registers associated with VBATT.

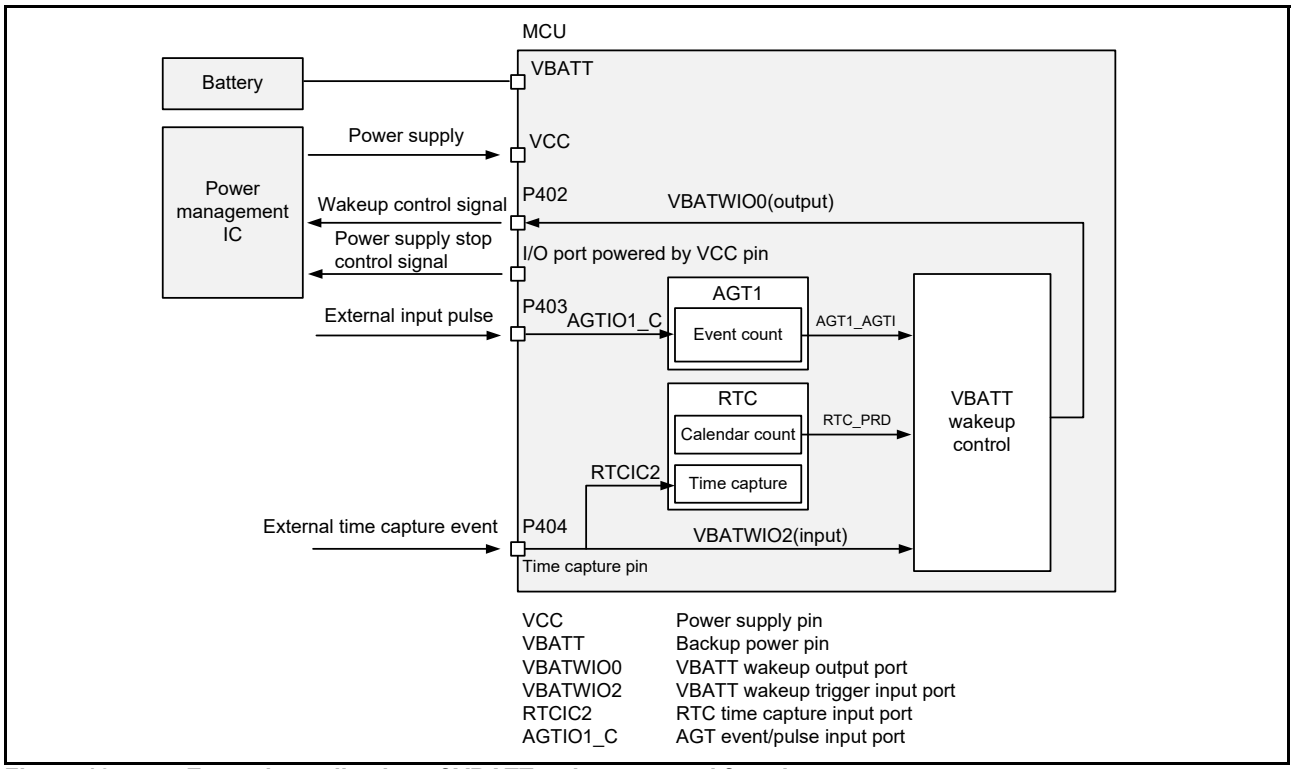
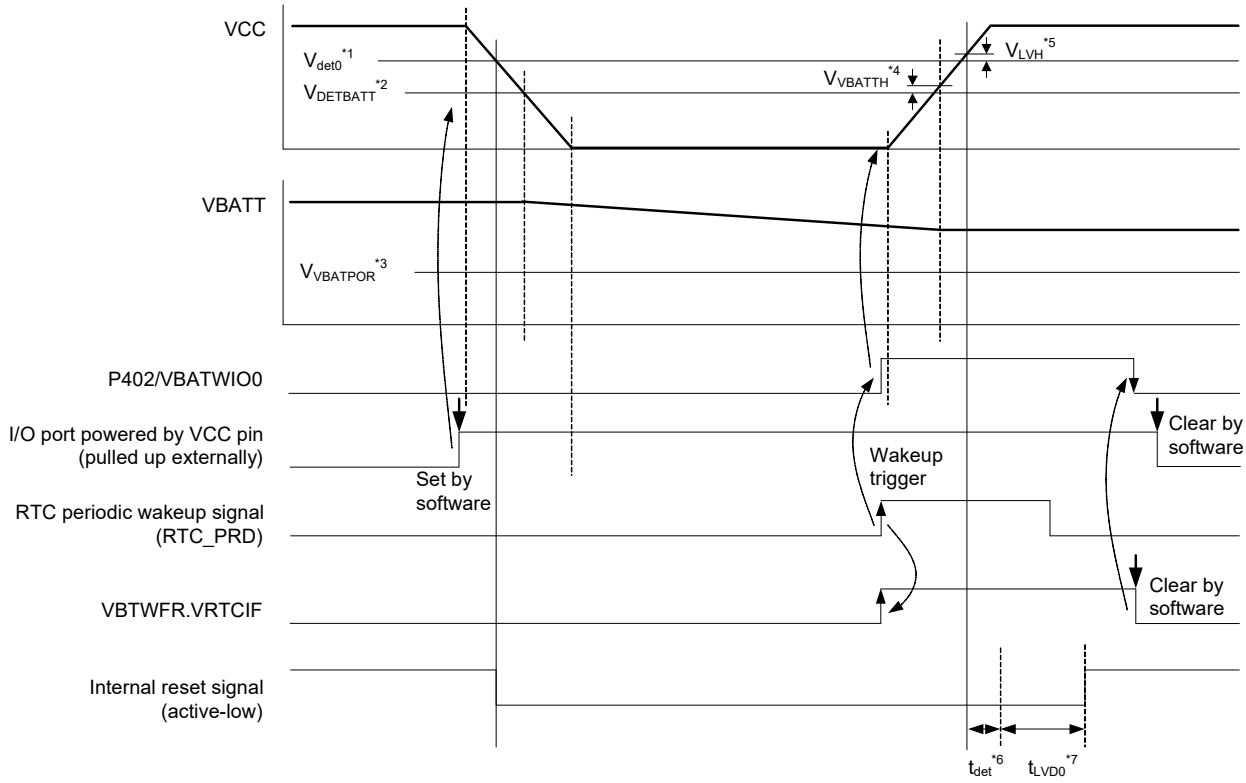


Figure 12.5 Example application of VBATT wakeup control function



Note 1. V_{det0} is the LVD0 voltage detection level selected by the OFS1.VDSEL1[2:0] bits.
Note 2. $V_{DETBATT}$ is the voltage level for switching to battery backup.
Note 3. $V_{VBATPOR}$ is the voltage detection level VBATT power on reset.
Note 4. V_{VBATTH} is the hysteresis width for switching to battery backup.
Note 5. V_{LVH} is the hysteresis width of LVD0.
Note 6. t_{det} is the response delay time of LVD0.
Note 7. t_{LVD0} is the internal reset time after release from LVD0 reset.

Figure 12.6 The timing chart of VBATT wakeup function

12.4 Usage Note

1. When the VBATT pin is not in use, connect the VBATT pin to the VCC pin.
2. When the voltage level at the VBATT is lower than the guaranteed operation range, operation of the sub-clock oscillator and RTC cannot be guaranteed. This voltage drop can be verified in the VBTSR register.
3. A reset generated while writing to the registers described in this section might destroy the register value.
4. During RTC operation powered by the VBATT pin, RTC supports the calendar/binary count operation, the alarm/ periodic trigger for the VBATT wakeup function, and the time capture function.
5. The VBATT wakeup control function can be used when VBATT_R is powered by VBATT pin only.
6. When a system in Software Standby mode accesses the VBATT backup registers, you must keep the LOCO clock active by setting the LOCOCR.LCSTP bit to 0 (LOCO is operating).
7. The voltage level on the I/O ports powered by the VCC pin transitions to high-impedance when the power supply is stopped. If these ports are used as the power supply stop control pin for VBATT wakeup function, these ports should be pulled up or down externally.

13. Register Write Protection

13.1 Overview

The register write protection function protects important registers from being overwritten because of software errors. The registers to be protected are set with the Protect Register (PRCR). Table 13.1 lists the association between the PRCR bits and the registers to be protected.

Table 13.1 Association between PRCR bits and registers to be protected

PRCR bit	Registers to be protected
PRC0	• Registers related to the Clock Generation Circuit: SCKDIVCR, SCKSCR, PLLCR, PLLCCR2, BCKCR, MEMWAIT, MOSCCR, HOCOCR, MOCOCR, CKOCR, TRCKCR, OSTDCR, OSTDSR, SLCDSCKCR, EBCKOCR, MOCOUTCR, HOCOUTCR, MOSCWTCR, MOMCR, SOSCCR, SOMCR, LOCOCR, LOCOUTCR, HOCOWTCR
PRC1	• Registers related to the low power modes: SBYCR, SNZCR, SNZEDCR, SNZREQCR, FLSTOP, PSMCR, OPCCR, SOPCCR, SYOCDCR • Registers related to the battery backup function: VBTCR1, VBTCR2, VBTSR, VBTCMPCR, VBTLVDICR, VBTWCTLR, VBTWCH0OTSR, VBTWCH1OTSR, VBTWCH2OTSR, VBTICTLR, VBTOCTLR, VBTWTER, VBTWEGR, VBTWFR, VBTBKRn (n = 0 to 511)
PRC3	• Registers related to the LVD: LVD1CR1, LVD1SR, LVD2CR1, LVD2SR, LVCMPCR, LVDLVLR, LVD1CR0, LVD2CR0

13.2 Register Descriptions

13.2.1 Protect Register (PRCR)

Address(es): SYSTEM.PRCR 4001 E3FEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit name	Function	R/W
b0	PRC0	Protect Bit 0	Enables writing to the registers related to the clock generation circuit: 0: Disable writes 1: Enable writes.	R/W
b1	PRC1	Protect Bit 1	Enables writing to the registers related to the low power modes and the battery backup function. 0: Disable writes 1: Enable writes.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	PRC3	Protect Bit 3	Enables writing to the registers related to the LVD: 0: Disable writes 1: Enable writes.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	PRC Key Code	These bits control write access to the PRCR register. To modify the PRCR register, write A5h to the upper 8 bits and the target value to the lower 8 bits as a 16-bit unit.	W*1

Note 1. Write data is not saved. Always reads 00h.

PRCn bits (Protect Bit n) (n = 0, 1, 3)

These bits enable or disable writing to the protected registers as described in Table 13.1. Setting the PRCn bits to 1 or 0 enables or disables writing, respectively.

14. Interrupt Controller Unit (ICU)

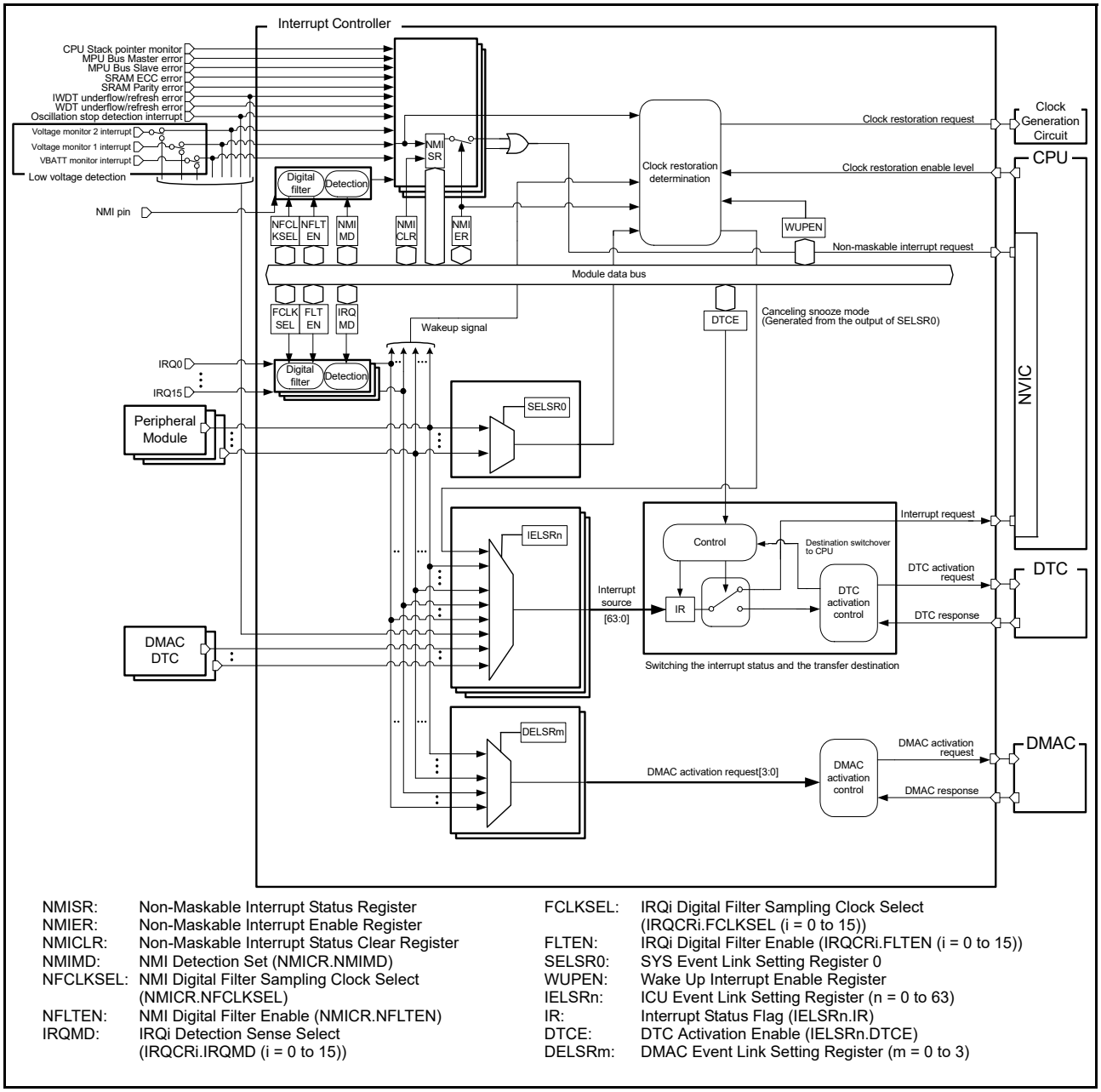
14.1 Overview

The Interrupt Controller Unit (ICU) controls which event signals are linked to the NVIC, DTC, and DMAC modules. The ICU also controls non-maskable interrupts. Table 14.1 lists the ICU specifications, Figure 14.1 shows the block diagram, and Table 14.2 lists the I/O pins.

Table 14.1 ICU specifications

Parameter		Description
Interrupts	Peripheral function interrupts	• Interrupts from peripheral modules Number of sources: 205 (select factors within event list numbers 32 to 458)
	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 • 16 sources, with interrupts from IRQ0 to IRQ15 pins.
	DTC and DMAC control	The DTC and DMAC can be activated by interrupt sources*1
	Interrupt sources for NVIC	• 64 sources
Non-maskable interrupts*2	NMI pin interrupt	• Interrupt from the NMI pin • Interrupt detection on falling edge or rising edge • Digital filter function supported.
	Oscillation stop detection interrupt*3	Interrupt on detecting that the main oscillation has stopped
	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
	Voltage monitor 1 interrupt*3	Voltage monitor interrupt of low voltage detection detector 1 (LVD_LVD1)
	Voltage monitor 2 interrupt*3	Voltage monitor interrupt of low voltage detection detector 2 (LVD_LVD2)
	VBATT interrupt	Voltage monitor interrupt of VBATT monitor
	RPEST	Interrupt on SRAM parity error
	RECCST	Interrupt on SRAM ECC error
	BUSSST	Interrupt on MPU bus slave error
	BUSMST	Interrupt on MPU bus master error
	SPEST	Interrupt on CPU stack pointer monitor
Return from low power mode		• Sleep mode: Return is initiated by non-maskable interrupts or any other interrupt source • Software Standby mode: Return is initiated by non-maskable interrupts. Interrupts can be selected in the WUPEN register • Snooze mode: Return is initiated by non-maskable interrupts. Interrupts can be selected in the SELSR0 and WUPEN registers. See section 14.2.8, SYS Event Link Setting Register (SELSR0) and section 14.2.9, Wake Up Interrupt Enable Register (WUPEN).

Note 1. For the DTC and DMAC activation sources, see Table 14.4, Event table.
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 event signals. When used as 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. To enable the VBATT monitor interrupt, set the VBTLVDICR.VBTLVDISEL bit to 1.
Note 4. Interrupt detection is not canceled if you do not clear it after a detection.



NMISR: Non-Maskable Interrupt Status Register
NMIER: Non-Maskable Interrupt Enable Register
NMICLR: Non-Maskable Interrupt Status Clear Register
NMIMD: NMI Detection Set (NMICR.NMIMD)
NFCLKSEL: NMI Digital Filter Sampling Clock Select (NMICR.NFCLKSEL)
NFLTEN: NMI Digital Filter Enable (NMICR.NFLTEN)
IRQMD: IRQi Detection Sense Select (IRQCRi.IRQMD (i = 0 to 15))

FCLKSEL: IRQi Digital Filter Sampling Clock Select (IRQCRi.FCLKSEL (i = 0 to 15))
FLTEN: IRQi Digital Filter Enable (IRQCRi.FLTEN (i = 0 to 15))
SELSR0: SYS Event Link Setting Register 0
WUPEN: Wake Up Interrupt Enable Register
IELSRn: ICU Event Link Setting Register (n = 0 to 63)
IR: Interrupt Status Flag (IELSRn.IR)
DTCE: DTC Activation Enable (IELSRn.DTCE)
DELSRm: DMAC Event Link Setting Register (m = 0 to 3)

Figure 14.1 ICU block diagram

Table 14.2 ICU configuration pins

Pin name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQ0 to IRQ15	Input	External interrupt request pins

14.2 Register Descriptions

This chapter does not describe Arm® NVIC internal registers. For information on these registers, see the *ARM® Cortex®-M4 Processor Technical Reference Manual (ARM DDI 0439D)*.

14.2.1 IRQ Control Register i (IRQCRi) (i = 0 to 15)

Address(es): ICU.IRQCR0 4000 6000h, ICU.IRQCR1 4000 6001h, ICU.IRQCR2 4000 6002h, ICU.IRQCR3 4000 6003h, ICU.IRQCR4 4000 6004h, ICU.IRQCR5 4000 6005h, ICU.IRQCR6 4000 6006h, ICU.IRQCR7 4000 6007h ICU.IRQCR8 4000 6008h, ICU.IRQCR9 4000 6009h, ICU.IRQCR10 4000 600Ah, ICU.IRQCR11 4000 600Bh, ICU.IRQCR12 4000 600Ch, ICU.IRQCR13 4000 600Dh, ICU.IRQCR14 4000 600Eh, ICU.IRQCR15 4000 600Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	FLTEN	—	FCLKSEL[1:0]		—	—	IRQMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	IRQMD[1:0]	IRQi Detection Sense Select	b1 b0 0 0: Falling edge 0 1: Rising edge 1 0: Rising and falling edges 1 1: Low level.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	FCLKSEL[1:0]	IRQi Digital Filter Sampling Clock Select	b5 b4 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	FLTEN	IRQi Digital Filter Enable	0: Disable digital filter 1: Enable digital filter.	R/W

IRQCRi register changes must satisfy the following:

- For a CPU interrupt or DTC trigger:
  Change the IRQCRi register setting before setting the target IELSRn (n = 0 to 63).
  You can change the register values only when the IELSRn.IELS[8:0] bits are 0.
- For a DMAC trigger:
  Change the IRQCRi register setting before setting the target DELSRn (n = 0 to 3).
  You can change the register values only when the DELSRn.DELS[8:0] bits are 0.
- For a wakeup enable signal:
  Change the IRQCRi register setting before setting the target WUPEN.IRQWUPENn (n = 0 to 15).
  You can change the register values only when the target WUPEN.IRQWUPENn 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 more information on the settings, see section 14.4.4, 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, selectable to:

- PCLKB (every cycle)
- PCKLB/8 (once every 8 cycles)
- PCKLB/32 (once every 32 cycles)
- PCKLB/64 (once every 64 cycles).

For details of the digital filter, see section 14.4.3, Digital Filter.

FLTEN bit (IRQi Digital Filter Enable)

The FLTEN bits enable the digital filter used for the IRQi external pin interrupt sources. The filter is enabled when the FLTEN bit is 1, and disabled when the FLTEN bit is 0. The IRQi pin level is sampled at the cycle specified in FCLKSEL[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 14.4.3, Digital Filter.

14.2.2 Non-Maskable Interrupt Status Register (NMISR)

Address(es): ICU.NMISR 4000 6140h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SPEST	BUSMST	BUSSST	RECCST	RPEST	NMIST	OSTST	—	VBATTST	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	Bit name	Description	R/W
b0	IWDTST	IWDT Underflow/Refresh Error Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b1	WDTST	WDT Underflow/Refresh Error Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b2	LVD1ST	Voltage Monitor 1 Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b3	LVD2ST	Voltage Monitor 2 Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b4	VBATTST	VBATT monitor Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b5	—	Reserved	This bit is read as 0.	R
b6	OSTST	Oscillation Stop Detection Interrupt Status Flag	0: Interrupt not requested for main oscillation stop 1: Interrupt requested for main oscillation stop.	R
b7	NMIST	NMI Status Flag	0: NMI pin interrupt not requested. 1: NMI pin interrupt requested.	R
b8	RPEST	SRAM Parity Error Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b9	RECCST	SRAM ECC Error Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b10	BUSSST	MPU Bus Slave Error Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b11	BUSMST	MPU Bus Master Error Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b12	SPEST	CPU Stack Pointer Monitor Interrupt Status Flag	0: Interrupt not requested 1: Interrupt requested.	R
b15 to b13	—	Reserved	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 have occurred during handler processing.

IWDTST flag (IWDT Underflow/Refresh Error 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 an IWDT underflow/refresh error interrupt occurs and this interrupt is enabled.

[Clearing condition]

When 1 is written to the NMICLR.IWDTCLR bit.

WDTST flag (WDT Underflow/Refresh Error 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 a WDT underflow/refresh error interrupt occurs.

[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 a voltage monitor 1 interrupt occurs and this interrupt 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 a voltage monitor 2 interrupt occurs and this interrupt is enabled.

[Clearing condition]

When 1 is written to the NMICLR.LVD2CLR bit.

VBATTST flag (VBATT monitor Interrupt Status Flag)

The VBATTST flag indicates a VBATT monitor interrupt request. It is read-only and cleared by the NMICLR.VBATTCLR bit.

[Setting condition]

When a VBATT monitor interrupt occurs.

[Clearing condition]

When 1 is written to the NMICLR.VBATTCLR bit.

OSTST flag (Oscillation Stop Detection Interrupt Status Flag)

The OSTST flag indicates a main oscillation stop detection interrupt request. It is read-only and cleared by the NMICLR.OSTCLR bit.

[Setting condition]

When an oscillation stop detection interrupt occurs.

[Clearing condition]

When 1 is written to the NMICLR.OSTCLR bit.

NMIST flag (NMI 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 occurs in response to an SRAM Parity error.

[Clearing condition]

When 1 is written to the NMICLR.RPECLR bit.

RECCST flag (SRAM ECC Error Interrupt Status Flag)

The RECCST flag indicates an SRAM ECC error interrupt request.

[Setting condition]

When an interrupt occurs in response to an SRAM ECC error.

[Clearing condition]

When 1 is written to the NMICLR.RECCCLR bit.

BUSSST flag (MPU Bus Slave Error Interrupt Status Flag)

The BUSSST flag indicates a bus slave error interrupt request.

[Setting condition]

When an interrupt occurs in response to a bus slave error.

[Clearing condition]

When 1 is written to the NMICLR.BUSSCLR bit.

BUSMST flag (MPU Bus Master Error Interrupt Status Flag)

The BUSMST flag indicates a bus master error interrupt request.

[Setting condition]

When an interrupt occurs 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 occurs in response to a CPU stack pointer monitor.

[Clearing condition]

When 1 is written to the NMICLR.SPECLR bit.

14.2.3 Non-Maskable Interrupt Enable Register (NMIER)

Address(es): ICU.NMIER 4000 6120h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SPEEN	BUSMEN	BUSSEN	RECCEN	RPEEN	NMIEN	OSTEN	—	VBATTEN	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	Bit name	Description	R/W
b0	IWDTEN	IWDT Underflow/Refresh Error Interrupt Enable	0: Disable 1: Enable.	R/(W)*1, *2
b1	WDTEN	WDT Underflow/Refresh Error Interrupt Enable	0: Disable 1: Enable.	R/(W)*1, *2
b2	LVD1EN	Voltage Monitor 1 Interrupt Enable	0: Disable 1: Enable.	R/(W)*1, *2
b3	LVD2EN	Voltage Monitor 2 Interrupt Enable	0: Disable 1: Enable.	R/(W)*1, *2
b4	VBATTEN	VBATT Monitor Interrupt Enable	0: Disable 1: Enable.	R/(W)*1, *2
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	OSTEN	Oscillation Stop Detection Interrupt Enable	0: Disable 1: Enable.	R/(W)*1, *2
b7	NMIEN	NMI Pin Interrupt Enable	0: Disable 1: Enable.	R/(W)*1
b8	RPEEN	SRAM Parity Error Interrupt Enable	0: Disable 1: Enable.	R/(W)*1, *2
b9	RECCEN	SRAM ECC Error Interrupt Enable	0: Disable 1: Enable.	R/(W)*1, *2
b10	BUSSEN	MPU Bus Slave Error Interrupt Enable	0: Disable 1: Enable.	R/(W)*1, *2
b11	BUSMEN	MPU Bus Master Error Interrupt Enable	0: Disable 1: Enable.	R/(W)*1, *2
b12	SPEEN	CPU Stack Pointer Monitor Interrupt Enable	0: Disable 1: Enable.	R/(W)*1, *2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. A 1 can be written to this bit only once after reset, and 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.

VBATTEN bit (VBATT Monitor Interrupt Enable)

The VBATTEN bit enables VBATT monitor interrupt as an NMI trigger.

OSTEN bit (Oscillation Stop Detection Interrupt Enable)

The OSTEN bit enables main 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.

RECCEN bit (SRAM ECC Error Interrupt Enable)

The RECCEN bit enables SRAM ECC error interrupt as an NMI trigger.

BUSSEN bit (MPU Bus Slave Error Interrupt Enable)

The BUSSEN bit enables bus slave error interrupt as an NMI trigger.

BUSMEN bit (MPU Bus Master 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.

14.2.4 Non-Maskable Interrupt Status Clear Register (NMICLR)

Address(es): ICU.NMICLR 4000 6130h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SPECLR	BUSMCLR	BUSSCLR	RECCCLR	RPECLR	NMICLR	OSTCLR	—	VBATTCLR	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	Bit name	Description	R/W
b0	IWDTCLR	IWDT Clear	0: No effect 1: Clear the NMISR.IWDTST flag.	R/(W)*1
b1	WDTCLR	WDT Clear	0: No effect 1: Clear the NMISR.WDTST flag.	R/(W)*1
b2	LVD1CLR	LVD1 Clear	0: No effect 1: Clear the NMISR.LVD1ST flag.	R/(W)*1
b3	LVD2CLR	LVD2 Clear	0: No effect 1: Clear the NMISR.LVD2ST flag.	R/(W)*1
b4	VBATTCLR	VBATT Clear	0: No effect 1: Clear the NMISR.VBATTST flag.	R/(W)*1
b5	—	Reserved	The write value should be 0.	R/(W)*1
b6	OSTCLR	OST Clear	0: No effect 1: Clear the NMISR.OSTST flag.	R/(W)*1
b7	NMICLR	NMI Clear	0: No effect 1: Clear the NMISR.NMIST flag.	R/(W)*1
b8	RPECLR	SRAM Parity Error Clear	0: No effect 1: Clear the NMISR.RPEST flag.	R/(W)*1
b9	RECCCLR	SRAM ECC Error Clear	0: No effect 1: Clear the NMISR.RECCST flag.	R/(W)*1
b10	BUSSCLR	Bus Slave Error Clear	0: No effect 1: Clear the NMISR.BUSSST flag.	R/(W)*1
b11	BUSMCLR	Bus Master Error Clear	0: No effect 1: Clear the NMISR.BUSMST flag.	R/(W)*1
b12	SPECLR	CPU Stack Pointer Monitor Interrupt Clear	0: No effect. 1: Clear the NMISR.SPEST flag.	R/(W)*1
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/(W)*1

Note 1. Only 1 can be written to this bit.

IWDTCLR bit (IWDT Clear)

Writing 1 to the IWDTCLR bit clears the NMISR.IWDTST flag. This bit is read as 0.

WDTCLR bit (WDT Clear)

Writing 1 to the WDTCLR bit clears the NMISR.WDTST flag. This bit is read as 0.

LVD1CLR bit (LVD1 Clear)

Writing 1 to the LVD1CLR bit clears the NMISR.LVD1ST flag. This bit is read as 0.

LVD2CLR bit (LVD2 Clear)

Writing 1 to the LVD2CLR clears the NMISR.LVD2ST flag. This bit is read as 0.

VBATTCLR bit (VBATT Clear)

Writing 1 to the VBATTCLR bit clears the NMISR.VBATTST flag. This bit is read as 0.

OSTCLR bit (OST Clear)

Writing 1 to the OSTCLR bit clears the NMISR.OSTST flag. This bit is read as 0.

NMICLR bit (NMI Clear)

Writing 1 to the NMICLR bit clears the NMISR.NMIST flag. This bit is read as 0.

RPECLR bit (SRAM Parity Error Clear)

Writing 1 to the RPECLR bit clears the NMISR.RPEST flag. This bit is read as 0.

RECCCLR bit (SRAM ECC Error Clear)

Writing 1 to the RECCCLR bit clears the NMISR.RECCST flag. This bit is read as 0.

BUSSCLR bit (Bus Slave Error Clear)

Writing 1 to the BUSSCLR bit clears the NMISR.BUSSST flag. This bit is read as 0.

BUSMCLR bit (Bus Master Error Clear)

Writing 1 to the BUSMCLR bit clears the NMISR.BUSMSST flag. This bit is read as 0.

SPECLR bit (CPU Stack Pointer Monitor Interrupt Clear)

Writing 1 to the SPECLR bit clears the NMISR.SPEST flag. This bit is read as 0.

14.2.5 NMI Pin Interrupt Control Register (NMICR)

Address(es): ICU.NMICR 4000 6100h

	b7	b6	b5	b4	b3	b2	b1	b0
	NFLTEN	—	NFCLKSEL[1:0]		—	—	—	NMIMD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	NMIMD	NMI Detection Set	0: Falling edge 1: Rising edge.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	NFCLKSEL[1:0]	NMI Digital Filter Sampling Clock Select	b5 b4 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	NFLTEN	NMI Digital Filter Enable	0: Disable the digital filter 1: Enable the digital filter.	R/W

Change the NMICR register settings before enabling NMI pin interrupts (before setting NMIER.NMIEN to 1).

NMIMD bit (NMI Detection Set)

The NMIMD bit selects the detection sensing method for NMI pin interrupts.

NFCLKSEL[1:0] bits (NMI Digital Filter Sampling Clock Select)

These bits select the digital filter sampling clock for NMI pin interrupts, selectable to:

- PCLKB (every cycle)
- PCKLB/8 (once every 8 cycles)
- PCKLB/32 (once every 32 cycles)
- PCKLB/64 (once every 64 cycles).

For the digital filter details, see section 14.4.3, 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 NMIFLTC.NFCLKSEL[1:0]. When the sampled level matches three times, the output level from the digital filter changes. For digital filter details, see section 14.4.3, Digital Filter.

14.2.6 ICU Event Link Setting Register n (IELSRn) (n= 0 to 63)

Address(es): ICU.IELSR0 4000 6300h, ICU.IELSR1 4000 6304h, ICU.IELSR2 4000 6308h, ICU.IELSR3 4000 630Ch..... .....ICU.IELSR60 4000 63F0h, ICU.IELSR61 4000 63F4h, ICU.IELSR62 4000 63F8h, ICU.IELSR63 4000 63FCh

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IELS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	IELS[8:0]	ICU Event Link Select	b8 b0 000000000: Disable interrupts to the associated NVIC or DTC 000000001 to 111001010: Event signal number to be linked. For details, see Table 14.4.	R/W*1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	IR	Interrupt Status Flag	0: No interrupt request occurred 1: An interrupt request occurred.	R/(W)*2
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	DTCE	DTC Activation Enable	0: Disable DTC activation 1: Enable DTC activation.	R/W
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This register requires halfword or word access.
Note 2. Writing 1 to the IR flag is prohibited.

The IELSRn register selects the IRQ source used by NVIC. For details, see Table 14.4, Event table. IELSRn, where n = 0 to 63, corresponds to the NVIC IRQ input source numbers 0 to 63.

IELS[8:0] bits (ICU Event Link Select)

The IELS[8:0] bits link an event signal to the associated NVIC or DTC module. All IELS[8:0] bits must be written to simultaneously.

IR flag (Interrupt Status Flag)

The IR flag indicates an individual interrupt request from the event specified in IELS[8:0].

[Setting condition]

When an interrupt request is received from the associated peripheral module or IRQi pin.

[Clearing conditions]

When 0 is written to the IR flag. The DTCE bit must be set to 0 before writing 0 to the IR flag.

To clear the IR flag:

1. Negate the input 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 conditions]

- 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 bit.

14.2.7 DMAC Event Link Setting Register n (DELSRn)

Address(es): ICU.DELSR0 4000 6280h, ICU.DELSR1 4000 6284h, ICU.DELSR2 4000 6288h, ICU.DELSR3 4000 628Ch

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DELS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	DELS[8:0]	DMAC Event Link Select	b8 b0 000000000:Disable DMA start request to the associated DMAC module. 000000001 to 111001010: Event signal number to be linked. Other settings are prohibited. For details, see Table 14.4, Event table.	R/W*1
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This register requires halfword or word access.

DELS[8:0] bits (DMAC Event Link Select)

The DELS[8:0] bits link an event signal for the DMAC module. All DELS[8:0] bits must be written to simultaneously.

14.2.8 SYS Event Link Setting Register (SELSR0)

Address(es): ICU.SELSR0 4000 6200h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SELS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	SELS[8:0]	SYS Event Link Select	b8 b0 000000000: Disable event output to the associated low power mode module 000000001 to 111001010: Event signal number to be linked. Other settings are prohibited. For details, refer to Table 14.4, Event table.	R/W*1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This register requires halfword access.

The SELSR0 register selects the events that wake the CPU from Snooze mode. You can only use the events listed in Table 14.4 checked under "Canceling Snooze using SELSR0". Events specified in this register are defined as ICU_SNZCANCEL (02Dh) in Table 14.4. When 02Dh is set in IELSRn.IELS, the SELSR0 event interrupt occurs.

SELS[8:0] bits (SYS Event Link Select)

All SELS[8:0] bits must be written to simultaneously.

14.2.9 Wake Up Interrupt Enable Register (WUPEN)

Address(es): ICU.WUPEN 4000 61A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IIC0WUPEN	AGT1CBWUPEN	AGT1CAWUPEN	AGT1UDWUPEN	USBFSWUPEN	—	RTCPRDWUPEN	RTCALMWUPEN	ACMPLP0WUPEN	—	—	VBATTWUPEN	LVD2WUPEN	LVD1WUPEN	KEYWUPEN	IWDTWUPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit name	Description	R/W
b15 to b0	IRQWUPEN[15:0]	IRQ Interrupt Software Standby Returns Enable	0: Disable Software standby returns by IRQ interrupts 1: Enable Software standby returns by IRQ interrupts.	R/W
b16	IWDTWUPEN	IWDT Interrupt Software Standby Returns Enable	0: Disable Software standby returns by IWDT interrupts 1: Enable Software standby returns by IWDT interrupts.	R/W
b17	KEYWUPEN	Key Interrupt Software Standby Returns Enable	0: Disable Software standby returns by KEY interrupts 1: Enable Software standby returns by KEY interrupts.	R/W
b18	LVD1WUPEN	LVD1 Interrupt Software Standby Returns Enable	0: Disable Software standby returns by LVD1 interrupts 1: Enable Software standby returns by LVD1 interrupts.	R/W
b19	LVD2WUPEN	LVD2 Interrupt Software Standby Returns Enable	0: Disable Software standby returns by LVD2 interrupts 1: Enable Software standby returns by LVD2 interrupts.	R/W
b20	VBATTWUPEN	VBATT Monitor Interrupt Software Standby Returns Enable	0: Disable Software standby returns by VBATT monitor interrupt 1: Enable Software standby returns by VBATT monitor interrupt.	R/W
b22, b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23	ACMPLP0WUPEN	ACMPLP0 Interrupt Software Standby Returns Enable	0: Disable Software standby returns by ACMPLP0 interrupts 1: Enable Software standby returns by ACMPLP0 interrupts.	R/W
b24	RTCALMWUPEN	RTC Alarm Interrupt Software Standby Returns Enable	0: Disable Software standby returns by RTC alarm interrupt 1: Enable Software standby returns by RTC alarm interrupt.	R/W
b25	RTCPRDWUPEN	RTC Period Interrupt Software Standby Returns Enable	0: Disable Software standby returns by RTC period interrupts 1: Enable Software standby returns by RTC period interrupts.	R/W
b26	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b27	USBFSWUPEN	USBFS Interrupt Software Standby Returns Enable	0: Disable Software standby returns by USBFS interrupts 1: Enable Software standby returns by USBFS interrupts.	R/W
b28	AGT1UDWUPEN	AGT1 Underflow Interrupt Software Standby Returns Enable	0: Disable Software standby returns by AGT1 underflow interrupts 1: Enable Software standby returns by AGT1 underflow interrupts.	R/W
b29	AGT1CAWUPEN	AGT1 Compare Match A Interrupt Software Standby Returns Enable	0: Disable Software standby returns by AGT1 compare match A interrupts 1: Enable Software standby returns by AGT1 compare match A interrupts.	R/W
b30	AGT1CBWUPEN	AGT1 Compare Match B Interrupt Software Standby Returns Enable	0: Disable Software standby returns by AGT1 compare match B interrupt 1: Enable Software standby returns by AGT1 compare match B interrupts.	R/W

Bit	Symbol	Bit name	Description	R/W
b31	IIC0WUPEN	IIC0 Address Match Interrupt Software Standby Returns Enable	0: Disable Software standby returns by IIC0 address match interrupts 1: Enable Software standby returns by IIC0 address match interrupts.	R/W

The bits in this register control whether the associated interrupt can wake up the CPU from Software Standby mode.

IRQWUPEN[15:0] bits (IRQ Interrupt Software Standby Returns Enable)

The IRQWUPEN[15:0] bits enable the use of IRQn interrupts to cancel Software Standby mode.

IWDTWUPEN bit (IWDT Interrupt Software Standby Returns Enable)

The IWDTWUPEN bit enables the use of IWDT interrupts to cancel Software Standby mode.

KEYWUPEN bit (Key Interrupt Software Standby Returns Enable)

The KEYWUPEN bit enables the use of Key interrupts to cancel Software Standby mode.

LVD1WUPEN bit (LVD1 Interrupt Software Standby Returns Enable)

The LVD1WUPEN bit enables the use of LVD1 interrupts to cancel Software Standby mode.

LVD2WUPEN bit (LVD2 Interrupt Software Standby Returns Enable)

The LVD2WUPEN bit enables the use of LVD2 interrupts to cancel Software Standby mode.

VBATTWUPEN bit (VBATT Monitor Interrupt Software Standby Returns Enable)

The VBATTWUPEN bit enables the use of VBATT monitor interrupt to cancel Software Standby mode.

ACMPLP0WUPEN bit (ACMPLP0 Interrupt Software Standby Returns Enable)

The ACMPLP0WUPEN bit enables the use of ACMPLP0 interrupt to cancel Software Standby mode.

RTCALMWUPEN bit (RTC Alarm Interrupt Software Standby Returns Enable)

The RTCALMWUPEN bit enables the use of RTC alarm interrupts to cancel Software Standby mode.

RTCPRDWUPEN bit (RTC Period Interrupt Software Standby Returns Enable)

The RTCPRDWUPEN bit enables the use of RTC period interrupt to cancel Software Standby mode.

USBFSWUPEN bit (USBFS Interrupt Software Standby Returns Enable)

The USBFSWUPEN bit enables the use of USBFS interrupt to cancel Software Standby mode.

AGT1UDWUPEN bit (AGT1 Underflow Interrupt Software Standby Returns Enable)

The AGT1UDWUPEN bit enables the use of AGT1 underflow interrupt to cancel Software Standby mode.

AGT1CAWUPEN bit (AGT1 Compare Match A Interrupt Software Standby Returns Enable)

The AGT1CAWUPEN bit enables the use of AGT1 compare match A interrupt to cancel Software Standby mode.

AGT1CBWUPEN bit (AGT1 Compare Match B Interrupt Software Standby Returns Enable)

The AGT1CBWUPEN bit enables the use of AGT1 compare match B interrupt to cancel Software Standby mode.

IIC0WUPEN bit (IIC0 Address Match Interrupt Software Standby Returns Enable)

The IIC0WUPEN bit enables the use of IIC0 interrupt to cancel Software Standby mode.

14.3 Vector Table

The ICU detects two types of interrupts, maskable and non-maskable interrupts. Interrupt priorities are set up in the Arm NVIC. See the NVIC chapter of the *ARM® Cortex®-M4 Processor Technical Reference Manual (ARM DDI 0439D)*.

14.3.1 Interrupt Vector Table

Table 14.3 describes the interrupt vector table. The addresses conform to the NVIC specifications.

Table 14.3 Interrupt vector table (1 of 2)

Exception number	IRQ number	Vector offset	Source	Description
0	-	000h	Arm	Initial Stack Pointer
1	-	004h	Arm	Initial Program Counter (Reset Vector)
2	-	008h	Arm	Non-maskable Interrupt (NMI)
3	-	00Ch	Arm	Hard Fault
4	-	010h	Arm	MemManage Fault
5	-	014h	Arm	Bus Fault
6	-	018h	Arm	Usage Fault
7	-	01Ch	Arm	Reserved
8	-	020h	Arm	Reserved
9	-	024h	Arm	Reserved
10	-	028h	Arm	Reserved
11	-	02Ch	Arm	Supervisor call (SVCall)
12	-	030h	Arm	Debug Monitor
13	-	034h	Arm	Reserved
14	-	038h	Arm	Pendable request for system service (PendableSrvReq)
15	-	03Ch	Arm	System tick timer (SysTick)
16	0	040h	ICU.IELSR0	Event selected in the ICU.IELSR0 register
17	1	044h	ICU.IELSR1	Event selected in the ICU.IELSR1 register
18	2	048h	ICU.IELSR2	Event selected in the ICU.IELSR2 register
19	3	04Ch	ICU.IELSR3	Event selected in the ICU.IELSR3 register
20	4	050h	ICU.IELSR4	Event selected in the ICU.IELSR4 register
21	5	054h	ICU.IELSR5	Event selected in the ICU.IELSR5 register
22	6	058h	ICU.IELSR6	Event selected in the ICU.IELSR6 register
23	7	05Ch	ICU.IELSR7	Event selected in the ICU.IELSR7 register
24	8	060h	ICU.IELSR8	Event selected in the ICU.IELSR8 register
25	9	064h	ICU.IELSR9	Event selected in the ICU.IELSR9 register
26	10	068h	ICU.IELSR10	Event selected in the ICU.IELSR10 register
27	11	06Ch	ICU.IELSR11	Event selected in the ICU.IELSR11 register
28	12	070h	ICU.IELSR12	Event selected in the ICU.IELSR12 register
29	13	074h	ICU.IELSR13	Event selected in the ICU.IELSR13 register
30	14	078h	ICU.IELSR14	Event selected in the ICU.IELSR14 register
31	15	07Ch	ICU.IELSR15	Event selected in the ICU.IELSR15 register
32	16	080h	ICU.IELSR16	Event selected in the ICU.IELSR16 register
33	17	084h	ICU.IELSR17	Event selected in the ICU.IELSR17 register
34	18	088h	ICU.IELSR18	Event selected in the ICU.IELSR18 register
35	19	08Ch	ICU.IELSR19	Event selected in the ICU.IELSR19 register
36	20	090h	ICU.IELSR20	Event selected in the ICU.IELSR20 register

Table 14.3 Interrupt vector table (2 of 2)

Exception number	IRQ number	Vector offset	Source	Description
37	21	094h	ICU.IELSR21	Event selected in the ICU.IELSR21 register
38	22	098h	ICU.IELSR22	Event selected in the ICU.IELSR22 register
39	23	09Ch	ICU.IELSR23	Event selected in the ICU.IELSR23 register
40	24	0A0h	ICU.IELSR24	Event selected in the ICU.IELSR24 register
41	25	0A4h	ICU.IELSR25	Event selected in the ICU.IELSR25 register
42	26	0A8h	ICU.IELSR26	Event selected in the ICU.IELSR26 register
43	27	0ACh	ICU.IELSR27	Event selected in the ICU.IELSR27 register
44	28	0B0h	ICU.IELSR28	Event selected in the ICU.IELSR28 register
45	29	0B4h	ICU.IELSR29	Event selected in the ICU.IELSR29 register
46	30	0B8h	ICU.IELSR30	Event selected in the ICU.IELSR30 register
47	31	0BCh	ICU.IELSR31	Event selected in the ICU.IELSR31 register
48	32	0C0h	ICU.IELSR32	Event selected in the ICU.IELSR32 register
49	33	0C4h	ICU.IELSR33	Event selected in the ICU.IELSR33 register
50	34	0C8h	ICU.IELSR34	Event selected in the ICU.IELSR34 register
51	35	0CCh	ICU.IELSR35	Event selected in the ICU.IELSR35 register
52	36	0D0h	ICU.IELSR36	Event selected in the ICU.IELSR36 register
53	37	0D4h	ICU.IELSR37	Event selected in the ICU.IELSR37 register
54	38	0D8h	ICU.IELSR38	Event selected in the ICU.IELSR38 register
55	39	0DCh	ICU.IELSR39	Event selected in the ICU.IELSR39 register
56	40	0E0h	ICU.IELSR40	Event selected in the ICU.IELSR40 register
57	41	0E4h	ICU.IELSR41	Event selected in the ICU.IELSR41 register
58	42	0E8h	ICU.IELSR42	Event selected in the ICU.IELSR42 register
59	43	0ECh	ICU.IELSR43	Event selected in the ICU.IELSR43 register
60	44	0F0h	ICU.IELSR44	Event selected in the ICU.IELSR44 register
61	45	0F4h	ICU.IELSR45	Event selected in the ICU.IELSR45 register
62	46	0F8h	ICU.IELSR46	Event selected in the ICU.IELSR46 register
63	47	0FCh	ICU.IELSR47	Event selected in the ICU.IELSR47 register
64	48	100h	ICU.IELSR48	Event selected in the ICU.IELSR48 register
65	49	104h	ICU.IELSR49	Event selected in the ICU.IELSR49 register
66	50	108h	ICU.IELSR50	Event selected in the ICU.IELSR50 register
67	51	10Ch	ICU.IELSR51	Event selected in the ICU.IELSR51 register
68	52	110h	ICU.IELSR52	Event selected in the ICU.IELSR52 register
69	53	114h	ICU.IELSR53	Event selected in the ICU.IELSR53 register
70	54	118h	ICU.IELSR54	Event selected in the ICU.IELSR54 register
71	55	11Ch	ICU.IELSR55	Event selected in the ICU.IELSR55 register
72	56	120h	ICU.IELSR56	Event selected in the ICU.IELSR56 register
73	57	124h	ICU.IELSR57	Event selected in the ICU.IELSR57 register
74	58	128h	ICU.IELSR58	Event selected in the ICU.IELSR58 register
75	59	12Ch	ICU.IELSR59	Event selected in the ICU.IELSR59 register
76	60	130h	ICU.IELSR60	Event selected in the ICU.IELSR60 register
77	61	134h	ICU.IELSR61	Event selected in the ICU.IELSR61 register
78	62	138h	ICU.IELSR62	Event selected in the ICU.IELSR62 register
79	63	13Ch	ICU.IELSR63	Event selected in the ICU.IELSR63 register

14.3.2 Event Number

The following table lists heading details for Table 14.4, which describes each event number.

Heading	Description
Interrupt request source	Name of the source generating the interrupt request
Name	Name of the interrupt
Form of interrupt detection (signal)	"Edge" or "level" as the method for detection of the interrupt. "✓" indicates usability as anNMI interrupt.
Connect to NVIC	"✓" indicates that the interrupt can be used as a CPU interrupt (IELSRn setting)
Invoke DTC	"✓" indicates that the interrupt can be used to request DTC activation (IELSRn setting)
Invoke DMAC	"✓" indicates that the interrupt can be used to request DMAC activation (DELSRn setting)
Canceling Snooze mode	"✓" indicates that the interrupt can be used to request a return from Snooze mode using SELSR0. Otherwise, "✓" indicates that it can be used directly.
Canceling Software Standby mode	"✓" indicates that the interrupt can be used to request a return from Software Standby mode

Table 14.4 Event table (1 of 6)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC		
001h	Port	PORT_IRQ0	✓	✓	✓	✓	✓
002h		PORT_IRQ1	✓	✓	✓	✓	✓
003h		PORT_IRQ2	✓	✓	✓	✓	✓
004h		PORT_IRQ3	✓	✓	✓	✓	✓
005h		PORT_IRQ4	✓	✓	✓	✓	✓
006h		PORT_IRQ5	✓	✓	✓	✓	✓
007h		PORT_IRQ6	✓	✓	✓	✓	✓
008h		PORT_IRQ7	✓	✓	✓	✓	✓
009h		PORT_IRQ8	✓	✓	✓	✓	✓
00Ah		PORT_IRQ9	✓	✓	✓	✓	✓
00Bh		PORT_IRQ10	✓	✓	✓	✓	✓
00Ch		PORT_IRQ11	✓	✓	✓	✓	✓
00Dh		PORT_IRQ12	✓	✓	✓	✓	✓
00Eh		PORT_IRQ13	✓	✓	✓	✓	✓
00Fh		PORT_IRQ14	✓	✓	✓	✓	✓
010h		PORT_IRQ15	✓	✓	✓	✓	✓
020h	DMAC0	DMAC0_INT	✓	✓			
021h	DMAC1	DMAC1_INT	✓	✓			
022h	DMAC2	DMAC2_INT	✓	✓			
023h	DMAC3	DMAC3_INT	✓	✓			
029h	DTC	DTC_COMPLETE	✓			✓*4	
02Dh	ICU	ICU_SNZCANCEL	✓			✓	
031h	FCU	FCU_FRDYI	✓				
038h	LVD	LVD_LVD1	✓			✓	✓
039h		LVD_LVD2	✓			✓	✓
03Ah	VBATT	VBATT_LVD	✓			✓	✓
03Bh	MOSC	MOSC_STOP	✓				

Table 14.4 Event table (2 of 6)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC		
03Ch	Low power mode	SYSTEM_SNZREQ		✓			
040h	AGT0	AGT0_AGTI	✓	✓	✓		
041h		AGT0_AGTCMAI	✓	✓	✓		
042h		AGT0_AGTCMBI	✓	✓	✓		
043h	AGT1	AGT1_AGTI	✓	✓	✓	✓	✓
044h		AGT1_AGTCMAI	✓	✓	✓	✓	✓
045h		AGT1_AGTCMBI	✓	✓	✓	✓	✓
046h	IWDT	IWDT_NMIUNDF	✓			✓	✓
047h	WDT	WDT_NMIUNDF	✓				
048h	RTC	RTC_ALM	✓			✓	✓
049h		RTC_PRD	✓			✓	✓
04Ah		RTC_CUP	✓				
04Bh	ADC140	ADC140_ADI	✓	✓	✓		
04Ch		ADC140_GBADI	✓	✓	✓		
04Dh		ADC140_CMPAI	✓				
04Eh		ADC140_CMPBI	✓				
04Fh		ADC140_WCMPM		✓	✓	✓*4	
050h		ADC140_WCMPUM		✓	✓	✓*4	
057h	ACMPHS	ACMP_HS0	✓				
058h		ACMP_HS1	✓				
05Dh	ACMPLP	ACMP_LP0	✓			✓	✓
05Eh		ACMP_LP1	✓				
05Fh	USBFS	USBFS_D0FIFO	✓	✓	✓		
060h		USBFS_D1FIFO	✓	✓	✓		
061h		USBFS_USBI	✓				
062h		USBFS_USBR	✓			✓	✓
063h	IIC0	IIC0_RXI	✓	✓	✓		
064h		IIC0_TXI	✓	✓	✓		
065h		IIC0_TEI	✓				
066h		IIC0_EEI	✓				
067h		IIC0_WUI	✓				✓
068h	IIC1	IIC1_RXI	✓	✓	✓		
069h		IIC1_TXI	✓	✓	✓		
06Ah		IIC1_TEI	✓				
06Bh		IIC1_EEI	✓				
06Dh	IIC2	IIC2_RXI	✓	✓	✓		
06Eh		IIC2_TXI	✓	✓	✓		
06Fh		IIC2_TEI	✓				
070h		IIC2_EEI	✓				
072h	SSI0	SSI0_SSITXI	✓	✓	✓		
073h		SSI0_SSIRXI	✓	✓	✓		
075h		SSI0_SSIF	✓				
078h	SSI1	SSI1_SSIRT	✓	✓	✓		
079h		SSI1_SSIF	✓				

Table 14.4 Event table (3 of 6)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC		
082h	CTSU	CTSU_CTSUWR	✓	✓	✓		
083h		CTSU_CTSURD	✓	✓	✓		
084h		CTSU_CTSUFN	✓			✓*4	
085h	KINT	KEY_INTKR	✓			✓*1	✓*1
086h	DOC	DOC_DOPCI	✓			✓*4	
087h	CAC	CAC_FERRI	✓				
088h		CAC_MENDI	✓				
089h		CAC_OVFI	✓				
08Ah	CAN0	CAN0_ERS	✓				
08Bh		CAN0_RXF	✓				
08Ch		CAN0_TXF	✓				
08Dh		CAN0_RXM	✓				
08Eh		CAN0_TXM	✓				
094h	I/O Port	IOPORT_GROUP1	✓	✓*2	✓*2		
095h		IOPORT_GROUP2	✓	✓*2	✓*2		
096h		IOPORT_GROUP3	✓	✓*2	✓*2		
097h		IOPORT_GROUP4	✓	✓*2	✓*2		
098h	ELC	ELC_SWEVT0	✓*3	✓			
099h		ELC_SWEVT1	✓*3	✓			
09Ah	POEG	POEG_GROUP0	✓				
09Bh		POEG_GROUP1	✓				
09Ch		POEG_GROUP2	✓				
09Dh		POEG_GROUP3	✓				
0B0h	GPT320	GPT0_CCMPA	✓	✓	✓		
0B1h		GPT0_CCMPB	✓	✓	✓		
0B2h		GPT0_CMPC	✓	✓	✓		
0B3h		GPT0_CMPD	✓	✓	✓		
0B4h		GPT0_CMPE	✓	✓	✓		
0B5h		GPT0_CMPF	✓	✓	✓		
0B6h		GPT0_OVF	✓	✓	✓		
0B7h		GPT0_UDF	✓	✓	✓		
0BAh	GPT321	GPT1_CCMPA	✓	✓	✓		
0BBh		GPT1_CCMPB	✓	✓	✓		
0BCh		GPT1_CMPC	✓	✓	✓		
0BDh		GPT1_CMPD	✓	✓	✓		
0BEh		GPT1_CMPE	✓	✓	✓		
0BFh		GPT1_CMPF	✓	✓	✓		
0C0h		GPT1_OVF	✓	✓	✓		
0C1h		GPT1_UDF	✓	✓	✓		

Table 14.4 Event table (4 of 6)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC		
0C4h	GPT322	GPT2_CCMPA	✓	✓	✓		
0C5h		GPT2_CCMPB	✓	✓	✓		
0C6h		GPT2_CMPC	✓	✓	✓		
0C7h		GPT2_CMPD	✓	✓	✓		
0C8h		GPT2_CMPE	✓	✓	✓		
0C9h		GPT2_CMPF	✓	✓	✓		
0CAh		GPT2_OVF	✓	✓	✓		
0CBh		GPT2_UDF	✓	✓	✓		
0CEh	GPT323	GPT3_CCMPA	✓	✓	✓		
0CFh		GPT3_CCMPB	✓	✓	✓		
0D0h		GPT3_CMPC	✓	✓	✓		
0D1h		GPT3_CMPD	✓	✓	✓		
0D2h		GPT3_CMPE	✓	✓	✓		
0D3h		GPT3_CMPF	✓	✓	✓		
0D4h		GPT3_OVF	✓	✓	✓		
0D5h		GPT3_UDF	✓	✓	✓		
0D8h	GPT324	GPT4_CCMPA	✓	✓	✓		
0D9h		GPT4_CCMPB	✓	✓	✓		
0DAh		GPT4_CMPC	✓	✓	✓		
0DBh		GPT4_CMPD	✓	✓	✓		
0DCh		GPT4_CMPE	✓	✓	✓		
0DDh		GPT4_CMPF	✓	✓	✓		
0DEh		GPT4_OVF	✓	✓	✓		
0DFh		GPT4_UDF	✓	✓	✓		
0E2h	GPT325	GPT5_CCMPA	✓	✓	✓		
0E3h		GPT5_CCMPB	✓	✓	✓		
0E4h		GPT5_CMPC	✓	✓	✓		
0E5h		GPT5_CMPD	✓	✓	✓		
0E6h		GPT5_CMPE	✓	✓	✓		
0E7h		GPT5_CMPF	✓	✓	✓		
0E8h		GPT5_OVF	✓	✓	✓		
0E9h		GPT5_UDF	✓	✓	✓		
0ECh	GPT326	GPT6_CCMPA	✓	✓	✓		
0EDh		GPT6_CCMPB	✓	✓	✓		
0EEh		GPT6_CMPC	✓	✓	✓		
0EFh		GPT6_CMPD	✓	✓	✓		
0F0h		GPT6_CMPE	✓	✓	✓		
0F1h		GPT6_CMPF	✓	✓	✓		
0F2h		GPT6_OVF	✓	✓	✓		
0F3h		GPT6_UDF	✓	✓	✓		

Table 14.4 Event table (5 of 6)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC		
0F6h	GPT327	GPT7_CCMPA	✓	✓	✓		
0F7h		GPT7_CCMPB	✓	✓	✓		
0F8h		GPT7_CMPC	✓	✓	✓		
0F9h		GPT7_CMPD	✓	✓	✓		
0FAh		GPT7_CMPE	✓	✓	✓		
0FBh		GPT7_CMPF	✓	✓	✓		
0FCh		GPT7_OVF	✓	✓	✓		
0FDh		GPT7_UDF	✓	✓	✓		
100h	GPT328	GPT8_CCMPA	✓	✓	✓		
101h		GPT8_CCMPB	✓	✓	✓		
102h		GPT8_CMPC	✓	✓	✓		
103h		GPT8_CMPD	✓	✓	✓		
104h		GPT8_CMPE	✓	✓	✓		
105h		GPT8_CMPF	✓	✓	✓		
106h		GPT8_OVF	✓	✓	✓		
107h		GPT8_UDF	✓	✓	✓		
10Ah	GPT329	GPT9_CCMPA	✓	✓	✓		
10Bh		GPT9_CCMPB	✓	✓	✓		
10Ch		GPT9_CMPC	✓	✓	✓		
10Dh		GPT9_CMPD	✓	✓	✓		
10Eh		GPT9_CMPE	✓	✓	✓		
10Fh		GPT9_CMPF	✓	✓	✓		
110h		GPT9_OVF	✓	✓	✓		
111h		GPT9_UDF	✓	✓	✓		
150h	GPT	GPT_UVWEDGE	✓				
174h	SCI0	SCI0_RXI	✓	✓	✓		
175h		SCI0_TXI	✓	✓	✓		
176h		SCI0_TEI	✓				
177h		SCI0_ERI	✓				
178h		SCI0_AM	✓			✓*4	
179h		SCI0_RXI_OR_ERI				✓*4	
17Ah	SCI1	SCI1_RXI	✓	✓	✓		
17Bh		SCI1_TXI	✓	✓	✓		
17Ch		SCI1_TEI	✓				
17Dh		SCI1_ERI	✓				
17Eh		SCI1_AM	✓				
180h	SCI2	SCI2_RXI	✓	✓	✓		
181h		SCI2_TXI	✓	✓	✓		
182h		SCI2_TEI	✓				
183h		SCI2_ERI	✓				
184h		SCI2_AM	✓				

Table 14.4 Event table (6 of 6)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling Snooze	Canceling Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC		
186h	SCI3	SCI3_RXI	✓	✓	✓		
187h		SCI3_TXI	✓	✓	✓		
188h		SCI3_TEI	✓				
189h		SCI3_ERI	✓				
18Ah		SCI3_AM	✓				
18Ch	SCI4	SCI4_RXI	✓	✓	✓		
18Dh		SCI4_TXI	✓	✓	✓		
18Eh		SCI4_TEI	✓				
18Fh		SCI4_ERI	✓				
190h		SCI4_AM	✓				
1AAh	SCI9	SCI9_RXI	✓	✓	✓		
1ABh		SCI9_TXI	✓	✓	✓		
1ACh		SCI9_TEI	✓				
1ADh		SCI9_ERI	✓				
1AEh		SCI9_AM	✓				
1BCh	SPI0	SPI0_SPRI	✓	✓	✓		
1BDh		SPI0_SPTI	✓	✓	✓		
1BEh		SPI0_SPII	✓				
1BFh		SPI0_SPEI	✓				
1C0h		SPI0_SPTEND	✓				
1C1h	SPI1	SPI1_SPRI	✓	✓	✓		
1C2h		SPI1_SPTI	✓	✓	✓		
1C3h		SPI1_SPII	✓				
1C4h		SPI1_SPEI	✓				
1C5h		SPI1_SPTEND	✓				
1C6h	QSPI	QSPI_INTR	✓				
1C7h	SDHI0	SDHI_MMC0_ACCS	✓				
1C8h		SDHI_MMC0_SDIO	✓				
1C9h		SDHI_MMC0_CARD	✓				
1CAh		SDHI_MMC0_ODMSDBREQ		✓	✓		

Note 1. Only supported when KRCTL.KRMD = 1.
Note 2. Only the first edge detection is valid.
Note 3. Only interrupts after DTC transfer are supported.
Note 4. Using SELSR0.

14.4 Interrupt Operation

The ICU performs the following functions:

- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations such as CPU interrupt, DTC activation, or DMAC activation.

14.4.1 Detecting Interrupts

External pin interrupt requests are detected by either the edge or level (falling edge, rising edge, rising and falling edge, or low level) of the interrupt signal. Set the IRQMD[1:0] bits in the IRQCRi register to select the detection mode for the IRQi pins. For interrupt sources associated with peripheral modules, see section 14.3.2, Event Number. Events must be accepted by the NVIC before an interrupt occurs and is accepted by the CPU.

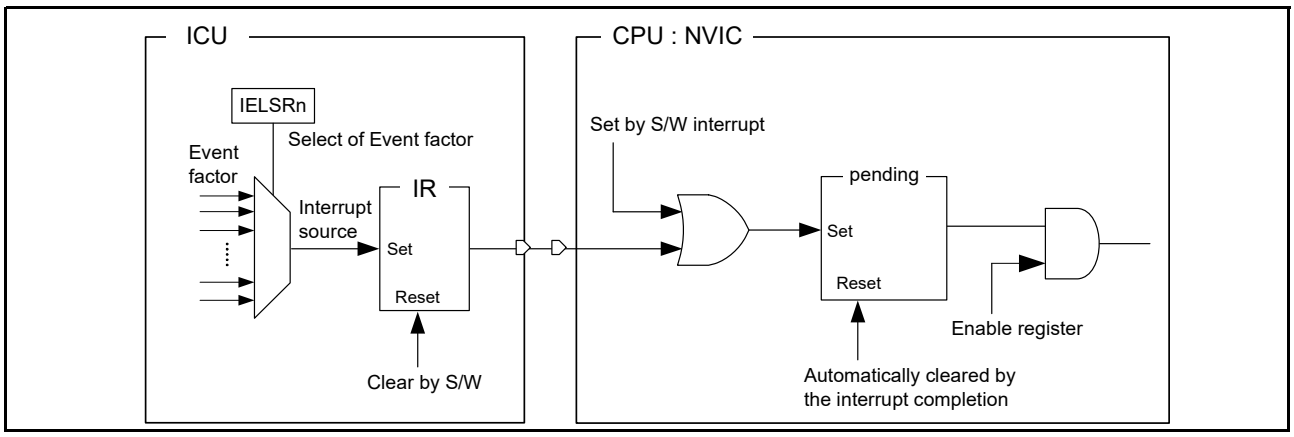


Figure 14.2 Interrupt path of the ICU and CPU: NVIC

- General operations during an interrupt
  - When a non-software interrupt occurs:
    The IELSRn.IR flag and Interrupt Set/Clear-Pending register (NVIC) are set.
  - When a software interrupt occurs:
    Set the Interrupt Set-Pending register.
  - When an interrupt is complete:
    Clear the IELSRn.IR flag in the software.
    The Interrupt Set/Clear-Pending register clears automatically.
- When interrupts are enabled:
  1) Set the Interrupt Set-Enable register
  2) Set the IELSRn.IELS bits as the interrupt source
  3) Specify the operation settings for the event source
- When interrupts are disabled:
  1) Disable the settings for the event source
  2) Clear the IELSRn.IELS bits (IELSRn.IELS[8:0] = 000h). Clear the IELSRn.IR flag as required
  3) Clear the Interrupt Clear-Enable register. Clear the Interrupt Clear-Pending register as required
- When polling for interrupts:
  1) Set the Interrupt Clear-Enable register (disabling interrupts)
  2) Set the IELSRn.IELS bits (selecting the source)
  3) Specify the operation settings for the event source
  4) Poll the Interrupt Set-pending register
  5) When polling is no longer required, follow the procedure for clearing an interrupt when it is complete

14.4.2 Selecting Interrupt Request Destinations

The interrupt output destination, CPU, DTC or DMAC, can be independently selected for each interrupt source. The available destinations are fixed for each interrupt, as described in Table 14.4, Event table.

Note: Do not use an interrupt request destination setting that is not indicated by a ✓ in the event list (Table 14.4).

If you select the CPU or DTC in one IELSRn register, setting the same interrupt factor in any other IELSRn register is prohibited. Similarly, if you select the DMAC in one DELSRn register, setting the same interrupt factor in any other DELSRn register is prohibited.

Note: Setting the same interrupt factor for IELSRn and DELSRn is prohibited.

If the DMAC or DTC is selected as the destination for requests from an IRQi pin, you must set the IRQMD[1:0] bits in IRQCRi for that interrupt to select edge detection.

14.4.2.1 CPU Interrupt Request

When IELSRn.DTCE = 0, the event specified in the IELSRn register is output to the NVIC. Use the following procedure:

Set the IELSRn.IELS and IELSRn.DTCE bits to 0.

14.4.2.2 DTC Activation

When IELSRn.DTCE = 1, the event specified in the IELSRn register is output to the DTC. After DTC transmission completes, the associated interrupt occurs. Use the following procedure:

1. Set the IELSRn.IELS bits to the target event and the IELSRn.DTCE bit to 1.
2. Set the DTC module start bit DTCST.DTCST to 1.

Table 14.5 shows operation when the DTC is the request destination.

Table 14.5 Operations when DTC is activated

Interrupt request destination	DISEL*1	Remaining transfer operations	Operations 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	The IELSRn.DTCE bit is cleared and the CPU becomes the destination
	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	The IELSRn.DTCE bit is cleared and the CPU becomes the destination

Note 1. Set the interrupt request mode for the DTC in the DTC.MRB.DISEL bit.
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. At this point, 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 18.3, Chain transfer conditions in section 18, Data Transfer Controller (DTC).

14.4.2.3 DMAC Activation

When IELSRn.DTCE = 0, the event specified in the IELSRn register is output to the NVIC. To set the interrupt source for DMAC, use the following procedure:

1. Set the DELSRn.DELS[8:0].
2. Set the IELSRn.IELS bits to the target event and the IELSRn.DTCE bit to 1.
3. Set the activation source for the target DMAC channel (DMACm.DMTMD.DCTG[1:0]) to 01b (interrupt module detection).
4. Set the DMAC transfer enable bit for the target DMAC channel (DMACm.DMCNT.DTE) to 1.
5. Set the DMAC operation enable bit (DMAST.DMST) to 1.

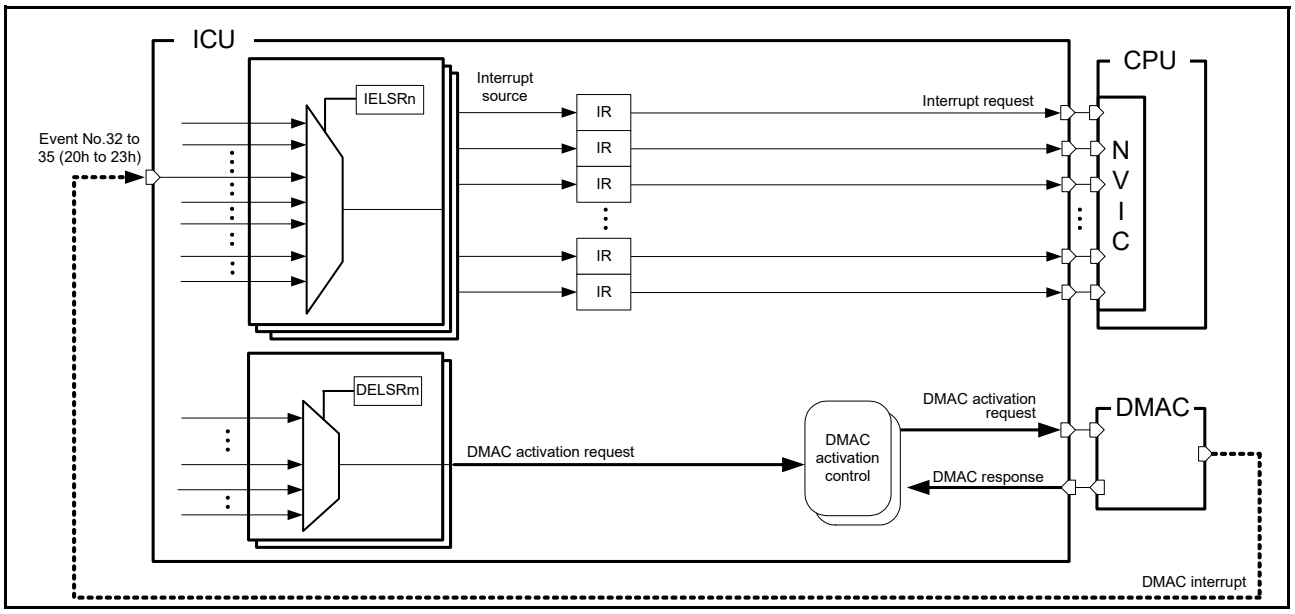


Figure 14.3 DMAC request trigger and interrupt path

14.4.3 Digital Filter

A digital filter function is provided for the external interrupt request pins (IRQi, i = 0 to 15) and NMI pin interrupt. It samples input signals on the filter sampling clock (PCLKB) and removes any signal with a pulse width less than three sampling cycles.

- To use the digital filter for a 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 15)
2)	Set the IRQCRi.FLTEN bit (i = 0 to 15) to 1 (digital filter enabled)

- To use the digital filter for the 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 14.4 shows an example of digital filter operation.

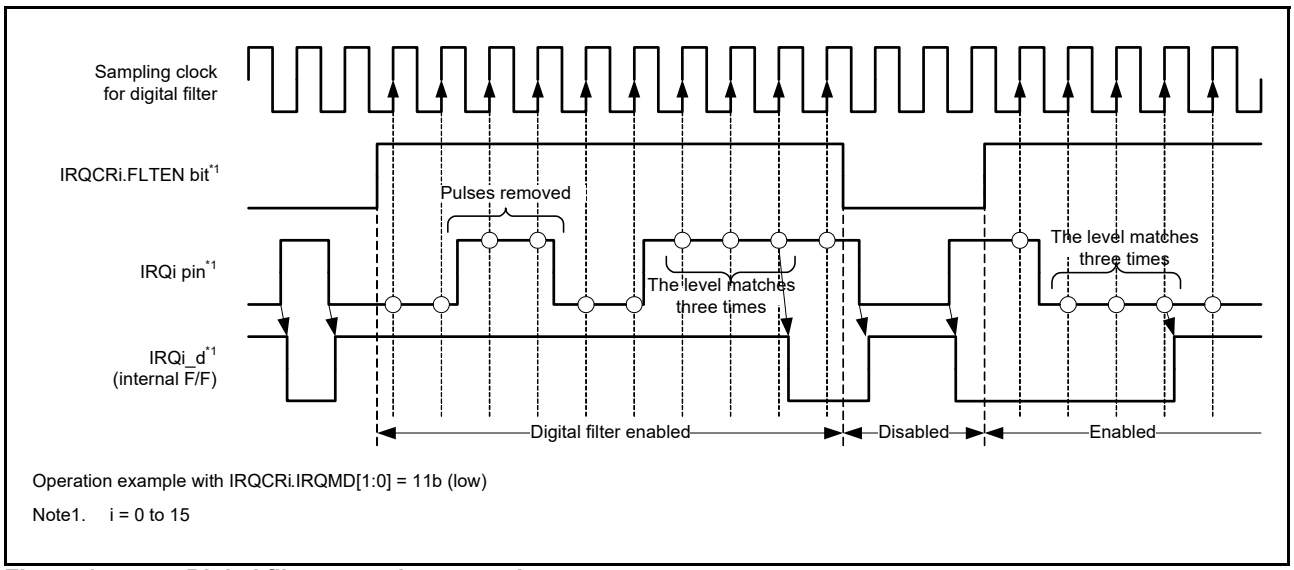


Figure 14.4 Digital filter operation example

Before entering Software Standby mode, disable the digital filters by clearing the IRQCRi.FLTEN and NMICR.NFLTEN bits. The clock for the ICU stops in Software Standby. On exiting Software Standby, the circuit detects the edge by comparing the state before standby to the state after standby release. If the input changes during Software Standby, an incorrect edge might be detected. You can enable the digital filters again after exiting Software Standby mode.

14.4.4 External Pin Interrupts

To use external pin interrupts:

1. Clear the IRQCRi.FLTEN bit (i = 0 to 15) to 0 (digital filter disabled).
2. Make or confirm the I/O port settings.
3. Set the IRQMD[1:0] bits, FCLKSEL[1:0] bits and FLTEN bit of IRQCRi register.
4. If the IRQ pin is to be used for CPU interrupt request, set the IELSRn.IELS[8:0] bits and IELSRn.DTCE bit to 0.
   If the IRQ pin is to be used for DTC activation, set the IELSRn.IELS[8:0] bits and IELSRn.DTCE bit to 1.
   If the IRQ pin is to be used for DMAC activation, set the DELSRn.DELS bits.

14.5 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
- VBATT monitor interrupt
- SRAM parity error interrupt
- SRAM ECC error interrupt
- MPU bus master error interrupt
- MPU bus slave error interrupt
- CPU stack pointer monitor interrupt.

Non-maskable interrupts can only be used with the CPU, not to activate the DTC or DMAC. 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, use the following procedure:

To use the NMI pin, follow steps 1 to 3.

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. When an NMI interrupt is enabled, it can be disabled only by a reset.

14.6 Return from Low Power Mode

Table 14.4, Event table lists the interrupt sources you can use to exit Sleep or Software Standby mode. For details, see section 11, Low Power Modes. Sections 14.6.1 to 14.6.3 describe how to use interrupts to recover from Sleep, Software Standby, and Snooze modes.

14.6.1 Return from Sleep mode

To return from Sleep mode in response to an interrupt:

1. Select the CPU as the interrupt request destination.
2. Enable the interrupt in the NVIC.

To return from Sleep mode in response to a non-maskable interrupt, use the NMIER register to enable the given interrupt request.

14.6.2 Return from Software Standby mode

The ICU can return from Software Standby mode using a non-maskable interrupt or an interrupt selected in the WUPEN register. See section 14.2.9, Wake Up Interrupt Enable Register (WUPEN).

To return from Software Standby mode, you must:

1. Select the interrupt source that enables return from Software Standby.

- For non-maskable interrupts, use the NMIER register to enable the wanted interrupt request
- For maskable interrupts, use the WUPEN register to enable the wanted interrupt request.

2. Select the CPU as the interrupt request destination.
3. Enable the interrupt in the NVIC.

Interrupt requests through the IRQ pins that do not satisfy these conditions are not detected while the clock is stopped in Software Standby mode.

14.6.3 Return from Snooze mode

The ICU can return from Snooze mode using the interrupts provided for this mode. Select the interrupt source in the SELSR0 register.

To return to Normal mode from Snooze mode:

1. Use either of the following methods to select the event that you want to trigger a return from Snooze mode to Normal mode:

   - Set the event that you want to trigger a return from Snooze mode to Normal mode in SELSR0.SEL and set the value 02Dh (ICU_SNZCANCEL) in IELSRn.IELS.
   - Set the event that you want to trigger a return from Snooze mode to Normal mode in IELSRn.IELS.

2. Select the CPU as the interrupt request destination.
3. Enable the interrupt in the NVIC.

Note: In Snooze mode, a clock is supplied to ICU. If an event selected in IELSRn is detected, the CPU can acknowledge the interrupt after returning to Normal mode from Software Standby mode. If an event selected in DELSRn is detected, the DMAC can acknowledge the interrupt after returning to Normal mode from Software Standby mode.

14.7 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.

14.8 Reference

ARM® Cortex®-M4 Processor Technical Reference Manual (ARM DDI 0439D).

15. Buses

15.1 Overview

Table 15.1 lists the bus specifications, Figure 15.1 shows the bus configuration, and Table 15.2 lists the addresses assigned for each bus.

Table 15.1 Bus specifications

Bus Type		Description
Main bus	ICode bus (CPU)	• Connected to the CPU • Connected to on-chip memory (code flash memory).
	DCode bus (CPU)	• Connected to the CPU • Connected to on-chip memory (code flash memory).
	System bus (CPU)	• Connected to the CPU • Connected to on-chip memory, internal peripheral bus, and external bus.
	DMA bus	• Connected to the DMAC or DTC • Connected to on-chip memory, internal peripheral bus, and external bus.
Slave interface	Memory bus 1	• Connected to code flash memory
	Memory bus 3	• Connected to code flash memory by DMA bus
	Memory bus 4	• Connected to SRAM0
	Memory bus 5	• Connected to SRAM1
	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, RTC, WDT, IWDT, IIC, CAN, SSI, ADC14, DAC12, and DOC)
	Internal peripheral bus 4	• Connected to peripheral modules (SCI, IrDA, SPI, CRC, GPT, and SDHI)
	Internal peripheral bus 5	• Connected to peripheral modules (KINT, AGT, USBFS, OPAMP, ACMPHS, ACMPLP, SLCDC, and CTSU)
	Internal peripheral bus 7	• Connected to Secure IPs
	Internal peripheral bus 9	• Connected to flash memory (in P/E)*1 and data flash memory
External bus	CS area	• Connected to the external devices
	QSPI area	• Connected to the external SPI devices

Note 1. P/E = Programming/Erasure.

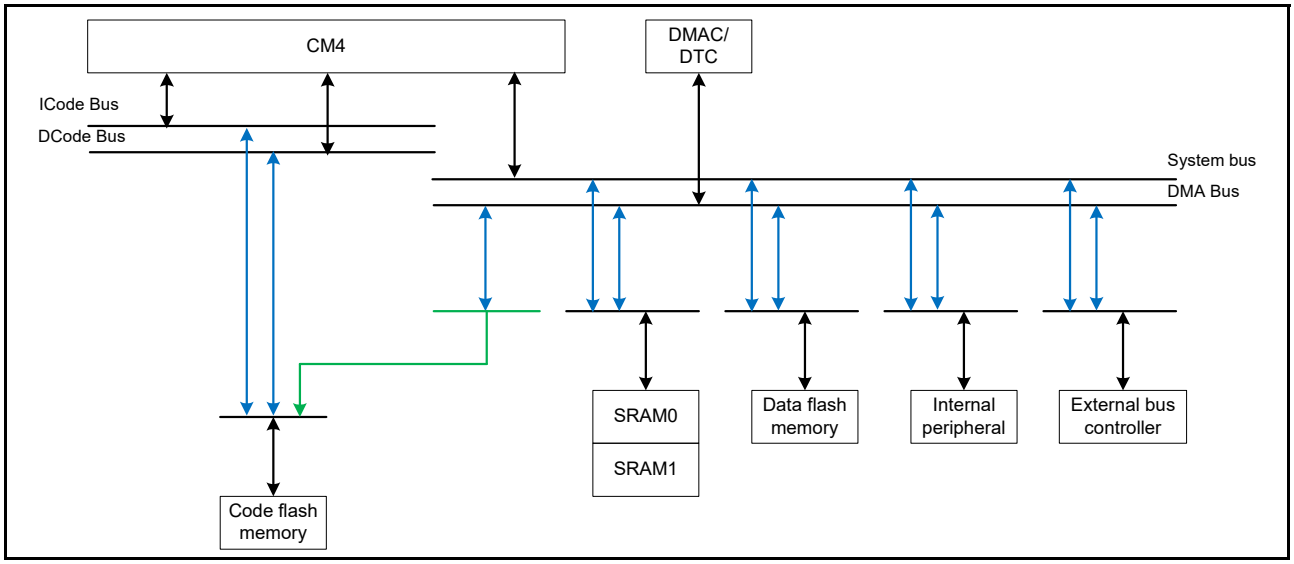


Figure 15.1 Bus configuration

Table 15.2 Addresses assigned for each bus

Address	Bus	Area
0000 0000h to 01FF FFFFh	Memory bus 1, 3	Code flash memory
2000 0000h to 2001 FFFFh	Memory bus 4	SRAM0
2002 0000h to 2002 FFFFh	Memory bus 5	SRAM1
4000 0000h to 4001 FFFFh	Internal peripheral bus 1	Peripheral I/O registers
4004 0000h to 4005 FFFFh	Internal peripheral bus 3	
4006 0000h to 4007 FFFFh	Internal peripheral bus 4	
4008 0000h to 4009 FFFFh	Internal peripheral bus 5	
400C 0000h to 400D FFFFh	Internal peripheral bus 7	Secure IPs
4010 0000h to 407F FFFFh	Internal peripheral bus 9	Flash memory (in P/E)*1 and data flash memory
6000 0000h to 67FF FFFFh	External bus	QSPI area
8000 0000h to 97FF FFFFh	External bus	CS area

Note 1. P/E = Programming/Erasure.

15.2 Description of Buses

15.2.1 Main Buses

The main bus for the CPU consists of the ICode bus, DCode bus, and system bus.

- The ICode bus and DCode bus are connected to code flash memory. The ICode bus is used for instruction access to the CPU and the DCode bus is used for data access to the CPU.
- The system bus is connected to SRAM0, SRAM1, data flash memory, internal peripheral bus, and external bus. The system bus is used for instruction and data access to the CPU.

The main bus for modules other than the CPU consists of the DMA bus. The DMA bus is connected to the code flash memory, SRAM0, SRAM1, data flash memory, internal peripheral bus, and external bus.

Different master and slave transfer combinations can proceed simultaneously.

Arbitration between DMAC and DTC for the mastership of the DMA bus occurs in the DMAC and DTC. The following fixed-priority order is used:

DMAC0, DMAC1, DMAC2, DMAC3, and DTC.

Only one DTC and DMAC channels that have accepted the activation requests can issue the bus mastership request. In addition, requests for bus access from masters other than the DTC are not accepted during reads of transfer control information for the DTC.

15.2.2 Slave Interface

Products using the Cortex®-M4 core contain ICode and DCode bus areas and a system bus area. To create the ICode and DCode bus areas, a bus matrix connects the ICode bus, DCode bus, and Memory bus 3 from the main bus to the slave interface of the code flash memory. To create a system bus area, a bus matrix connects the system bus and DMA bus from the main bus to the slave interfaces of SRAM0, SRAM1, data flash memory, internal peripheral, and external bus. For connections from the main bus to the slave interfaces, see the slave interfaces in Table 15.1. For a description of the external bus, see section 15.2.3, External Bus.

Arbitration between the ICode bus, DCode bus, and Memory bus 3 occurs in the slave interface of the ICode and DCode bus areas. The arbitration method is selectable as either fixed priority or round-robin. For more information, see section 15.3.8, Slave Bus Control Register (BUSSCNT<slave>).

Arbitration between the system bus and DMA bus occurs in the slave interface of the system bus area. The arbitration method is selectable as either fixed priority or round-robin. For more information, see section 15.3.8.

Different master and slave transfer combinations can proceed simultaneously.

15.2.3 External Bus

Table 15.3 lists the external bus specifications. The external bus controller arbitrates requests for bus access on the external address space from the CPU system bus and the DMA bus. The priority order can be set using the external bus priority control bits (BUSSCNT.ARBMET). For more information, see section 15, Slave Bus Control Register (BUSSCNT<slave>).

The bus system provides an external space for QSPI. See section 34, Quad Serial Peripheral Interface (QSPI).

Table 15.3 lists the external bus specifications and Table 15.4 lists the I/O pins.

Table 15.3 External bus specifications

Parameter	Description
External address space	• The external address space is divided into 4 CS areas (CS0 to CS3) for management • Chip select signals can be output for each area • The bus width can be set for each area: - Separate bus: Selectable to 8-bit or 16-bit bus space. • Endian mode can be specified for each area.
CS area controller	• Recovery cycles can be inserted: - Read recovery: up to 15 cycles - Write recovery: up to 15 cycles. • Cycle wait function: wait for up to 31 cycles (page access: up to 7 cycles) • Wait control can be used to set up the following: - Assertion and negation timing of chip select signals (CS0 to CS3) - Assertion timing of the read signal (RD) and write signals (WR0/WR and WR1) - Timing of data output starts and ends. • Write access mode: - Single write strobe mode/byte strobe mode.
Write buffer function	When write data from the bus master is written to the write buffer, write access by the bus master is complete.
Frequency	• The CS area controller (CSC) operates in synchronization with the external bus clock (BCLK). • The frequency of the EBCLK pin output is the same as BCLK by default. Half of the BCLK clock cycle can be supplied by setting the EBCLK Pin Output Select bit, BCKCR.BCLKDIV, in the External Bus Clock Control Register. For more information, see section 9, Clock Generation Circuit.

Table 15.4 External pin configurations (1 of 2)

Pin name	I/O	Description
EBCLK	Output	Clock output pin
A16 to A00*1	Output	Address output pins
D15 to D00	I/O	Data input/output pins: • D15 to D00 pins are enabled when the 16-bit bus space is specified • D07 to D00 pins are enabled when the 8-bit bus space is specified.
BC0*1	Output	Strobe signal (when low) indicates that D07 to D00 are valid during access to an external address space in single write strobe mode, active-low. When the 8-bit bus space is specified, this output pin is always held low regardless of the write access mode.
BC1	Output	Strobe signal (when low) indicates that D15 to D08 are valid during access to an external address space in single write strobe mode, active-low. This pin is not used when the 8-bit bus space is specified.
CS0	Output	Chip select signal for area 0 (CS0), active-low.
CS1	Output	Chip select signal for area 1 (CS1), active-low.
CS2	Output	Chip select signal for area 2 (CS2), active-low.
CS3	Output	Chip select signal for area 3 (CS3), active-low.
RD	Output	Strobe signal indicates that a read from an external address space (CS0 to CS3) is in progress, active-low.

Table 15.4 External pin configurations (2 of 2)

Pin name	I/O	Description
WR0/WR*2	Output	WR0 signal is a strobe signal that indicates (when low) a write to an external address space is in progress in byte strobe mode, and D07 to D00 are valid, active-low. WR signal is a strobe signal that indicates that a write to an external address space is in progress in single write strobe mode, active-low. When an 8-bit bus space is specified, this output pin is held low during a write access regardless of write access mode.
WR1	Output	Strobe signal (when low) indicates that D15 to D08 are valid during a write to an external address space in byte strobe mode, active-low. This signal is invalid in single write strobe mode. This pin is not used when the 8-bit bus space is specified.
WAIT	Input	Wait request signal (when low) when accessing the external address space (CS0 to CS3), active-low.

Note 1. The A00 and BC0 pin functions share the same pin, and either becomes effective according to the area. The function is A00 in byte strobe mode and BC0 in single write strobe mode. Setting the 8-bit external bus width is prohibited in single write strobe mode. For information on other multiplexed pin functions, see section 20, I/O Ports.
Note 2. The WR0 signal and WR signal are identical. The WR0 signal is referred to as WR in single write strobe mode.

15.2.4 Parallel Operation

Parallel operation is possible when different bus masters request access to different slave modules. For example, if the CPU fetches an instruction from the flash and an operand from the SRAM, the DMAC can handle transfers between a peripheral bus and the external bus at the same time.

An example of parallel operation is shown in Figure 15.2. In this example, the CPU uses the instruction and operand buses for simultaneous access to the flash and SRAM, respectively. Additionally, the DMAC/DTC simultaneously use the DMA bus for access to a peripheral bus or external bus during access to the flash memory and SRAM by the CPU.

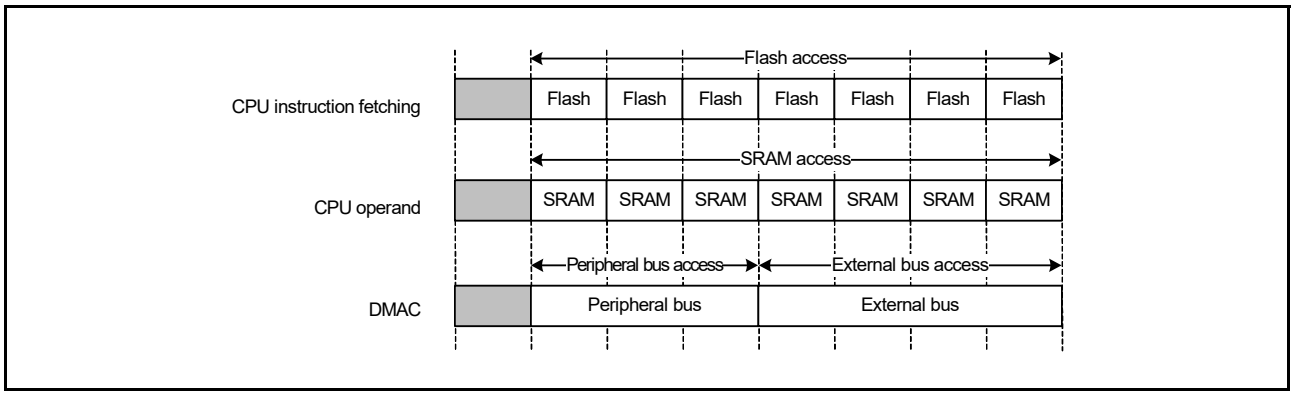


Figure 15.2 Example of parallel operation

15.2.5 Bus Settings

Set up the external bus with the following registers:

- Mode settings:
  CSn Mode Register (CSnMOD), CSn Wait Control Register 1 (CSnWCR1), CSn Wait Control Register 2 (CSnWCR2), CSn Control Register (CSnCR), CSn Recovery Cycle Setting Register (CSnREC), CS Recovery Cycle Insertion Enable Register (CSRECEN), and Bus Priority Control Register (BUSSCNT)
- I/O port assignments:
  PmnPFS.PMR = 1 and PmnPFS.PSEL[4:0] = 0Bh
- Frequency of the external bus clock (BCLK):
  SCKDIVCR register.

See section 20, I/O Ports for information on PmnPFS and section 9, Clock Generation Circuit for information on SCKDIVCR.

15.2.6 Restrictions

(1) Restriction on Endianness

Memory space must be little-endian to execute code on the Cortex-M4 core.

15.3 Register Descriptions

15.3.1 CSn Control Register (CSnCR) (n = 0 to 3)

Address(es): BUS.CS0CR 4000 3802h

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

Address(es): BUS.CS1CR 4000 3812h, BUS.CS2CR 4000 3822h, BUS.CS3CR 4000 3832h

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

Bit	Symbol	Bit name	Description	R/W
b0	EXENB	Operation Enable	0: Disable 1: Enable.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	BSIZE[1:0]	External Bus Width Select	b5 b4 0 0: 16-bit bus space 0 1: Setting prohibited 1 0: 8-bit bus space 1 1: Setting prohibited.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	EMODE	Endian Mode	0: Little endian 1: Big endian.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Do not attempt to write to the CSnCR register while the external bus is being accessed.

EXENB bit (Operation Enable)

The EXENB bit enables or disables operation of the associated CS areas. On MCU reset, operation is enabled (EXENB = 1) only for area 0. Operation in other areas is disabled (EXENB = 0). Attempts to access disabled areas have no effect.

BSIZE[1:0] bits (External Bus Width Select)

The BSIZE[1:0] bits specify the data bus width for the associated area.

EMODE bit (Endian Mode)

The EMODE bit specifies the endianness for the associated area. The Cortex-M4 core is fixed at little-endian order, so instruction code can only be allocated to external spaces with little-endian specified. If an area is specified as big-endian, no instruction code can be allocated to it.

15.3.2 CSn Recovery Cycle Register (CSnREC) (n = 0 to 3)

Address(es): BUS.CS0REC 4000 380Ah, BUS.CS1REC 4000 381Ah, BUS.CS2REC 4000 382Ah, BUS.CS3REC 4000 383Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	WRCV[3:0]				—	—	—	—	RRCV[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	RRCV[3:0]	Read Recovery	b3 b0 0 0 0 0: Do not insert any recovery cycles 0 0 0 1: Insert 1 recovery cycle 0 0 1 0: Insert 2 recovery cycles 0 0 1 1: Insert 3 recovery cycles 0 1 0 0: Insert 4 recovery cycles 0 1 0 1: Insert 5 recovery cycles 0 1 1 0: Insert 6 recovery cycles 0 1 1 1: Insert 7 recovery cycles 1 0 0 0: Insert 8 recovery cycles 1 0 0 1: Insert 9 recovery cycles 1 0 1 0: Insert 10 recovery cycles 1 0 1 1: Insert 11 recovery cycles 1 1 0 0: Insert 12 recovery cycles 1 1 0 1: Insert 13 recovery cycles 1 1 1 0: Insert 14 recovery cycles 1 1 1 1: Insert 15 recovery cycles.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	WRCV[3:0]	Write Recovery	b11 b8 0 0 0 0: No recovery cycle is inserted 0 0 0 1: Insert 1 recovery cycle 0 0 1 0: Insert 2 recovery cycles 0 0 1 1: Insert 3 recovery cycles 0 1 0 0: Insert 4 recovery cycles 0 1 0 1: Insert 5 recovery cycles 0 1 1 0: Insert 6 recovery cycles 0 1 1 1: Insert 7 recovery cycles 1 0 0 0: Insert 8 recovery cycles 1 0 0 1: Insert 9 recovery cycles 1 0 1 0: Insert 10 recovery cycles 1 0 1 1: Insert 11 recovery cycles 1 1 0 0: Insert 12 recovery cycles 1 1 0 1: Insert 13 recovery cycles 1 1 1 0: Insert 14 recovery cycles 1 1 1 1: Insert 15 recovery cycles.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Do not attempt to write to the CSnREC register while the external bus is being accessed.

When the preceding bus access is from a separate bus, CSnREC is valid when the recovery cycle insertion is enabled in the Separate Bus Recovery Cycle Insertion Enable bit (RCVENj (j = 0 to 7)) in the CSRECEN register. For more information, see section 15.5.3, Insertion of Recovery Cycles.

RRCV[3:0] bits (Read Recovery)

The RRCV[3:0] bits specify the number of recovery cycles inserted after a read access on the external bus for each CSn (n = 0 to 3). When recovery cycle insertion is enabled and a value other than 0000b is written to these bits, 1 to 15 recovery cycles are inserted in the following cases:

- After a read access to the external bus, a read access is made to the external bus in the same area
- After a read access to the external bus, a read access is made to the external bus in a different area
- After a read access to the external bus, a write access is made to the external bus in the same area
- After a read access to the external bus, a write access is made to the external bus in a different area.

WRCV[3:0] bits (Write Recovery)

The WRCV[3:0] bits specify the number of recovery cycles inserted after a write access on the external bus for CSn (n = 0 to 3). When the recovery cycle insertion is enabled and a value other than 0000b is written to these bits, 1 to 15 recovery cycles are inserted in the following cases:

- After a write access to the external bus, a read access is made to the external bus in the same area
- After a write access to the external bus, a read access is made to the external bus in a different area
- After a write access to the external bus, a write access is made to the external bus in the same area
- After a write access to the external bus, a write access is made to the external bus in a different area.

15.3.3 CS Recovery Cycle Insertion Enable Register (CSRECEN)

Address(es): BUS.CSRECEN 4000 3880h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	RCVEN7	RCVEN6	RCVEN5	RCVEN4	RCVEN3	RCVEN2	RCVEN1	RCVEN0
Value after reset:	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0

Bit	Symbol	Bit name	Description	R/W
b0	RCVEN0	Separate Bus Recovery Cycle Insertion Enable 0	0: Disable 1: Enable.	R/W
b1	RCVEN1	Separate Bus Recovery Cycle Insertion Enable 1	0: Disable 1: Enable.	R/W
b2	RCVEN2	Separate Bus Recovery Cycle Insertion Enable 2	0: Disable 1: Enable.	R/W
b3	RCVEN3	Separate Bus Recovery Cycle Insertion Enable 3	0: Disable 1: Enable.	R/W
b4	RCVEN4	Separate Bus Recovery Cycle Insertion Enable 4	0: Disable 1: Enable.	R/W
b5	RCVEN5	Separate Bus Recovery Cycle Insertion Enable 5	0: Disable 1: Enable.	R/W
b6	RCVEN6	Separate Bus Recovery Cycle Insertion Enable 6	0: Disable 1: Enable.	R/W
b7	RCVEN7	Separate Bus Recovery Cycle Insertion Enable 7	0: Disable 1: Enable.	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13 to b9	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Do not attempt to write to the CSRECEN register while the external bus is being accessed. For more information, see section 15.5.3, Insertion of Recovery Cycles.

RCVENj bit (Separate Bus Recovery Cycle Insertion Enable j) (j = 0 to 7)

The RCVENj bit enables or disables the insertion of write or read recovery cycles when, after a write or read access to the external bus, a write or read access is made to the external bus in the same or different area.

Table 15.5 Access type association with RCVENj bits

Access type	External address space	Insertion of recovery cycles	Associated bits
Read access after read access	Same area	Recovery cycles specified in the RRCV[3:0] bits for the precedence access area are inserted	RCVEN0
	Different area	Recovery cycles specified in the RRCV[3:0] bits for the precedence access area are inserted	RCVEN1
Write access after read access	Same area	Recovery cycles specified in the RRCV[3:0] bits for the precedence access area are inserted	RCVEN2
	Different area	Recovery cycles specified in the RRCV[3:0] bits for the precedence access area are inserted	RCVEN3
Read access after write access	Same area	Recovery cycles specified in the WRCV[3:0] bits for the precedence access area are inserted	RCVEN4
	Different area	Recovery cycles specified in the WRCV[3:0] bits for the precedence access area are inserted	RCVEN5
Write access after write access	Same area	Recovery cycles specified in the WRCV[3:0] bits for the precedence access area are inserted	RCVEN6
	Different area	Recovery cycles specified in the WRCV[3:0] bits for the precedence access area are inserted	RCVEN7

15.3.4 CSn Mode Register (CSnMOD) (n = 0 to 3)

Address(es): BUS.CS0MOD 4000 3002h, BUS.CS1MOD 4000 3012h, BUS.CS2MOD 4000 3022h, BUS.CS3MOD 4000 3032h

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

Bit	Symbol	Bit name	Description	R/W
b0	WRMOD	Write Access Mode Select	0: Byte strobe mode 1: Single write strobe mode.	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	EWENB	External Wait Enable	0: Disable 1: Enable.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PRENB	Page Read Access Enable	0: Disable 1: Enable.	R/W
b9	PWENB	Page Write Access Enable	0: Disable 1: Enable.	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	PRMOD	Page Read Access Mode Select	0: Normal access compatible mode 1: External data read continuous assertion mode.	R/W

Do not write to the CSnMOD register while access to the CSn area is in progress.

WRMOD bit (Write Access Mode Select)

The WRMOD bit selects the write access operating mode. Writing 0 selects byte strobe mode, in which data writes are controlled by the WRn signals (n = 0 and 1) associated with the respective byte positions. Writing 1 selects single write strobe mode, in which data writes are controlled by the BCn (n = 0 and 1) and the WR signals associated with the respective byte positions.

Note: Setting the external bus width to 8 bits is prohibited in single write strobe mode.

Table 15.6 Control signals for write access mode

	Pin name			
Write access mode	**WR1**	**WR0/WR**	**BC1**	**BC0**
Byte strobe mode	✓	✓ (WR0)	×	×
Single write strobe mode	×	✓ (WR)	✓	✓

✓: Enabled, ×: Disabled

EWENB bit (External Wait Enable)

The EWENB bit enables external waits. Writing 0 disables the WAIT signal. Writing 1 selects external wait and allows the WAIT signal to control the number of waits per cycle. In this state, wait cycles are inserted when the WAIT signal is low.

PRENB bit (Page Read Access Enable)

The PRENB bit enables page read accesses.

PWENB bit (Page Write Access Enable)

The PWENB bit enables page write accesses.

PRMOD bit (Page Read Access Mode Select)

The PRMOD bit selects the operating mode for page read accesses. Writing 0 selects normal access compatible mode, in which the RD signal is negated and an RD assert wait is inserted each time a unit of data is read. When there is no RD assert wait, the RD signal is negated only in the final transfer of the external bus access.

Writing 1 selects external data read continuous assertion mode, in which an RD assert wait is inserted and the RD signal is continuously asserted during the wait.

15.3.5 CSn Wait Control Register 1 (CSnWCR1) (n = 0 to 3)

Address(es): BUS.CS0WCR1 4000 3004h, BUS.CS1WCR1 4000 3014h, BUS.CS2WCR1 4000 3024h, BUS.CS3WCR1 4000 3034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	CSRWAIT[4:0]					—	—	—	CSWWAIT[4:0]				
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

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

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CSPWWAIT[2:0]	Page Write Cycle Wait Select*1	b2 b1 b0 0 0 0: Do not insert wait 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	CSPRWAIT[2:0]	Page Read Cycle Wait Select*2	b10 b9 b8 0 0 0: Do not insert wait 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W

Bit	Symbol	Bit name	Description	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20 to b16	CSWWAIT[4:0]	Normal Write Cycle Wait Select	b20 b16 0 0 0 0 0: Do not insert wait 0 0 0 0 1: Insert wait of 1 clock cycle 0 0 0 1 0: Insert wait of 2 clock cycles 0 0 0 1 1: Insert wait of 3 clock cycles Set value = the n-bit number of clock cycles inserted 1 1 1 0 1: Insert wait of 29 clock cycles 1 1 1 1 0: Insert wait of 30 clock cycles 1 1 1 1 1: Insert wait of 31 clock cycles.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28 to b24	CSRWAIT[4:0]	Normal Read Cycle Wait Select	b28 b24 0 0 0 0 0: Do not insert wait 0 0 0 0 1: Insert wait of 1 clock cycle 0 0 0 1 0: Insert wait of 2 clock cycles 0 0 0 1 1: Insert wait of 3 clock cycles Set value = the n-bit number of clock cycles inserted 1 1 1 0 1: Insert wait of 29 clock cycles 1 1 1 1 0: Insert wait of 30 clock cycles 1 1 1 1 1: Insert wait of 31 clock cycles.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The CSPWWAIT[2:0] value is only valid when the PWENB bit in CSnMOD is set to 1.
Note 2. The CSPRWAIT[2:0] value is only valid when the PRENB bit in CSnMOD is set to 1.

Do not attempt to write to the CSnWCR1 register while the external bus is being accessed. Set each of these bits to satisfy the restrictions described in section 15.5.6 (1) Constraints on using separate bus interface.

CSPWWAIT[2:0] bits (Page Write Cycle Wait Select)

The CSPWWAIT[2:0] bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page write cycle. The setting is enabled when the PWENB bit in CSnMOD is set to 1.

Note: The settings must satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value, and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

CSPRWAIT[2:0] bits (Page Read Cycle Wait Select)

The CSPRWAIT[2:0] bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page read cycle. The setting is enabled when the PRENB bit in CSnMOD is set to 1.

Note: The settings must satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.

CSWWAIT[4:0] bits (Normal Write Cycle Wait Select)

The CSWWAIT[4:0] bits specify the number of wait cycles to be inserted into the first access during a normal write cycle or page write cycle.

Note: The settings must satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value, and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

CSRWAIT[4:0] bits (Normal Read Cycle Wait Select)

The CSRWAIT[4:0] bits specify the number of wait cycles to be inserted into the first access during a normal read cycle or page read cycle.

Note: The settings must satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

15.3.6 CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 3)

Address(es): BUS.CS0WCR2 4000 3008h, BUS.CS1WCR2 4000 3018h, BUS.CS2WCR2 4000 3028h, BUS.CS3WCR2 4000 3038h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CSON[2:0]			—	WDON[2:0]			—	WRON[2:0]			—	RDON[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CSROFF[2:0]	Read-Access CS Extension Cycle Select	b2 b0 0 0 0: No wait is inserted 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	CSWOFF[2:0]	Write-Access CS Extension Cycle Select	b6 b4 0 0 0: No wait is inserted 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	WDOFF[2:0]	Write Data Output Extension Cycle Select	b10 b8 0 0 0: No wait is inserted 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	RDON[2:0]	RD Assert Wait Select	b18 b16 0 0 0: No wait is inserted. 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b22 to b20	WRON[2:0]	WR Assert Wait Select	b22 b20 0 0 0: No wait is inserted 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b26 to b24	WDON[2:0]	Write Data Output Wait Select	b26 b24 0 0 0: No wait is inserted 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30 to b28	CSON[2:0]	CS Assert Wait Select	b30 b28 0 0 0: No wait is inserted 0 0 1: Insert wait of 1 clock cycle 0 1 0: Insert wait of 2 clock cycles 0 1 1: Insert wait of 3 clock cycles 1 0 0: Insert wait of 4 clock cycles 1 0 1: Insert wait of 5 clock cycles 1 1 0: Insert wait of 6 clock cycles 1 1 1: Insert wait of 7 clock cycles.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Do not attempt to write to the CSnWCR2 register while the external bus is being accessed. Set each of these bits to satisfy the restrictions described in section 15.5.6 (1), Constraints on using separate bus interface.

CSROFF[2:0] bits (Read-Access CS Extension Cycle Select)

The CSROFF[2:0] bits specify the number of wait cycles to be inserted during the period from the end of a wait cycle (RD signal negated) until the CSn signal (n = 0 to 3) is negated in read access mode.

CSWOFF[2:0] bits (Write-Access CS Extension Cycle Select)

The CSWOFF[2:0] bits specify the number of wait cycles to be inserted during the period from the end of a wait cycle (WRn signal (n = 0 and 1) negated) until the CSn signal (n = 0 to 3) is negated in write access mode.

Note: The settings must satisfy CSnWCR2.WDOFF[2:0] value ≤ CSnWCR2.CSWOFF[2:0] value.

WDOFF[2:0] bits (Write Data Output Extension Cycle Select)

The WDOFF[2:0] bits specify the number of wait cycles to be inserted during the period from the end of a wait cycle (WRn signal (n = 0 and 1) negated) until the write data output is complete in write access mode.

Note: The settings must satisfy CSnWCR2.WDOFF[2:0] value ≤ CSnWCR2.CSWOFF[2:0] value.

RDON[2:0] bits (RD Assert Wait Select)

The RDON[2:0] bits specify the number of wait cycles to be inserted before the RD signal is asserted.

Note: For normal read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For page read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.

WRON[2:0] bits (WR Assert Wait Select)

The WRON[2:0] bits specify the number of wait cycles to be inserted before the WRn signal (n = 0 to 1) is asserted.

Note: For normal write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value, and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value, and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

WDON[2:0] bits (Write Data Output Wait Select)

The WDON[2:0] bits specify the number of wait cycles to be inserted before the write data is output.

Note: For normal write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

CSON[2:0] bits (CS Assert Wait Select)

The CSON[2:0] bits specify the number of wait cycles to be inserted before the CSn signal (n = 0 to 3) is asserted.

Note: For normal read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For page read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.
For normal write access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

15.3.7 Master Bus Control Register (BUSMCNT<master>)

Address(es): BUS.BUSMCNTM4I 4000 4000h, BUS.BUSMCNTM4D 4000 4004h, BUS.BUSMCNTSYS 4000 4008h, BUS.BUSMCNTDMA 4000 400Ch

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

Bit	Symbol	Bit name	Description	R/W
b14 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	IERES	Ignore Error Responses	0: Report a bus error 1: Do not report a bus error.	R/W

Note: Changing reserved bits from the initial value 0 is prohibited. Operation during the change is not guaranteed.

IERES bit (Ignore Error Responses)

The IERES bit specifies the enable or disable of the AHB-Lite protocol error response.

Table 15.7 lists the registers associated with each bus type.

Table 15.7 Associations between bus types and registers

Bus type	Master Bus Control Register	Slave Bus Control Register	Bus Error Address Register	Bus Error Status Register
ICode bus (CPU)	BUSMCNTM4I	-	BUS1ERRADD	BUS1ERRSTAT
DCode bus (CPU)	BUSMCNTM4D	-	BUS2ERRADD	BUS2ERRSTAT
SYSTEM bus (CPU)	BUSMCNTSYS	-	BUS3ERRADD	BUS3ERRSTAT
DMA bus	BUSMCNTDMA	-	BUS4ERRADD	BUS4ERRSTAT
Memory bus 1	-	BUSSCNTFLI	-	-
Memory bus 3	-	BUSSCNTMBIU	-	-
Memory bus 4	-	BUSSCNTRAM0	-	-
Memory bus 5	-	BUSSCNTRAM1	-	-
Internal peripheral bus 1, 3, 4, 5, 7	-	BUSSCNTPnB [n = 0, 2, 3, 4, 6]	-	-
Internal peripheral bus 9	-	BUSSCNTFBU	-	-
External bus (CS area)	-	BUSSCNTEXT	-	-
External bus (QSPI area)	-	BUSSCNTEXT2	-	-

15.3.8 Slave Bus Control Register (BUSSCNT<slave>)

Address(es): BUS.BUSSCNTFLI 4000 4100h, BUS.BUSSCNTMBIU 4000 4108h, BUS.BUSSCNTRAM0 4000 410Ch, BUS.BUSSCNTRAM1 4000 4110h, BUS.BUSSCNTP0B 4000 4114h, BUS.BUSSCNTP2B 4000 4118h, BUS.BUSSCNTP3B 4000 411Ch, BUS.BUSSCNTP4B 4000 4120h, BUS.BUSSCNTP6B 4000 4128h, BUS.BUSSCNTFBU 4000 4130h, BUS.BUSSCNTEXT 4000 4134h, BUS.BUSSCNTEXT2 4000 4138h

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

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	ARBMET[1:0]	Arbitration Method	Specifies the group priorities: b5 b4 0 0: Fixed priority 0 1: Round-robin 1 0: Setting prohibited 1 1: Setting prohibited.	R/W
b15 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Changing the reserved bits from the initial value of 0 is prohibited. Operation during the change is not guaranteed.

ARBMET[1:0] bit (Arbitration Method)

The ARBMET [1:0] bits specify the priority of each bus master. For fixed priority, see Table 15.8. For round-robin, see Table 15.9. Table 15.7 lists the registers associated with each bus type.

Table 15.8 Fixed priority (ARBMET[1:0] = 00b)

Slave Bus Control Register	Slave interface	Priority
BUSSCNTFLI	Memory bus 1	Memory bus 3 > DCode bus (CPU) > ICode bus (CPU)
BUSSCNTRAM0	Memory bus 4	DMA bus > System bus (CPU)
BUSSCNTRAM1	Memory bus 5	DMA bus > System bus (CPU)
BUSSCNTPnB [n = 0, 2, 3, 4, 6]	Internal peripheral bus 1, 3, 4, 5, 7	DMA bus > System bus (CPU)
BUSSCNTFBU	Internal peripheral bus 9	DMA bus > System bus (CPU)
BUSSCNTEXT	External bus (CS area)	DMA bus > System bus (CPU)
BUSSCNTEXT2	External bus (QSPI area)	DMA bus > System bus (CPU)

Table 15.9 Round-robin priority (ARBMET[1:0] = 01b)

Slave Bus Control Register	Slave interface	Priority "↔": Round-Robin
BUSSCNTFLI	Memory bus 1	Memory bus 3 ↔ DCode bus (CPU) ↔ ICode bus (CPU)
BUSSCNTRAM0	Memory bus 4	DMA bus ↔ System bus (CPU)
BUSSCNTRAM1	Memory bus 5	DMA bus ↔ System bus (CPU)
BUSSCNTPnB [n = 0, 2, 3, 4, 6]	Internal peripheral bus 1, 3, 4, 5, 7	DMA bus ↔ System bus (CPU)
BUSSCNTFBU	Internal peripheral bus 9	DMA bus ↔ System bus (CPU)
BUSSCNTEXT	External bus (CS area)	DMA bus ↔ System bus (CPU)
BUSSCNTEXT2	External bus (QSPI area)	DMA bus ↔ System bus (CPU)

15.3.9 Bus Error Address Register (BUSnERRADD) (n = 1 to 4)

Address(es): BUS.BUS1ERRADD 4000 4800h, BUS.BUS2ERRADD 4000 4810h, BUS.BUS3ERRADD 4000 4820h, BUS.BUS4ERRADD 4000 4830h

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

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

Bit	Symbol	Bit name	Description	R/W
b31 to b0	BERAD[31:0]	Bus Error Address	When a bus error occurs, these bits store the error address	R

Note: This register is cleared only by resets other than MPU related resets. For more information, see section 6, Resets and section 16, Memory Protection Unit (MPU).

Table 15.7 lists the registers associated with each bus type.

BERAD[31:0] bits (Bus Error Address)

When a bus error occurs, these bits store the access address. For more information, see BUSnERRSTAT.ERRSTAT and section 15.6, Bus Error Monitoring Section. A value of the BUSnERRADD.BERAD[31:0] (n = 1 to 4) is effective only when BUSnERRSTAT.ERRSTAT(n = 1 to 4) is set to 1.

15.3.10 Bus Error Status Register (BUSnERRSTAT) (n = 1 to 4)

Address(es): BUS.BUS1ERRSTAT 4000 4804h, BUS.BUS2ERRSTAT 4000 4814h, BUS.BUS3ERRSTAT 4000 4824h, BUS.BUS4ERRSTAT 4000 4834h

	b7	b6	b5	b4	b3	b2	b1	b0
	ERRSTAT	—	—	—	—	—	—	ACCSTAT
Value after reset:	0	0	0	0	0	0	0	x

Bit	Symbol	Bit name	Description	R/W
b0	ACCSTAT	Error Access Status	Access status when the error occurred: 1: Write access 0: Read access.	R
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ERRSTAT	Bus Error Status	0: No bus error occurred 1: Bus error occurred.	R

Note: This register is cleared only by resets other than MPU related resets. For more information, see section 6, Resets and section 16, Memory Protection Unit (MPU).

Table 15.7 lists the registers associated with each bus type.

ACCSTAT bits (Error Access Status)

The ACCSTAT bit indicates the access status, write access or read access, when an error occurs on the associated bus. For more information, see BUSnERRSTAT.ERRSTAT and section 15.6, Bus Error Monitoring Section.

The value is valid only when BUSnERRSTAT.ERRSTAT (n = 1 to 4) is set to 1.

ERRSTAT bit (Bus Error Status)

The ERRSTAT bit indicates whether a bus error occurred. When an error occurs, the access address and status of write or read access are stored. BUSnERRSTAT.ERRSTAT (n = 1 to 4) is set to 1.

The following four types of errors can occur on each bus:

- Illegal address access
- Bus master MPU error
- Bus slave MPU error
- Time out.

When detecting bus master or bus slave MPU errors, with the reset selected in the OAD bit, this bit does not set to 1 if the bus access causing the MPU error completes later than the internal reset signal being generated, which can occur with the wait setting.

When detecting bus master MPU errors or bus slave MPU errors, with the non-maskable interrupt selected in the OAD bit, this bit is set to 1 once the bus access causing the MPU error completes.

For more information on errors that occur on each bus, see section 15.6, Bus Error Monitoring Section, and section 16, Memory Protection Unit (MPU).

15.4 Endianness and Data Alignment

The external bus has a data alignment function to control which byte of the data bus (D15 to D08, or D07 to D00) is used when accessing the external address space (the CS area). Alignment is based on the bus specifications of the area to be accessed (8-bit or 16-bit bus space), the data size, and the endian order.

15.4.1 Data Alignment Control for the CS Areas

(1) 16-bit bus space

When a 16-bit bus space is selected in the BSIZE[1:0] bits in CSnCR, address buses A16 to A01 are enabled to output address signals in 16-bit units, and the address bus A00 is disabled (always outputting low).

When byte strobe mode is selected (WRMOD = 0 in CSnMOD), the WR0 and WR1 pins are enabled. The BC0 and BC1 pins are not used.

When single write strobe mode is selected (WRMOD = 1 in CSnMOD), only the WR0 pin is enabled, and it always outputs low during write access, regardless of the data size. The WR1 pin is invalid (always outputting high). The valid byte position is indicated by the BC0 and BC1 pins.

The valid positions of control signals and data external to the chip differ according to the endian order. See Figure 15.3 and Figure 15.4.

Page accesses can occur for accesses to data in 32-bit units. Page accesses can only occur when an access does not extend over a 32-bit boundary and causes no change in the BC0 and BC1 signals. The situations in which page accesses occur are indicated by the letter (p) in Figure 15.3 and Figure 15.4.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1/BC1 WR0/BC0 RD Data Bus D15 D08 D07 D00
8 bits	4n	One	First	8 bits	4n		7 0
	4n+1	One	First	8 bits	4n		7 0
	4n+2	One	First	8 bits	4n+2		7 0
	4n+3	One	First	8 bits	4n+2		7 0
16 bits	4n	One	First	16 bits	4n		15 8 7 0
	4n+2	One	First	16 bits	4n+2		15 8 7 0
32 bits	4n	Two	First	16 bits	4n		15 8 7 0
			Second	16 bits	4n+2	(p)	31 24 23 16

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 15.3 Data alignment in 16-bit bus space with little-endian order

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1/BC1 WR0/BC0 RD Data Bus D15 D08 D07 D00
8 bits	4n	One	First	8 bits	4n		7 0
	4n+1	One	First	8 bits	4n		7 0
	4n+2	One	First	8 bits	4n+2		7 0
	4n+3	One	First	8 bits	4n+2		7 0
16 bits	4n	One	First	16 bits	4n		15 8 7 0
	4n+2	One	First	16 bits	4n+2		15 8 7 0
32 bits	4n	Two	First	16 bits	4n		31 24 23 16
			Second	16 bits	4n+2	(p)	15 8 7 0

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 15.4 Data alignment in 16-bit bus space with big-endian order

(2) 8-Bit Bus Space

When an 8-bit bus space is selected in the BSIZE[1:0] bits in CSnCR, the address buses A16 to A00 are enabled to output address signals in byte units.

In 8-bit bus space, only the WR0 pin is valid, regardless of the write access mode, and it always outputs low during write access. The WR1 pin and the BC0 pin are not used.

The valid positions of data external to the chip are D07 to D00, and WR0 is used as the control signal, regardless of the endian mode. See Figure 15.5 and Figure 15.6.

Page access can occur for accesses to data in 16-bit or 32-bit units. Page access can only occur when an access does not extend over a 32-bit boundary. The situations in which page access occurs are indicated by the letter (p) in Figure 15.5 and Figure 15.6.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1/BC1 RD D15 D08	WR0/BC0 RD D07 D00
8 bits	4n	One	First	8 bits	4n			7 0
	4n+1	One	First	8 bits	4n+1			7 0
	4n+2	One	First	8 bits	4n+2			7 0
	4n+3	One	First	8 bits	4n+3			7 0
16 bits	4n	Two	First	8 bits	4n			7 0
			Second	8 bits	4n+1	(p)		15 8
	4n+2	Two	First	8 bits	4n+2			7 0
			Second	8 bits	4n+3	(p)		15 8
32 bits	4n	Four	First	8 bits	4n			7 0
			Second	8 bits	4n+1	(p)		15 8
			Third	8 bits	4n+2	(p)		23 16
			Fourth	8 bits	4n+3	(p)		31 24

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 15.5 Data alignment in 8-bit bus space with little-endian order

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1/BC1 RD D15 D08	WR0/BC0 RD D07 D00
8 bits	4n	One	First	8 bits	4n			7 0
	4n+1	One	First	8 bits	4n+1			7 0
	4n+2	One	First	8 bits	4n+2			7 0
	4n+3	One	First	8 bits	4n+3			7 0
16 bits	4n	Two	First	8 bits	4n			15 8
			Second	8 bits	4n+1	(p)		7 0
	4n+2	Two	First	8 bits	4n+2			15 8
			Second	8 bits	4n+3	(p)		7 0
32 bits	4n	Four	First	8 bits	4n			31 24
			Second	8 bits	4n+1	(p)		23 16
			Third	8 bits	4n+2	(p)		15 8
			Fourth	8 bits	4n+3	(p)		7 0

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 15.6 Data alignment in 8-bit bus space with big-endian order

15.5 Operation of CS Area Controller

15.5.1 Separate Bus

This section describes the periods shown in the timing charts. The CS area controller (CSC) operates in synchronization with the external bus clock, BCLK. Operation cycles, such as wait cycles, specified in the CSC register, are counted on BCLK. In the following description, the frequencies of BCLK and EBCLK pin output are the same, unless otherwise noted. Access through the external bus starts at the same point as the output of a rising edge on the EBCLK pin. However, if the external bus clock, BCLK and the output on the EBCLK pin are at different frequencies, the wait settings can cause the start of access for the second and subsequent rounds to coincide with the falling edge of the output on the EBCLK pin. See Figure 15.12 to Figure 15.16. If recovery cycles are inserted for bus access, the setting for the number of recovery cycles can also cause the start of access for the second and subsequent rounds to coincide with the falling edge of the output on the EBCLK pin, see Figure 15.28.

(a) Tw1 to Twn (clock cycles for waiting for a normal read cycle or normal write cycle)

The period Tw1 to Twn is made up of the number of clock cycles between the start of access through the external bus clock to one cycle before the strobe signal is valid. The number of cycles is selectable from 0 to 31. Within this period, the timing of CSn, RD, and WRn assertion (placing the signals low) is determined by the respective wait settings. The wait periods are controlled by the CS Assert Wait Select Bits (CSON), the RD Assert Wait Select Bits (RDON), the WR Assert Wait Select Bits (WRON), and the Write Data Output Wait Select Bits (WDON) in the CSn Wait Control Register 2 (CSnWCR2). The number of clock cycles for each of these wait periods is selectable from 0 to 7, counted from the start of external bus access. The selectable number of cycles is also within the overall number of clock cycles required for waiting to read or write.

(b) Tend (clock cycle where the strobe signal is valid)

Tend is the next clock cycle after completion of the wait period for a normal cycle of read or write, or for a cycle of page reading or page writing. If the wait select bit for these cycles is 0, bus access starts on the clock cycle where the strobe signal is valid. The RD and WRn signals are negated in the next clock cycle. For a read access, the clock cycle where the strobe signal is valid is where the data to be read is sampled. If an external wait is enabled, the wait signal is sampled on the cycle where the strobe signal is valid. The bus cycle is extended if the wait signal is low. The bus cycle completes in the next clock cycle if the wait signal is high. Tend indicates the cycle where sampling of the wait signal starts.

After the 1st cycle where the strobe signal is valid during page access, second and subsequent page access operations (see section (e), Tpw1 to Tpwn (Page Read Cycle Wait or Page Write Cycle Wait)) start in the next cycle, except during write access with a setting other than 0 for write-data output extension clock cycles (see section (d), Tdw1 to Tdwn (Clock Cycles for Write-Data Output Extension)). If the setting for the RD or WR assertion wait is any value other than 0, the RD and WRn signals are negated in the next clock cycle. If the setting is 0, assertion continues. Additionally, the CSn signal continues to be asserted rather than negated.

(c) Tn1 to Tnm (Clock Cycles of CS Extension)

For normal access, Tn1 to Tnm represent the clock cycles of the period following the cycle where the strobe signal is valid (Tend) up to negation of the CSn signal. For read or write access, the negation timing can be controlled by the read-access CS Extension Cycle Select Bits (CSROFF) and the write-access CS Extension Cycle Select Bits (CSWOFF) in the CSn Wait Control Register 2 (CSnWCR2). The number of cycles is counted from the cycle following the cycle where the strobe signal is valid.

For page access, Tn1 to Tnm represent the clock cycles of the period following the last cycle where the strobe signal is valid up to negation of the CSn signal.

For write access, setting the Write Data Output Extension Cycle Select Bits (WDOFF) controls extension of the period where the address and output data is valid.

(d) Tdw1 to Tdwn (Clock Cycles for Write-Data Output Extension)

For write access, if the wait setting for the write-data output extension is any value other than 0, the specified clock cycles are inserted from the cycle following the cycle where the strobe signal is valid (Tend).

For normal access, this is inserted within the clock period for CS extension (see section (c), Tn1 to Tnm (Clock Cycles of CS Extension)).

For page access, this is inserted within the clock cycle period where the strobe signal is valid and subsequent page accesses, or within the clock cycle period for the CS extension (see section (c), Tn1 to Tnm (Clock Cycles of CS Extension)). Valid address and data output are extended over this period, and the WRn signal is negated.

(e) Tpw1 to Tpwn (Page Read Cycle Wait or Page Write Cycle Wait)

For the second and subsequent bus cycles during page access, the values for a page read cycle wait or page write cycle wait are used instead of the settings for a normal read or write cycle wait. The settings in the WR Assert Wait Select Bits become enabled in the same way as for the first access. The RD assertion control operation depends on the page read access mode setting (the PRMOD bit in CSnMOD) as follows:

CSnMOD.PRMOD = 0: A wait for RD assertion is inserted in the same way as for the first access, and the RD signal is negated.

CSnMOD.PRMOD = 1: Although a wait for RD assertion is inserted in the same way as for normal-access compatibility mode, the RD signal continues to be asserted over this period.

(f) Tr1 to Trn (Recovery Cycles)

Recovery cycles can be inserted from the point where a bus cycle is complete (CSn signal negation). The number of recovery cycles can be controlled by setting the Read Recovery (RRCV) or Write Recovery (WRCV) bits in the CSn Recovery Cycle Register (CSnREC). Both numbers of recovery cycles are counted from the end of a bus cycle (CSn negation) and can be selected from 0 to 15 cycles. For more information, see section 15.5.3, Insertion of Recovery Cycles.

(1) Normal Access

When the PRENB and PWENB bits in CSnMOD are set to 0 to disable page read and page write access, all bus accesses take the form of normal read and write operations. Even when the PRENB and PWENB bits in CSnMOD are set to 1 to enable page read and page write access, bus access other than page access takes the form of normal read and write operations. Figure 15.7 to Figure 15.9 show the normal access operations.

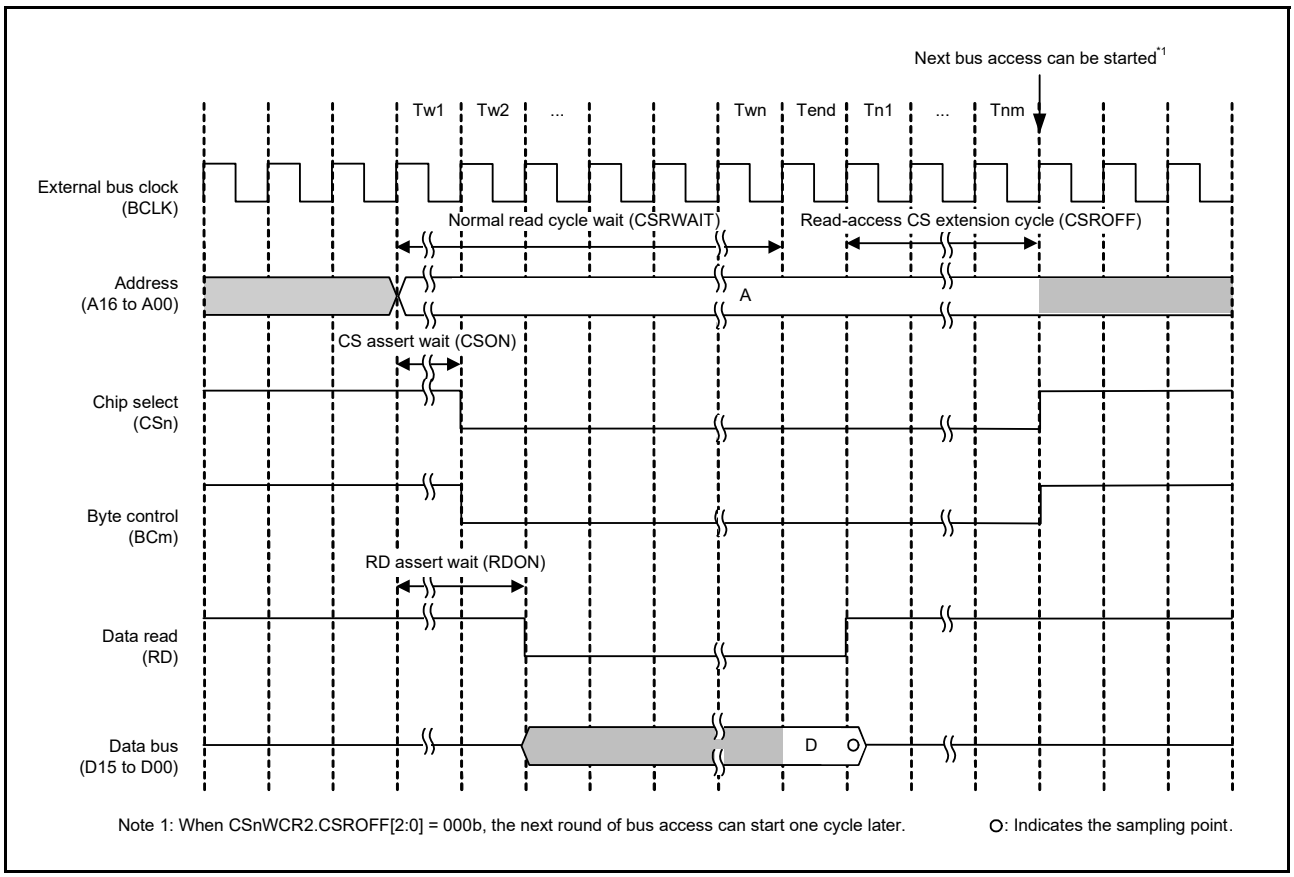


Figure 15.7 Bus timing for normal read operation (n = 0 to 3; m = 0, 1)

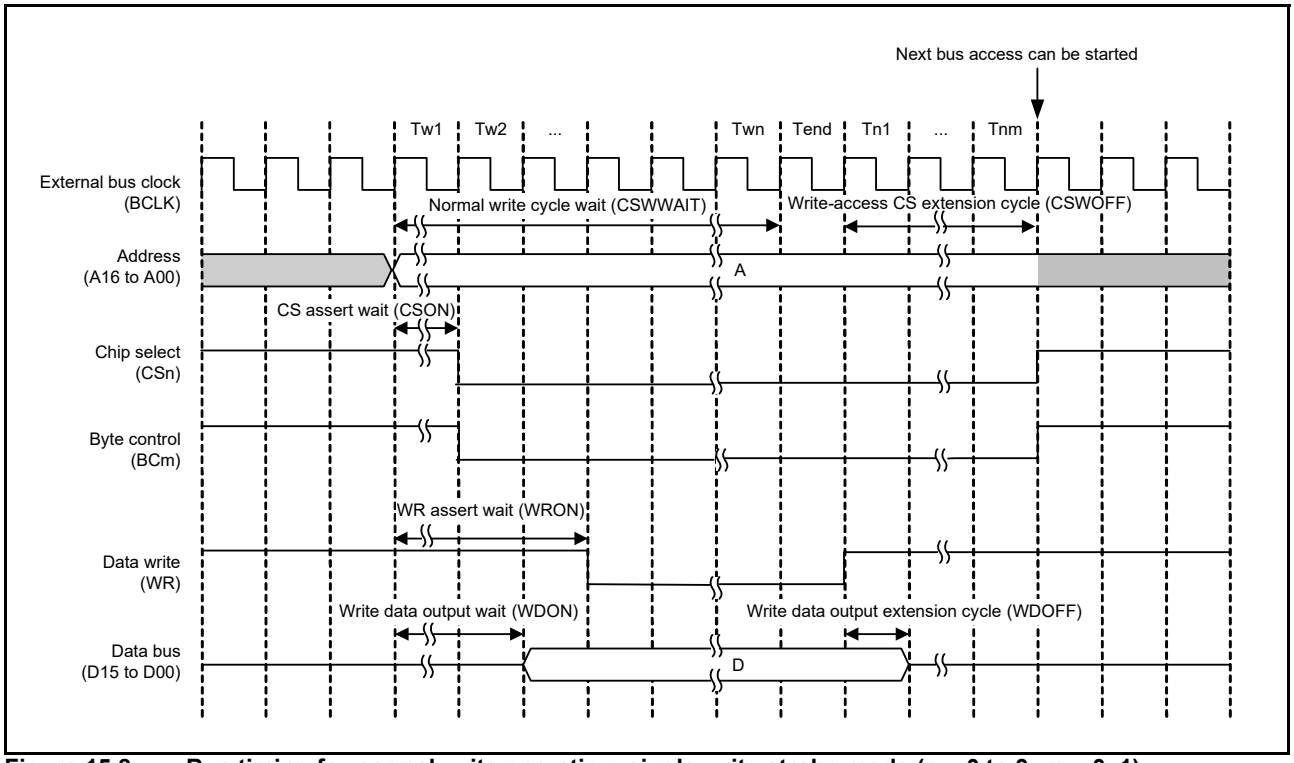


Figure 15.8 Bus timing for normal write operation, single write strobe mode (n = 0 to 3; m = 0, 1)

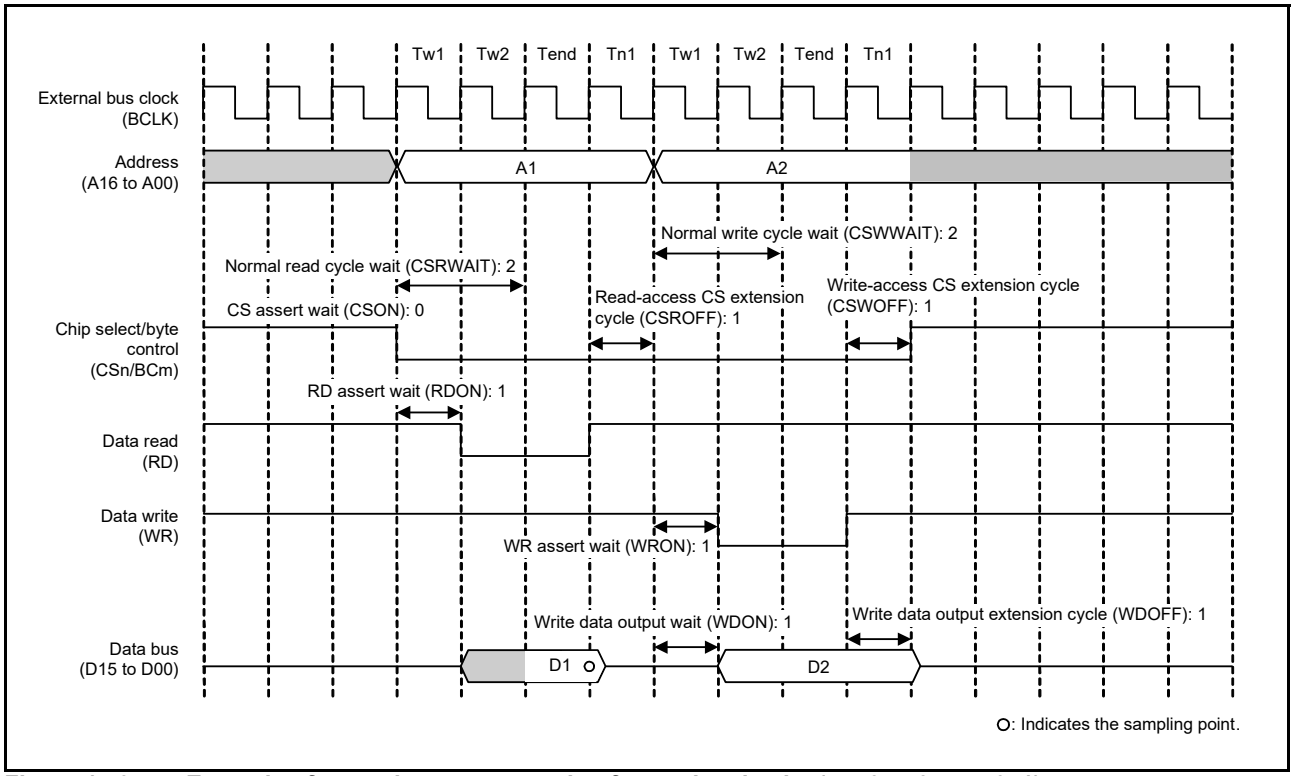


Figure 15.9 Example of normal access operation for read and write (n = 0 to 3; m = 0, 1)

When two or more rounds of external bus access are required in response to a single request for transfer from a bus master, normal access operations are repeated. See section (a), Tw1 to Twn (clock cycles for waiting for a normal read cycle or normal write cycle) to section (d), Tdw1 to Tdwn (Clock Cycles for Write-Data Output Extension). Figure 15.10 and Figure 15.11 show examples of operations when two rounds of bus access are generated in response to a single transfer request. If the recovery cycle insertion condition is satisfied, recovery cycles (section (f), Tr1 to Trn (Recovery Cycles)) are also inserted in the second and subsequent external bus accesses. See Figure 15.26.

The values in the wait control registers shown in the figures are example settings. In your application, set the register bits according to the specifications of connected devices.

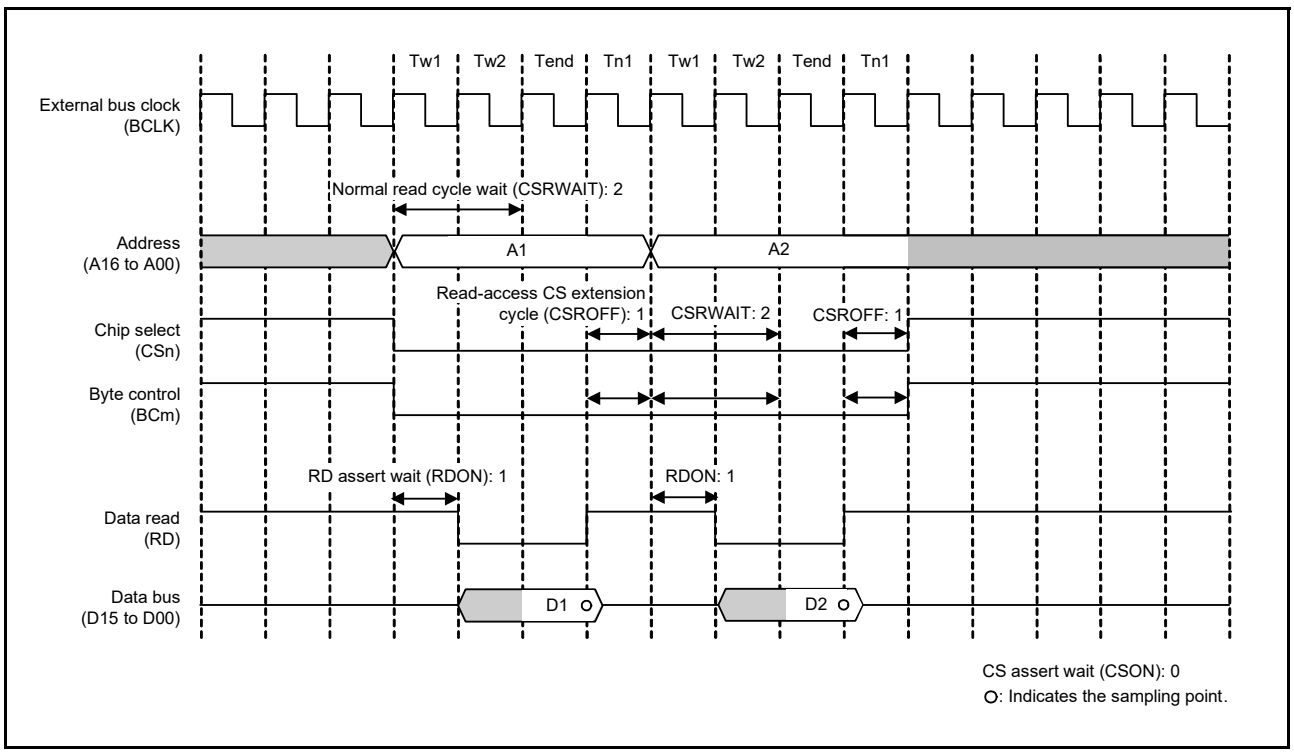


Figure 15.10 Example of normal read operation when two rounds of bus access are generated in response to a single transfer request (n = 0 to 3; m = 0, 1)

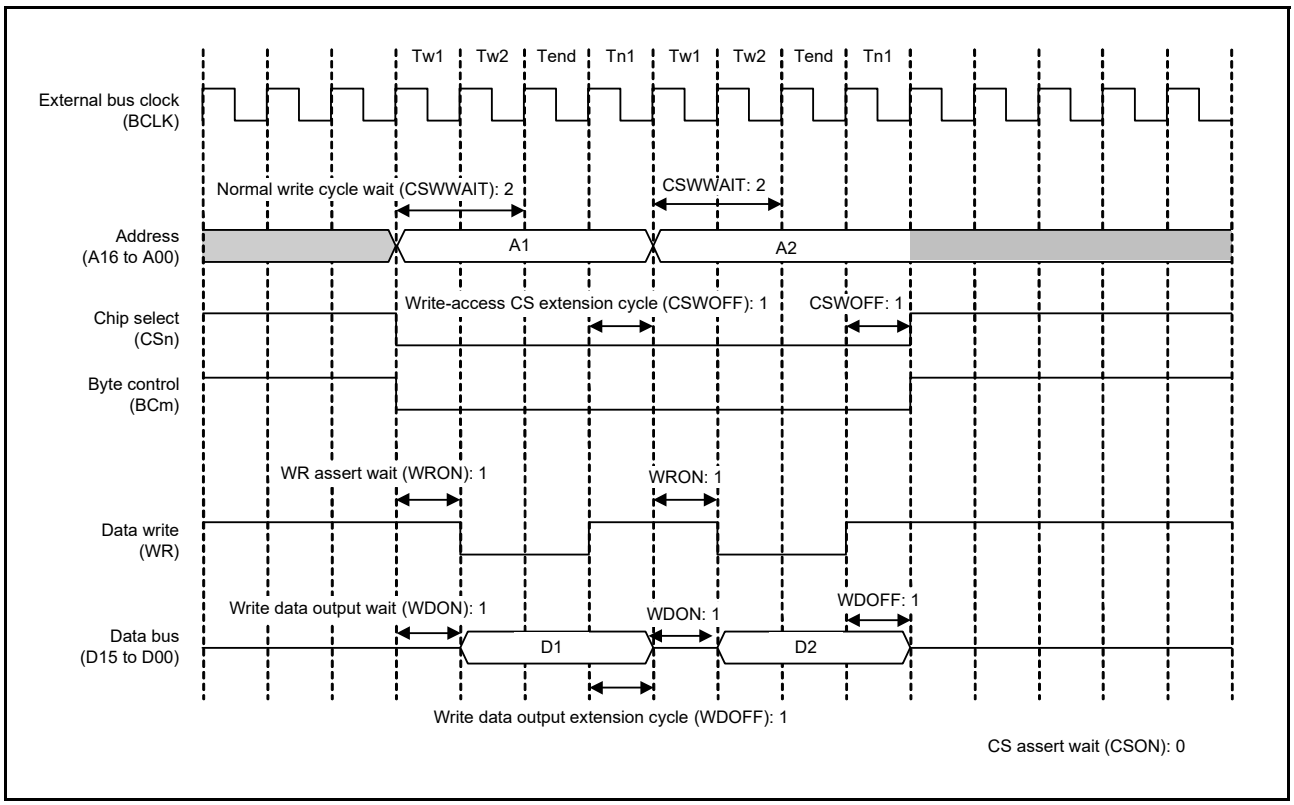


Figure 15.11 Example of normal write operation when two rounds of bus access are generated in response to a single transfer request in single write strobe mode (n = 0 to 3; m = 0, 1)

Figure 15.12 to Figure 15.16 show examples of normal accesses made when BCLK/2 is selected in the EBCLK Pin Output Select bit.

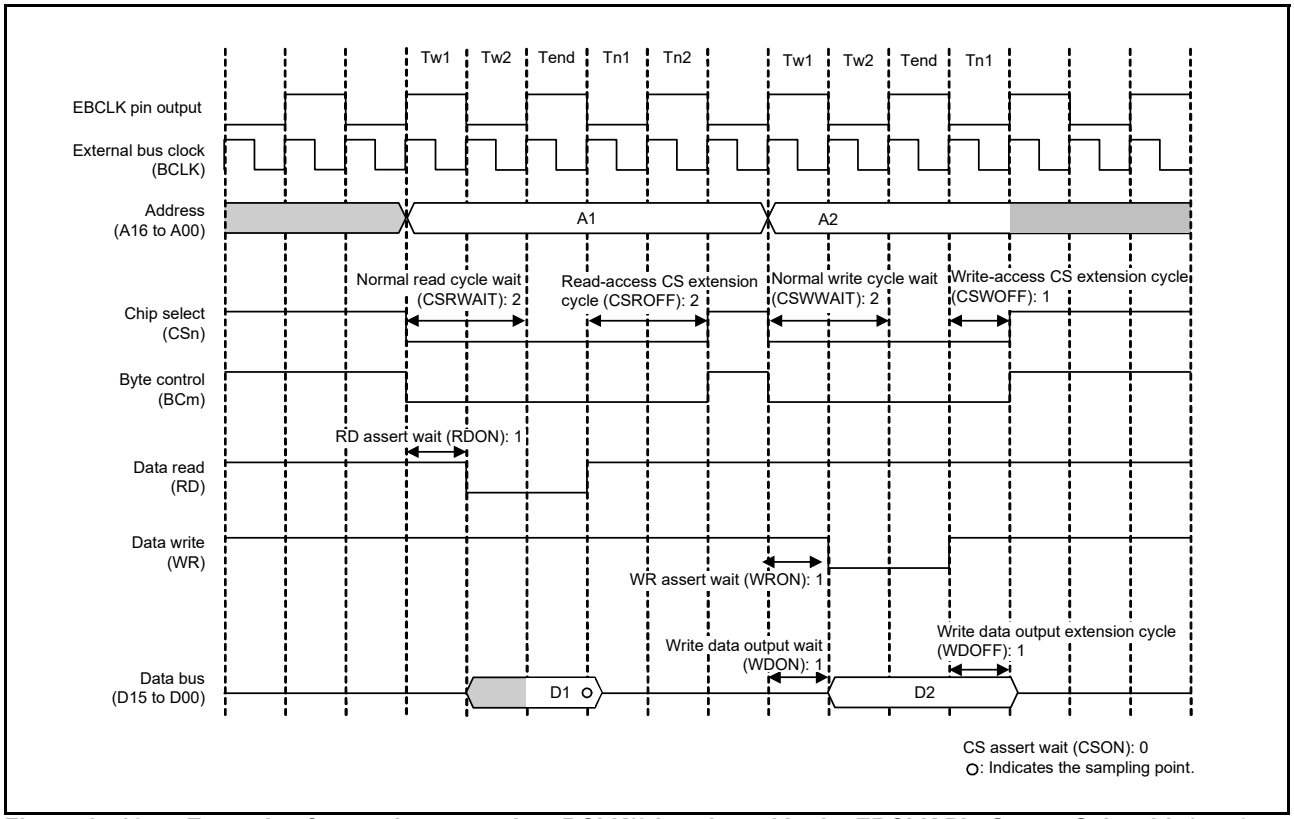


Figure 15.12 Example of normal access when BCLK/2 is selected in the EBCLK Pin Output Select bit (n = 0 to 3, m = 0, 1)

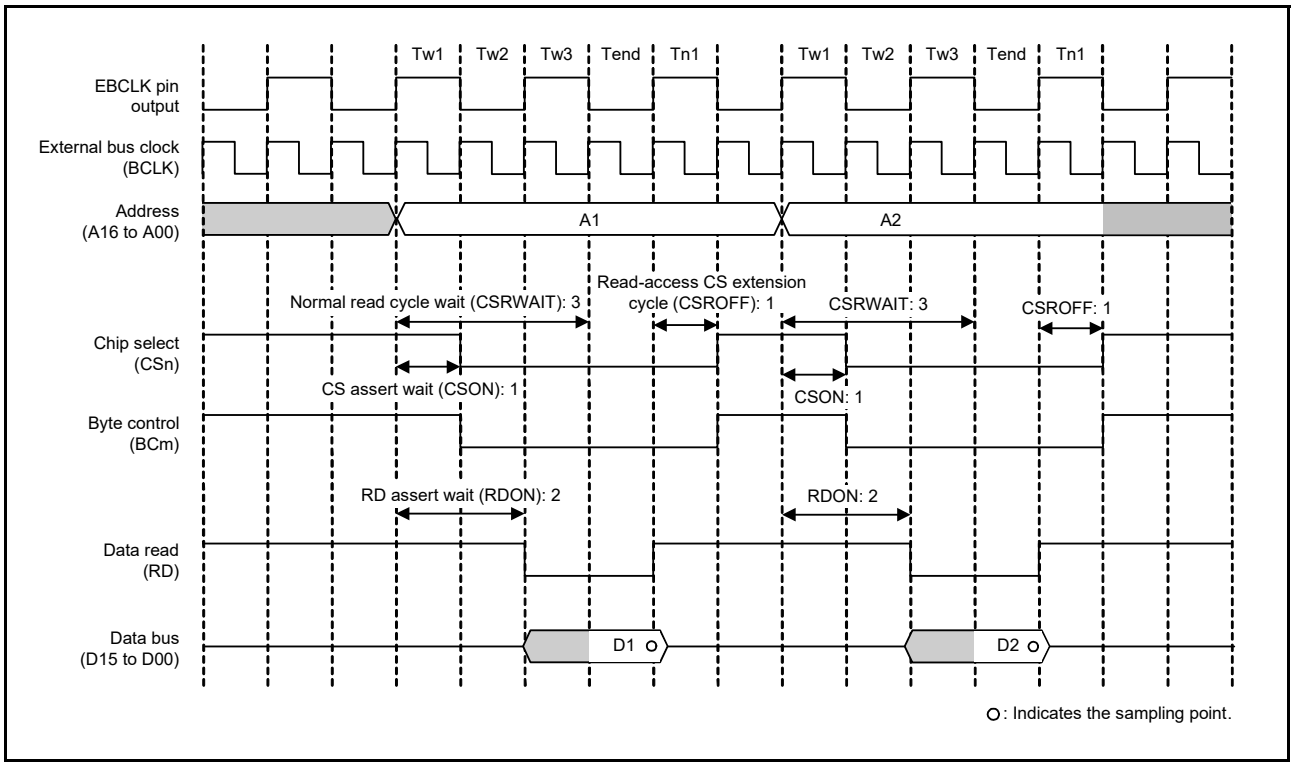


Figure 15.13 Example of normal read operation when BCLK/2 is selected in the EBCLK Pin Output Select bit (n = 0 to 3, m = 0, 1)

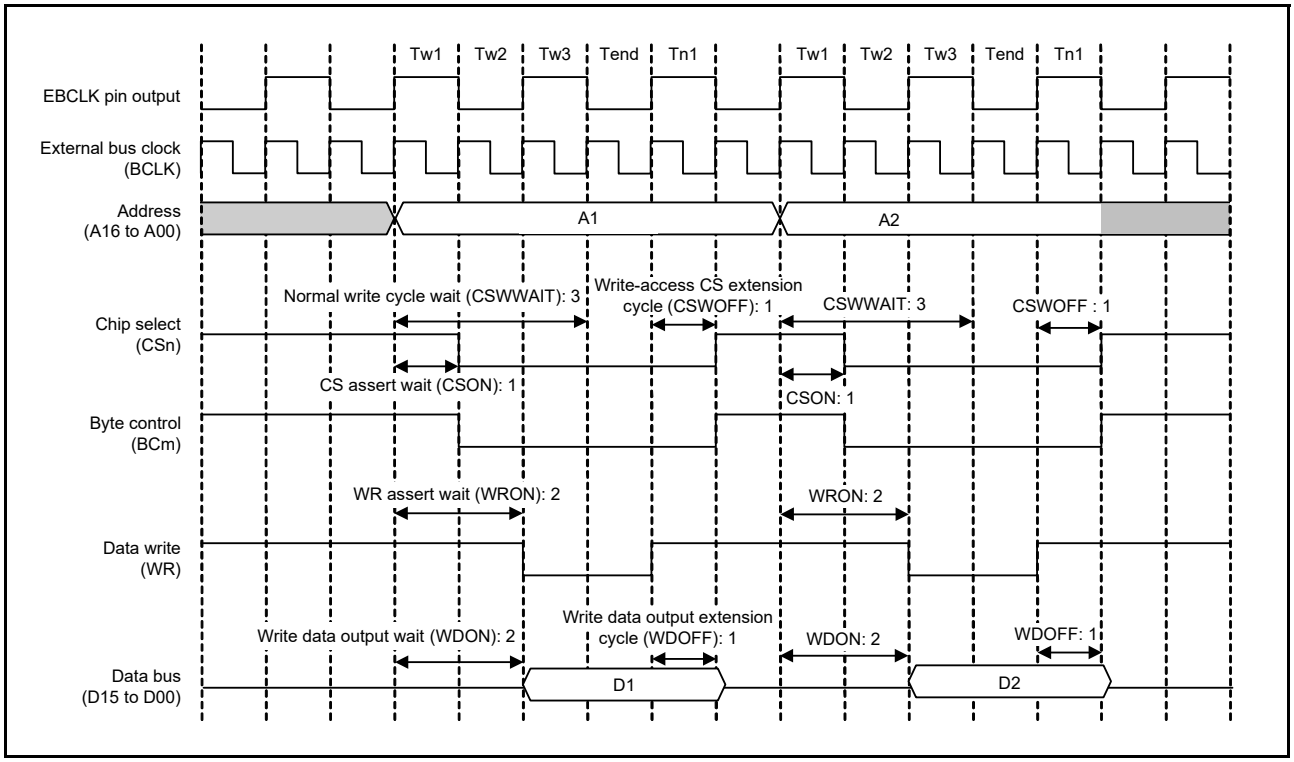


Figure 15.14 Example of normal write operation when BCLK/2 is selected in the EBCLK Pin Output Select bit (n = 0 to 3, m = 0, 1)

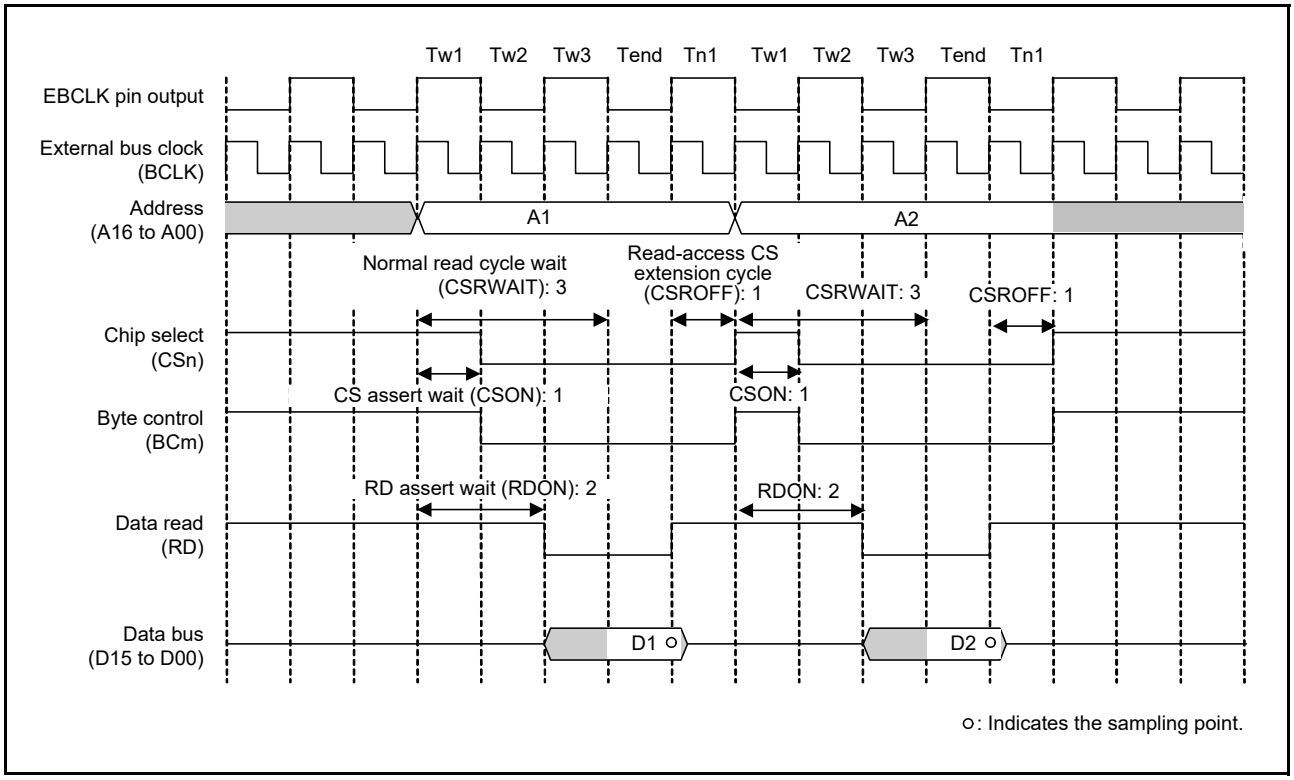


Figure 15.15 Example of normal read operation when BCLK/2 is selected in the EBCLK Pin Output Select bit and two rounds of bus access are generated in response to a single transfer request (n = 0 to 3, m = 0, 1)

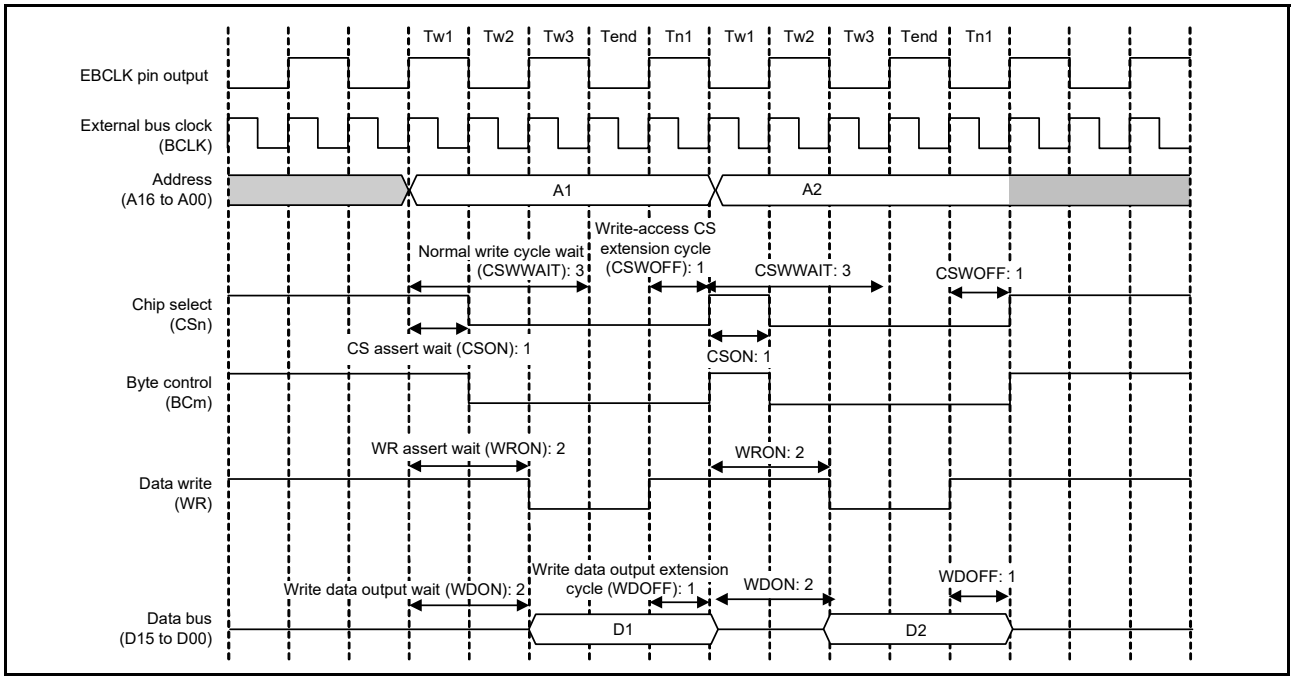


Figure 15.16 Example of normal write operation when BCLK/2 is selected in the EBCLK Pin Output Select bit and two rounds of bus access are generated in response to a single transfer request (n = 0 to 3, m = 0, 1)

(2) Page Access

When the PRENB and PWENB bits in CSnMOD are set to 1 to enable page read and page write access, the bus access for page access operations becomes page reading and writing. Page access can only occur when two or more rounds of external bus access are required for a single transfer request from the bus master. See Figure 15.3 to Figure 15.6 for the conditions under which page access occurs.

Figure 15.17 and Figure 15.18 show examples of page access operations.

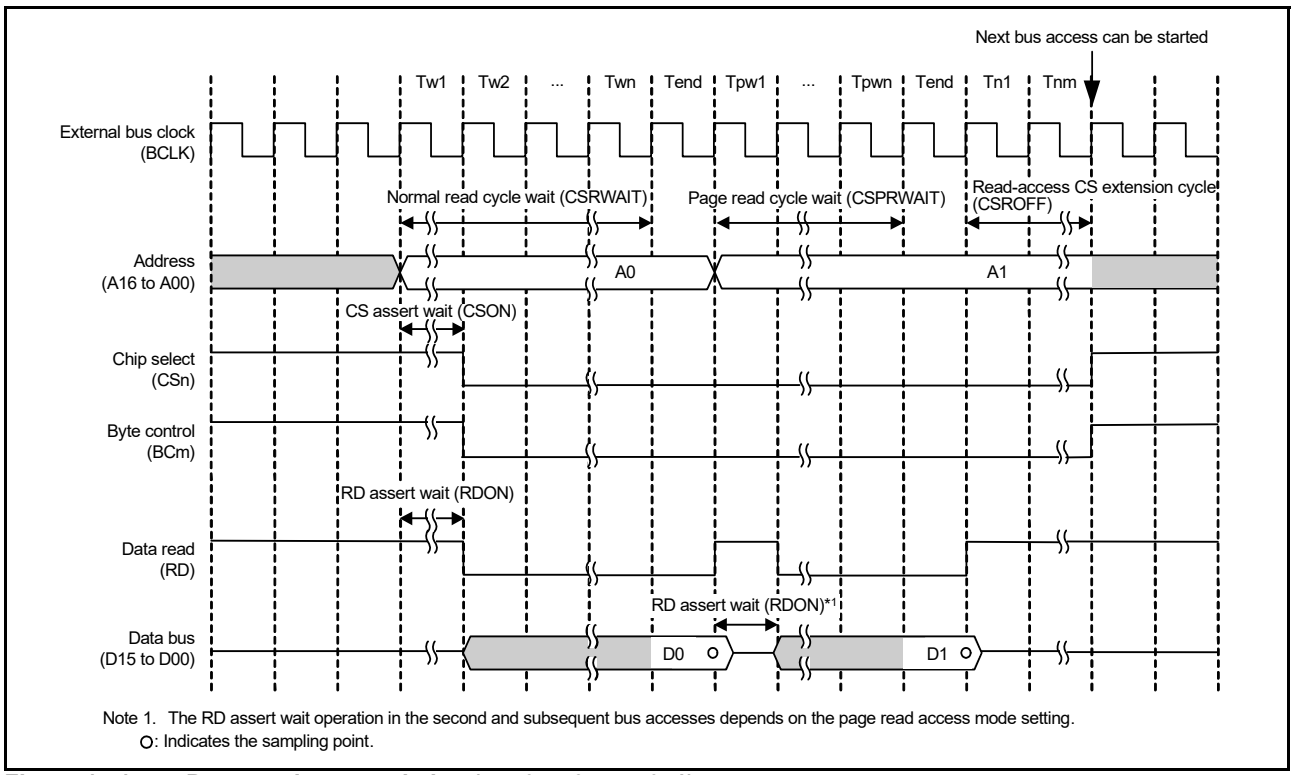


Figure 15.17 Page read access timing (n = 0 to 3, m = 0, 1)

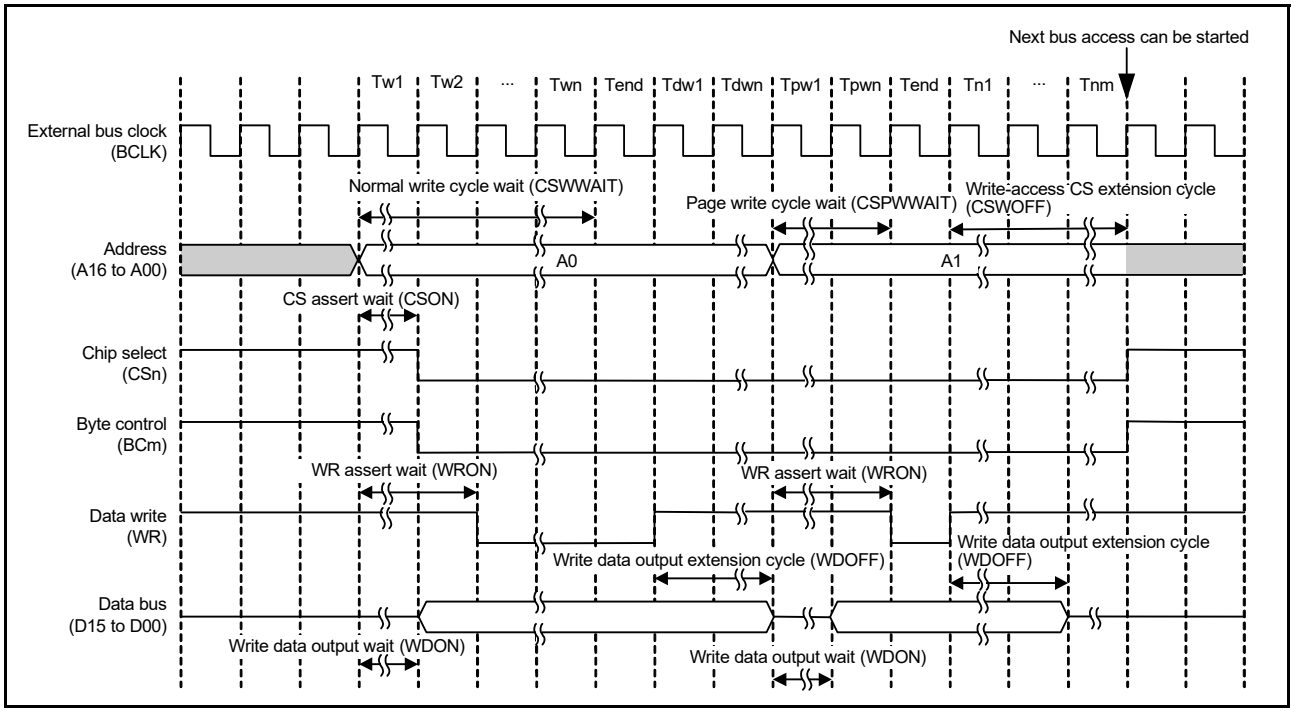


Figure 15.18 Page write access timing (n = 0 to 3, m = 0, 1)

Figure 15.19 and Figure 15.20 show examples of operations for access to a 16-bit bus space in 32 bits. The values of the wait control registers shown in the figures are example settings. In your application, set the registers according to the specifications of connected devices.

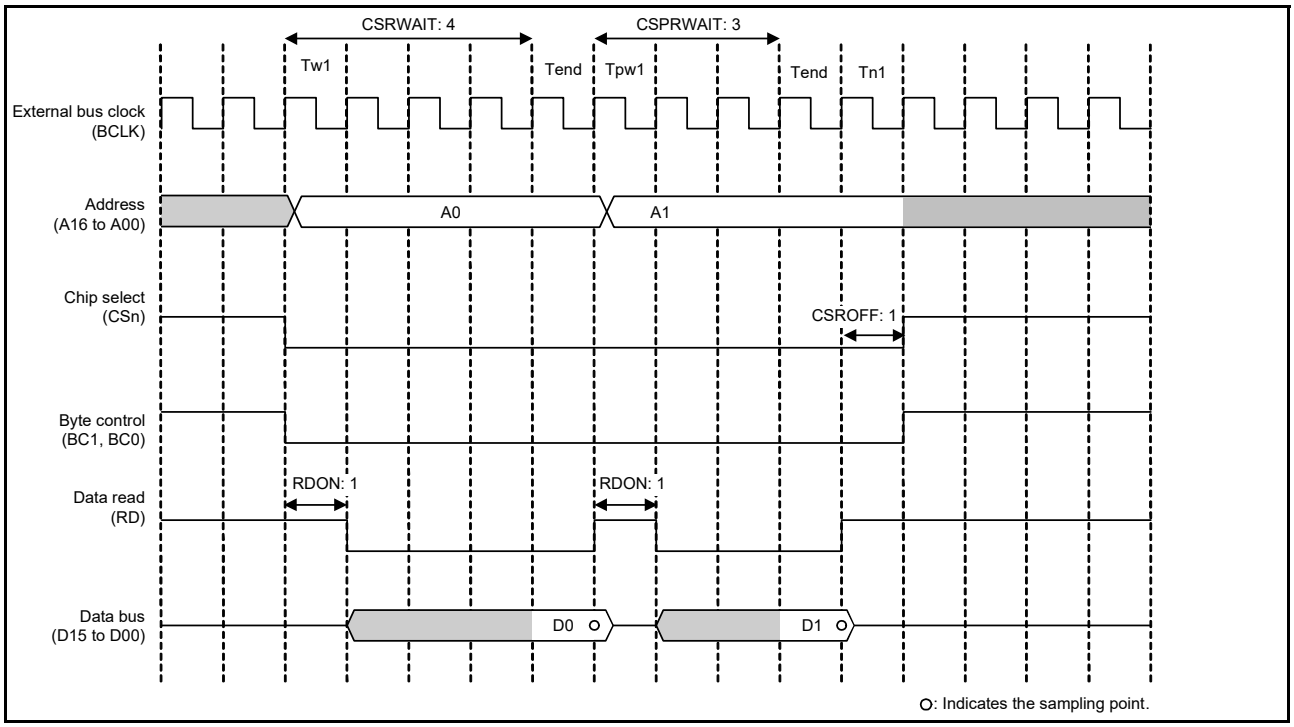


Figure 15.19 Example of page read access operation when 16-bit bus space is accessed in 32 bits (n = 0 to 3)

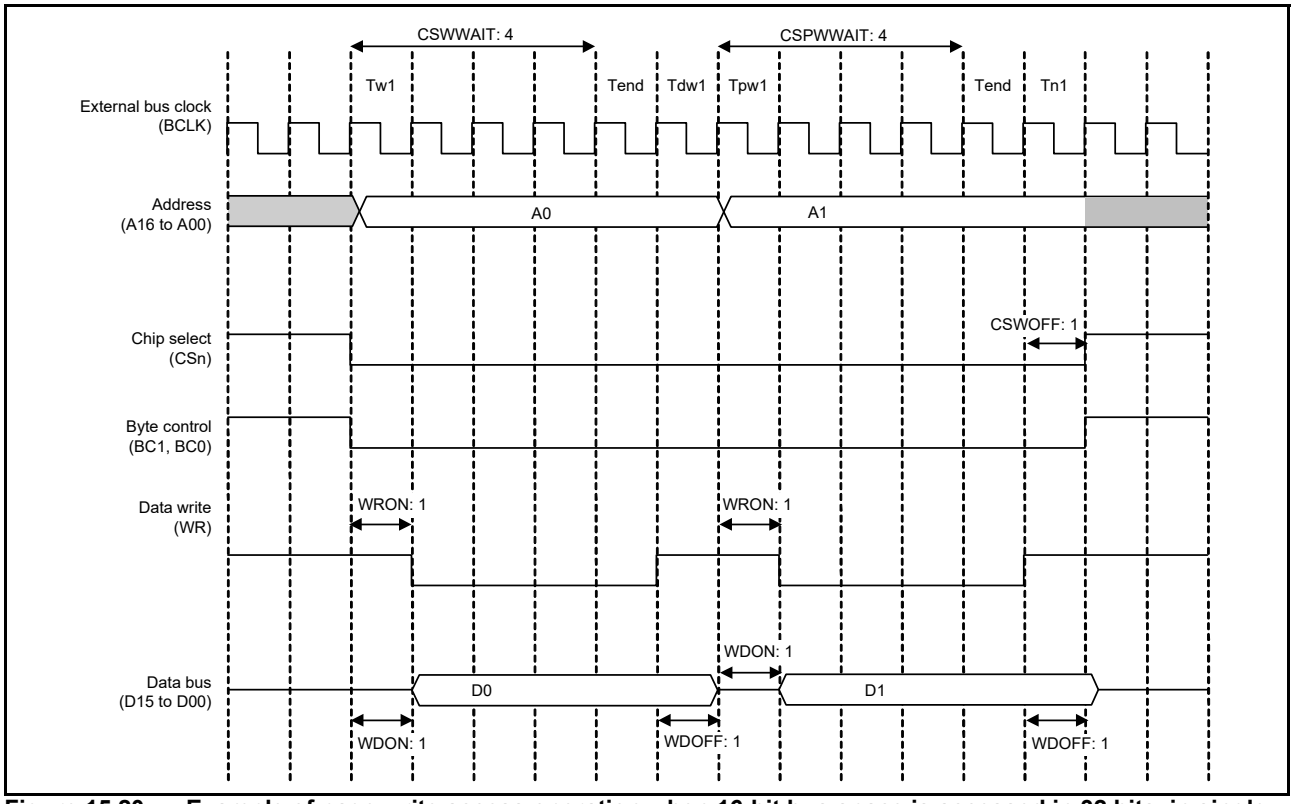


Figure 15.20 Example of page write access operation when 16-bit bus space is accessed in 32 bits, in single write strobe mode (n = 0 to 3)

Figure 15.21 and Figure 15.22 show examples of page access operations performed with the BCLK/2 is selected in the EBCLK Pin Output Select bit.

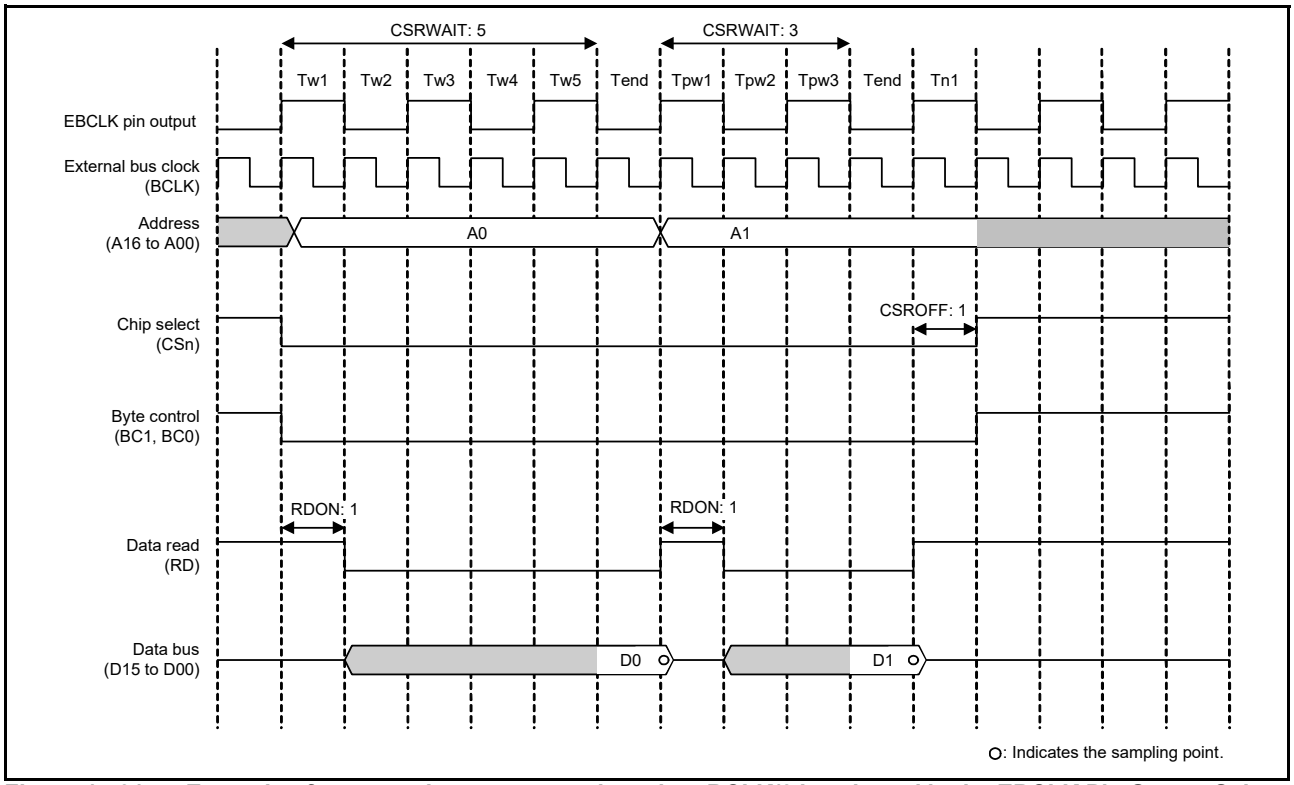


Figure 15.21 Example of page read access operation when BCLK/2 is selected in the EBCLK Pin Output Select bit and two rounds of bus access are generated in response to a single transfer request (n = 0 to 3)

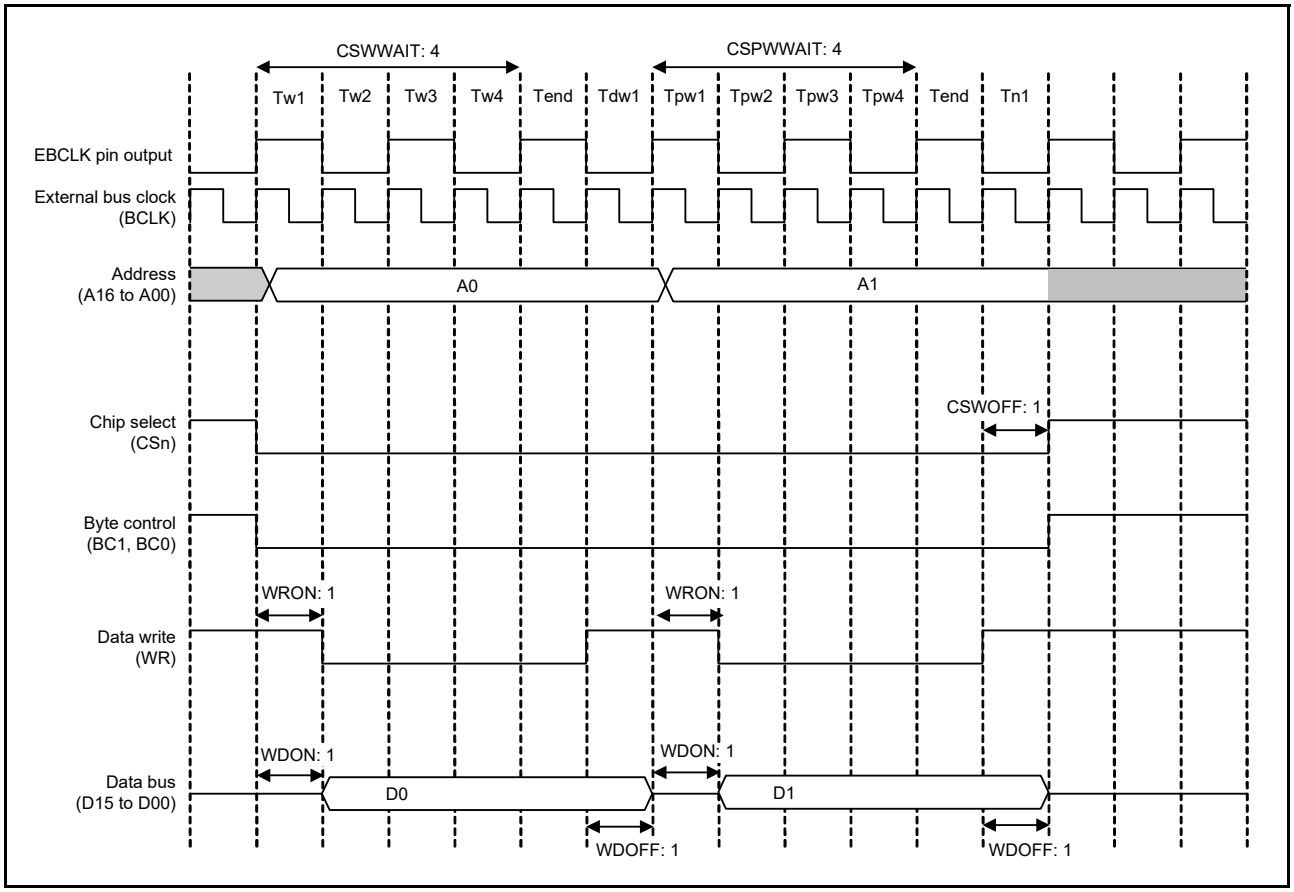


Figure 15.22 Example of page write access operation when BCLK/2 is selected in the EBCLK Pin Output Select bit and two rounds of bus access are generated in response to a single transfer request, in single write strobe mode (n = 0 to 3)

15.5.2 External Wait Function

Wait cycles can be extended by the WAIT signal beyond the length of normal access cycle wait specified in the CSRWAIT[4:0] and CSWWAIT[4:0] bits in CSnWCR1, and the page access cycle wait specified in the CSPRWAIT[2:0] and CSPWWAIT[2:0] bits in CSnWCR1.

When external wait is enabled (EWENB bit = 1 in CSnMOD), wait cycles are inserted while the WAIT signal is held low. When external wait is disabled (EWENB bit = 0 in CSnMOD), the WAIT signal has no effect. All wait cycles specified in CSnWCR1 are inserted independently of the WAIT signal. When external wait is enabled (EWENB bit = 1 in CSnMOD), BCLK and EBCLK must be operated at the same frequency.

(1) Normal Access

Sampling of the WAIT signal begins on completion of the wait cycle (Tend) specified in CSnWCR1. The bus cycle is extended while the WAIT signal is held low. The wait cycle ends (Tend) at the next cycle after the WAIT signal goes high.

(2) Page Access

The first access operation is the same as the normal access operation. Sampling of the WAIT signal begins on completion of the wait cycle (Tend) specified in the CSnWCR1 register. The bus cycle is extended while the WAIT signal is held low. The wait cycle ends (Tend) at the next cycle after the WAIT signal goes high.

For the second and subsequent accesses, sampling of the WAIT signal begins on completion of the page access wait cycle (Tend). The page access wait cycle is extended while the WAIT signal is held low, and ends (Tend) at the next cycle after the WAIT signal goes high.

Figure 15.23 and Figure 15.24 show examples of external wait insertion timing with the separate bus interface.

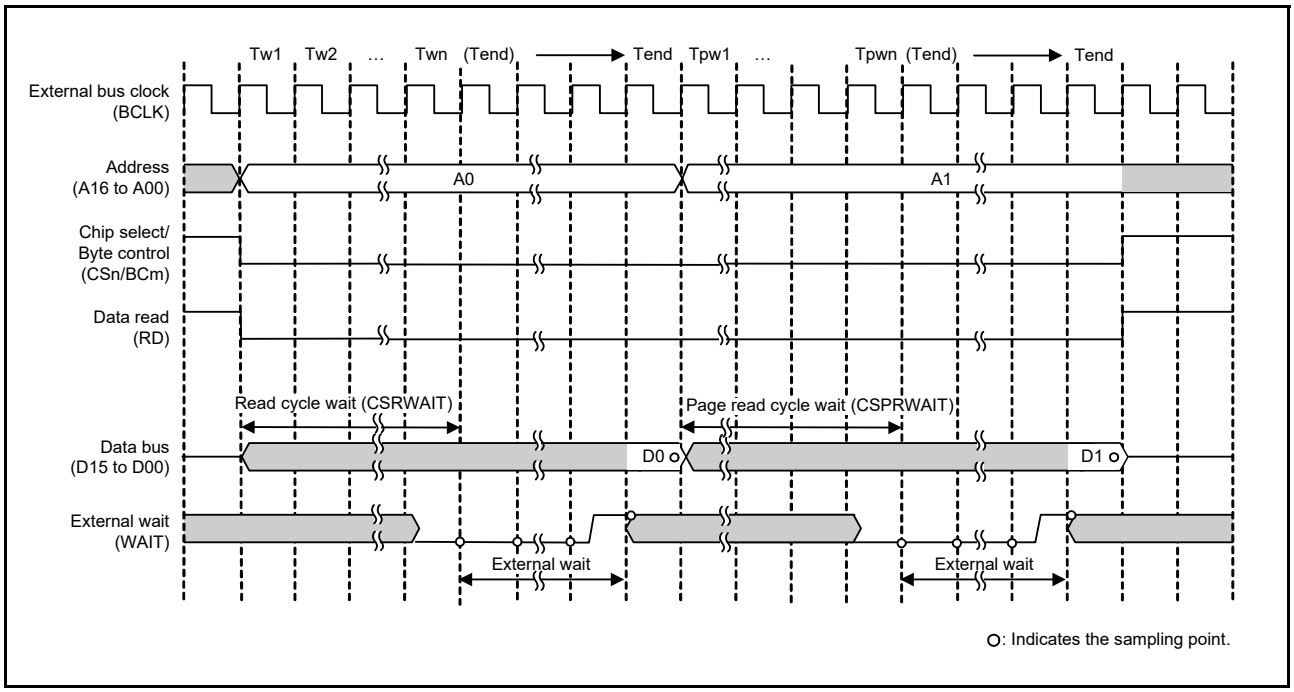


Figure 15.23 Example of external wait timing for page read access to 16-bit bus space (n = 0 to 3, m = 0, 1)

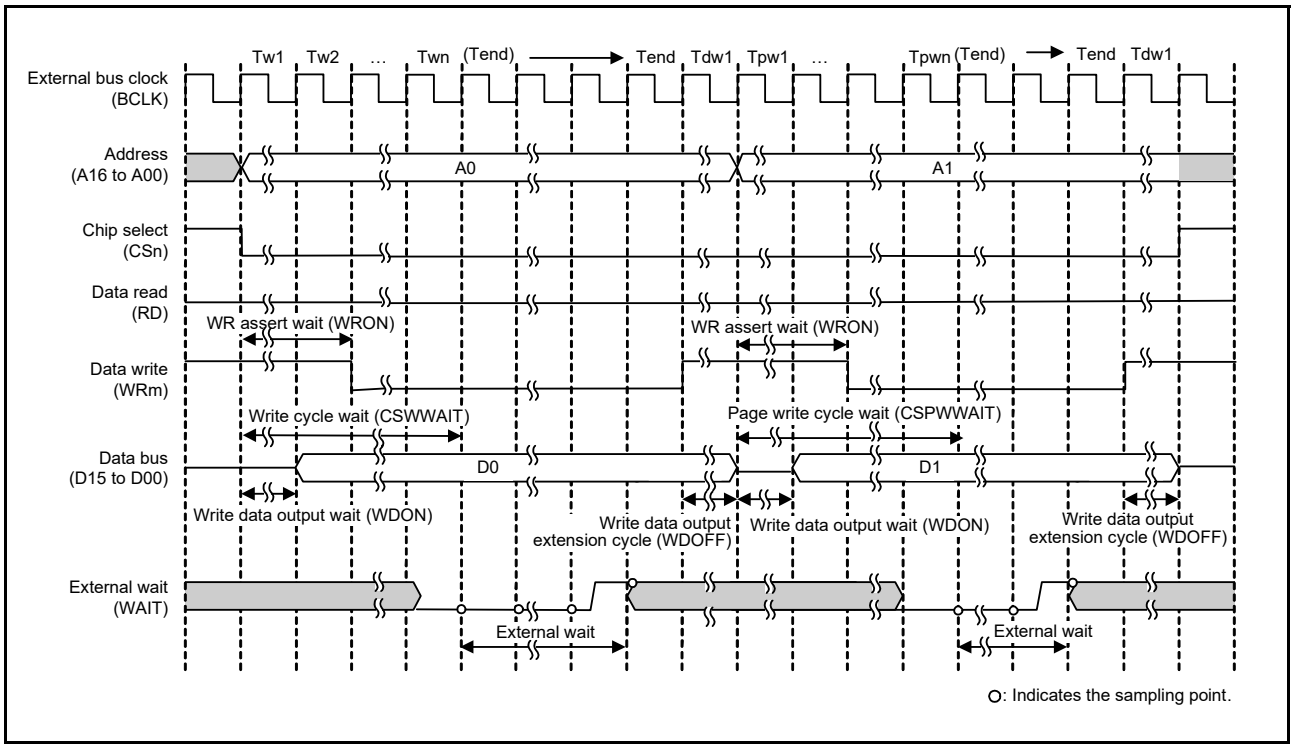


Figure 15.24 Example of external wait timing for page write access to 16-bit bus space, in byte strobe mode (n = 0 to 3, m = 0, 1)

15.5.3 Insertion of Recovery Cycles

Recovery cycles can be inserted between consecutive rounds of external bus access by setting the Recovery Cycle Insertion Enable bit in CSRECEN to 1.

The number of recovery cycles to be inserted after read cycles and write cycles can be independently set for each area using CSnREC. When the preceding bus cycle is a write access, the number of write recovery cycles must be set in the WRCV[3:0] bits for the associated area. When the preceding bus cycle is a read access, the number of read recovery cycles must be set in the RRCV[3:0] bits for the associated area. For example, when a CS1 read access occurs after a CS0 read access, the number of recovery cycles to be inserted between them is set in the RRCV[3:0] bits in CS0REC.

The recovery cycle insertion can be enabled or disabled with RCVENj (j = 0 to 7) in CSRECEN when the preceding bus access is a separate bus access. Recovery cycles can be inserted on any of the following eight conditions:

- After a read access to the external bus, a read access is made to the external bus in the same area
- After a read access to the external bus, a read access is made to the external bus in a different area
- After a read access to the external bus, a write access is made to the external bus in the same area
- After a read access to the external bus, a write access is made to the external bus in a different area
- After a write access to the external bus, a read access is made to the external bus in the same area
- After a write access to the external bus, a read access is made to the external bus in a different area
- After a write access to the external bus, a write access is made to the external bus in the same area
- After a write access to the external bus, a write access is made to the external bus in a different area.

The recovery cycle starts at the end of the preceding bus cycle, for example when the CSn signal (n = 0 to 3) is negated. A high-level period of the CSn signal is inserted for the specified recovery cycle period starting from this point.

In the fastest case, the CSn signal for the next round of bus access is asserted immediately after the end of the recovery cycles. Even if the next request for access to an external address space is generated during the recovery period, the next access over the external bus starts immediately after the end of the recovery cycles.

When two or more external bus access cycles are required for a single transfer request from a bus master, and the recovery cycle insertion condition is satisfied, recovery cycles are also inserted between these bus access cycles. However, when page read access is enabled (CSnMOD.PRENB = 1) or page write access is enabled (CSnMOD.PWENB = 1), recovery cycles are not inserted except after the last bus access cycle of the transfer, even if the recovery cycle insertion condition is satisfied. See Figure 15.27.

Similarly, during normal access with page access enabled, recovery cycles are not inserted between bus access cycles but only after the last bus access cycle of the transfer. When the recovery cycle insertion condition is satisfied, recovery cycles are inserted between bus access cycles regardless of the page access enable setting.

Figure 15.25 to Figure 15.27 show examples of recovery cycle insertion with the separate bus interface.

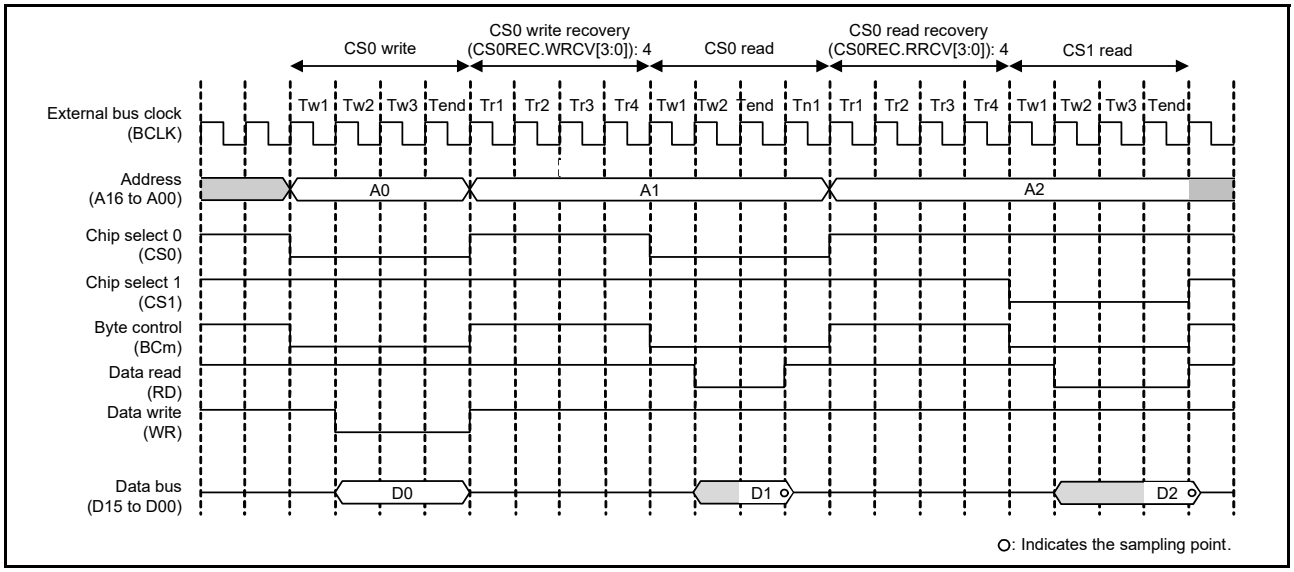


Figure 15.25 Example of recovery cycle insertion with separate bus interface (m = 0, 1)

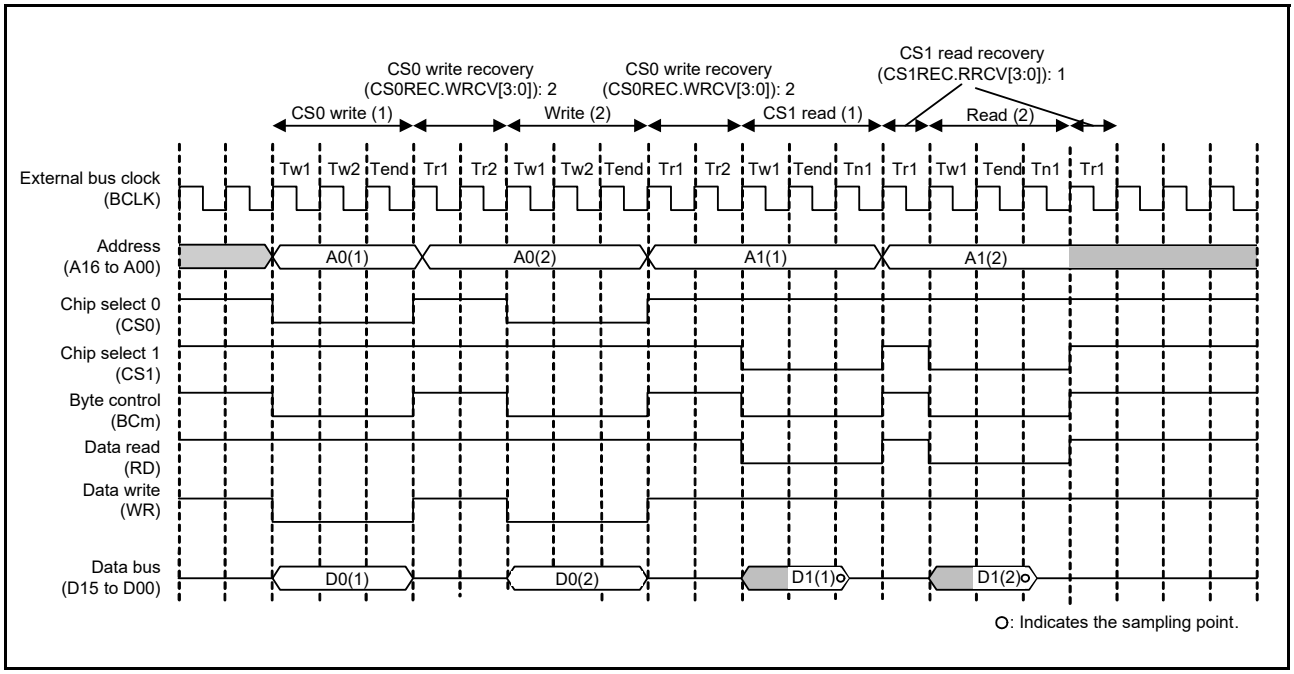


Figure 15.26 Example of recovery cycle insertion when bus access is split with separate bus interface and normal access (m = 0, 1)

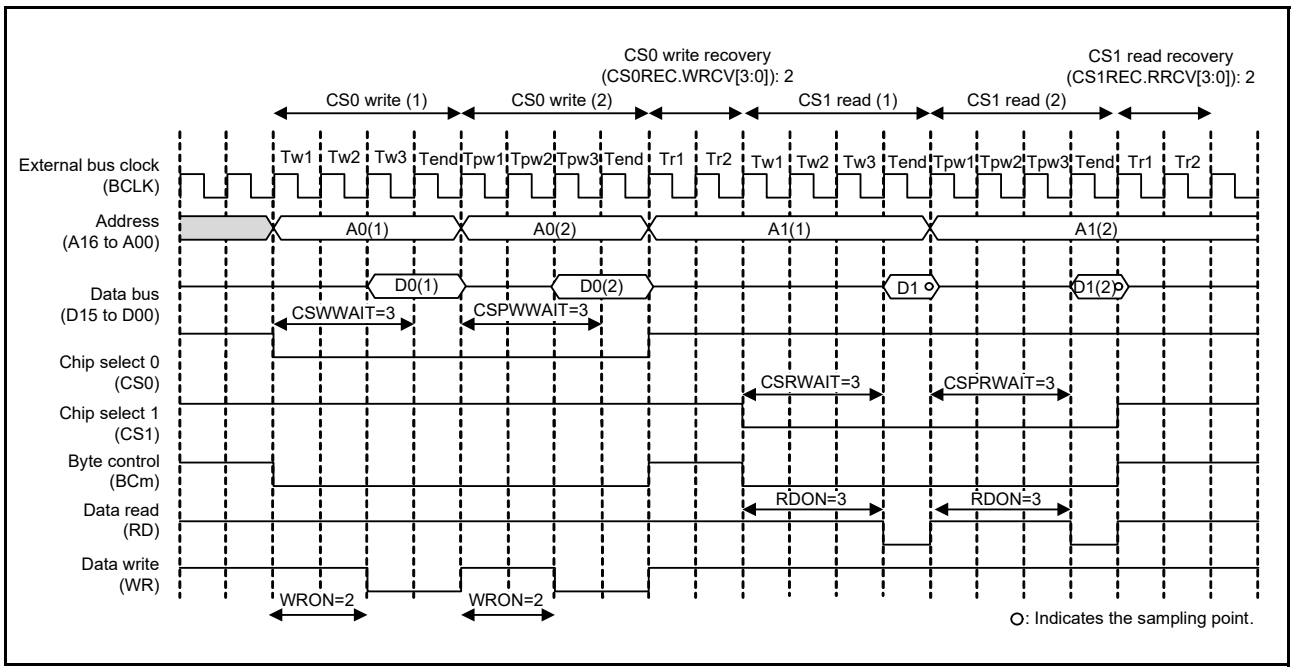


Figure 15.27 Example of recovery cycle insertion when bus access is split with separate bus interface, page access (m = 0, 1)

Figure 15.28 shows example of operation when the EBCLK Pin Output Selection bits are set for a frequency division of BCLK/2.

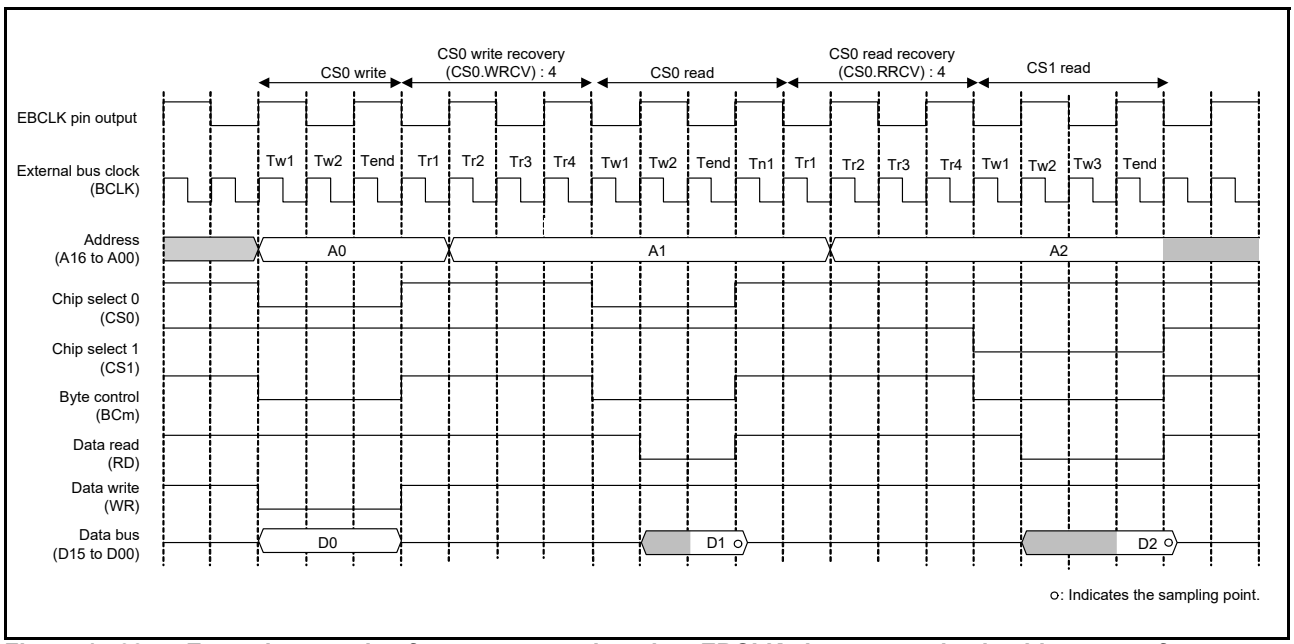


Figure 15.28 Example operation for recovery cycles when EBCLK pin output selection bits are set for frequency division of BCLK/2 for normal access through a separate bus interface (m = 0, 1)

15.5.4 No Access State

When no external address space is accessed, the CSn, BCn, WRn, and RDn signals are high, and D15 to D00 are in the high-impedance state.

15.5.5 Write Buffer Function (External Bus)

In write access, the main bus is released by writing data to the write buffer before the write access completes. This allows the next round of bus access to start. However, if the next access is to an external address space or to a register of the external bus controller, it is suspended until the external bus operations already in progress are complete.

Figure 15.29 shows an example of operation when the write buffer function is in use. When this function is in use, if the next operation after an external write is an internal access, the internal access is executed in parallel with the external write, for example without waiting for completion of the latter operation.

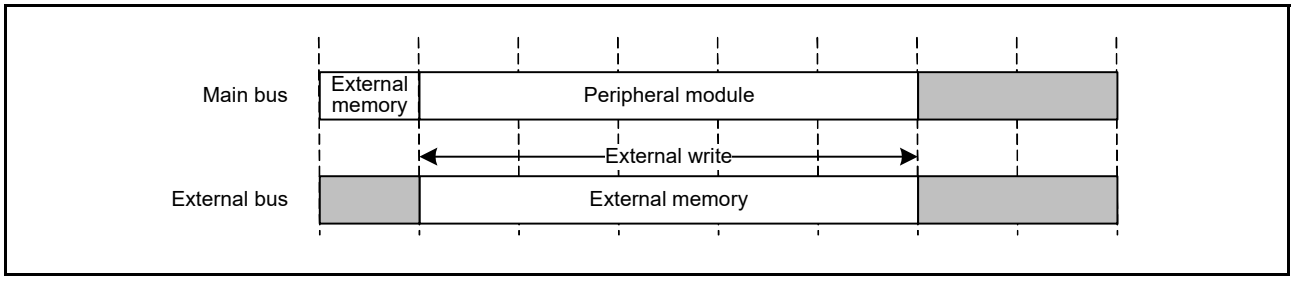


Figure 15.29 Example operation when the write buffer function is in use

15.5.6 Constraints

(1) Constraints on using separate bus interface

Table 15.10 lists the constraints that apply to bits in the CSn Wait Control Register 1 (CSnWCR1) and CSn Wait Control Register 2 (CSnWCR2) when normal and page accesses occur.

Even if the Page Read Access Enable bit or Page Write Access Enable bit in the CSn Mode Register is set to enable (CSnMOD.PRENB = 1 or CSnMOD.PWENB = 1), the first page access or access that does not fall within the scope of a page access is a normal access operation. Because of this, constraints on normal access must be satisfied.

Table 15.10 Constraints on normal access and page access

Constraints on normal access		Constraints on page access	
Reading	**Writing**	**Reading**	**Writing**
CSON[2:0] ≤ CSRWAIT RDON[2:0] ≤ CSRWAIT CSON[2:0] ≤ RDON	1 ≤ WDON[2:0] CSON[2:0] ≤ CSWWAIT WRON[2:0] ≤ CSWWAIT WDON[2:0] ≤ CSWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON	CSON[2:0] ≤ CSPRWAIT RDON[2:0] ≤ CSPRWAIT CSON[2:0] ≤ RDON	1 ≤ WDON[2:0] CSON[2:0] ≤ CSPWWAIT WRON[2:0] ≤ CSPWWAIT WDON[2:0] ≤ CSPWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON

Note: When two or more external bus access cycles are required for a single transfer request from a bus master, and the recovery cycle insertion condition is satisfied, with page read access enabled (CSnMOD.PRENB = 1) or page write access enabled (CSnMOD.PWENB = 1), recovery cycles are not inserted between bus access cycles, but are inserted only after the last bus access cycle of the transfer.

(2) Constraint on pin multiplex between the A00 and BC0 functions

Setting the single write strobe mode is prohibited in the 8-bit bus space.

(3) Constraints when BCLK/2 is selected in the EBCLK Pin Output Select bit

When BCLK/2 is selected in the EBCLK Pin Output Select bit, the external bus access cycle starts on the rising edge of the EBCLK pin output. However, when two or more external bus access cycles are generated for a single transfer request from a bus master, the second or subsequent external bus access cycle can start on the falling edge of the EBCLK pin output, depending on the wait cycle settings. Use the appropriate register settings according to the specifications of connected devices. When BCLK/2 is selected in the EBCLK Pin Output Select bit, enabling an external wait (CSnMOD.EWENB = 1) is prohibited.

(4) Restriction on instruction code

You must fix the instruction code to little endian order.

15.6 Bus Error Monitoring Section

This 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.

15.6.1 Error Type that Occurs by Bus

The following errors can occur on each bus:

- Illegal address access
- Bus master MPU error
- Bus slave MPU error
- Timeout.

Table 15.11 lists the address ranges where access leads to illegal address access errors. However, 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 16, Memory Protection Unit (MPU).

15.6.2 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 errors that occur for each master are stored in the BUSnERRADD and BUSnERRSTAT registers. These registers must be cleared by reset only. For more information, see sections 15.3.9 and 15.3.10.

Note: The DMAC and DTC do not receive bus errors. If the DMAC or DTC accesses the bus, the transfer continues.

15.6.3 Conditions Leading to Illegal Address Access Errors

Table 15.11 lists the address spaces for each bus that trigger illegal address access errors.

Table 15.11 Conditions leading to illegal address access errors

Address	Slave bus name	Master bus	
		CPU (ICode/DCode/ System)	DMA
0000 0000h to 01FF FFFFh	Memory bus 1 Memory bus 3	—	—
0200 0000h to 027F FFFFh	Memory mapping area	*1	E
0280 0000h to 1FFF FFFFh	Reserved	E	E
2000 0000h to 2001 FFFFh	Memory bus 4	—	—
2002 0000h to 2002 FFFFh	Memory bus 5	—	—
2003 0000h to 3FFF FFFFh	Reserved	E	E
4000 0000h to 4001 FFFFh	Peripheral bus 1	—	—
4002 0000h to 4003 FFFFh	Reserved	E	E
4004 0000h to 4005 FFFFh	Peripheral bus 3	—	—
4006 0000h to 4007 FFFFh	Peripheral bus 4	—	—
4008 0000h to 4009 FFFFh	Peripheral bus 5	—	—
400A 0000h to 400B FFFFh	Reserved	—	—
400C 0000h to 400D FFFFh	Peripheral bus 7	—	—
400E 0000h to 400F FFFFh	Reserved	E	E
4010 0000h to 407F FFFFh	Peripheral bus 9	—	—
4080 0000h to 5FFF FFFFh	Reserved	E	E
6000 0000h to 67FF FFFFh	QSPI area	—	—
6800 0000h to 7FFF FFFFh	Reserved	E	E
8000 0000h to 97FF FFFFh	CS area	—	—
9800 0000h to DFFF FFFFh	Reserved	E	E
E000 0000h to FFFF FFFFh	System for Cortex-M4	—	E

E indicates the path where an illegal address access error occurs.

"—" indicates the path where an illegal address access error does not occur or the path where access does not occur.

Note: If MMF (Memory Mirror Function) is enabled, the access to mapped area (0200 0000h to 027F FFFFh) is switched to the user specific area (MMF output address = CPU output address + offset).
The bus module does not detect whether the MMF switched the address. Therefore if the MMF is enabled and the CPU accesses 0200 0000h, no error can occur (depending on the switched address). If the MMF is disabled and the CPU accesses 0200 0000h, the bus module can detect the error.

Note 1. The bus module does not detect whether the MMF switched the address. Therefore if the MMF is enabled and the CPU accesses 0200 0000h, no error occurs (depending on the switched address).
If the MMF is disabled and the CPU accesses 0200 0000h, the bus module can detect the error.

The bus module detects an access error resulting from access to reserved area, for example if no area has been assigned for the slave.
0280 0000h to 1FFF FFFFh: Access error detection.
0000 0000h to 01FF FFFFh: Memory bus 1 no access error detection.

15.6.4 Timeout

For some peripheral modules, a timeout error occurs with the module-stop function. When there is no response from the slave for a certain period of time, a timeout error is detected. A timeout error is returned to the requesting master IP using the AHB-Lite error response protocol.

15.7 Notes on using Flash Cache

When using flash cache through access from the CPU, Arm MPU should also be set to cacheable. See references 1. and 2. for more information.

15.8 References

1. *ARM®v7-M Architecture Reference Manual* (ARM DDI 0403D)
2. *ARM® Cortex®-M4 Devices Generic User Guide* (ARM DUI 0553A)
3. *ARM® AMBA 3 AHB-Lite Protocol v1.0 Specification* (ARM IHI 0033A).

16. Memory Protection Unit (MPU)

16.1 Overview

The MCU has four Memory Protection Units (MPUs) and a CPU stack pointer monitor function. Table 16.1 lists the supported MPU features and Table 16.2 shows the behavior of each MPU error detection.

Table 16.1 MPU specifications

Classification	Module/Function	Description
Illegal memory access	Arm® Cortex®-M4 CPU	• Arm CPU has a default memory map. If the CPU makes an illegal access, an exception interrupt occurs. • The 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 subregions and background region.
	Bus master MPU	Memory protection function for each bus master except for the CPU: • Bus master MPU group A: 16 regions.
	Bus slave MPU	Memory protection function for each bus slave
Security	Security MPU	Protects accesses from non-secure programs to the following secure regions: • 2 regions (PC) • 1 region (code flash).

Table 16.2 MPU error detection behavior

MPU type	Notice method	Bus access on 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.	Stored in the Cortex-M4 processor
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 • Hard fault.	• 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 Arm MPU, see section 16.7, References. For information on error access for other MPUs, see section 15.3.9, Bus Error Address Register (BUSnERRADD) (n = 1 to 4) and section 15.3.10, Bus Error Status Register (BUSnERRSTAT) (n = 1 to 4) in section 15, Buses.

16.2 CPU Stack Pointer Monitor

The MCU provides a CPU stack pointer monitor that detects underflows and overflows of the stack pointer. Because the Arm CPU has two stack pointers, a Main Stack Pointer (MSP) and 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 bit in the Stack Pointer Monitor Access Control Register (MSPMPUCTL, PSPMPUCTL) to 1.

Table 16.3 lists the specifications of the CPU stack pointer monitor. Figure 16.1 shows the CPU stack pointer monitor block diagram and Figure 16.2 shows the register setting flow.

Table 16.3 CPU stack pointer monitor specifications

Parameter	Description
Protected region	SRAM region
Number of regions	2 regions (Main Stack Pointer (MSP), Process Stack Pointer (PSP))
Address specification for individual regions	Setting the address where regions start and end
Enable/disable setting for stack pointer monitor in individual regions	Settings enabled or disabled for the associated region
Operation on error detection	Reset or non-maskable interrupts
Register protection	Registers can be protected from illegal writes

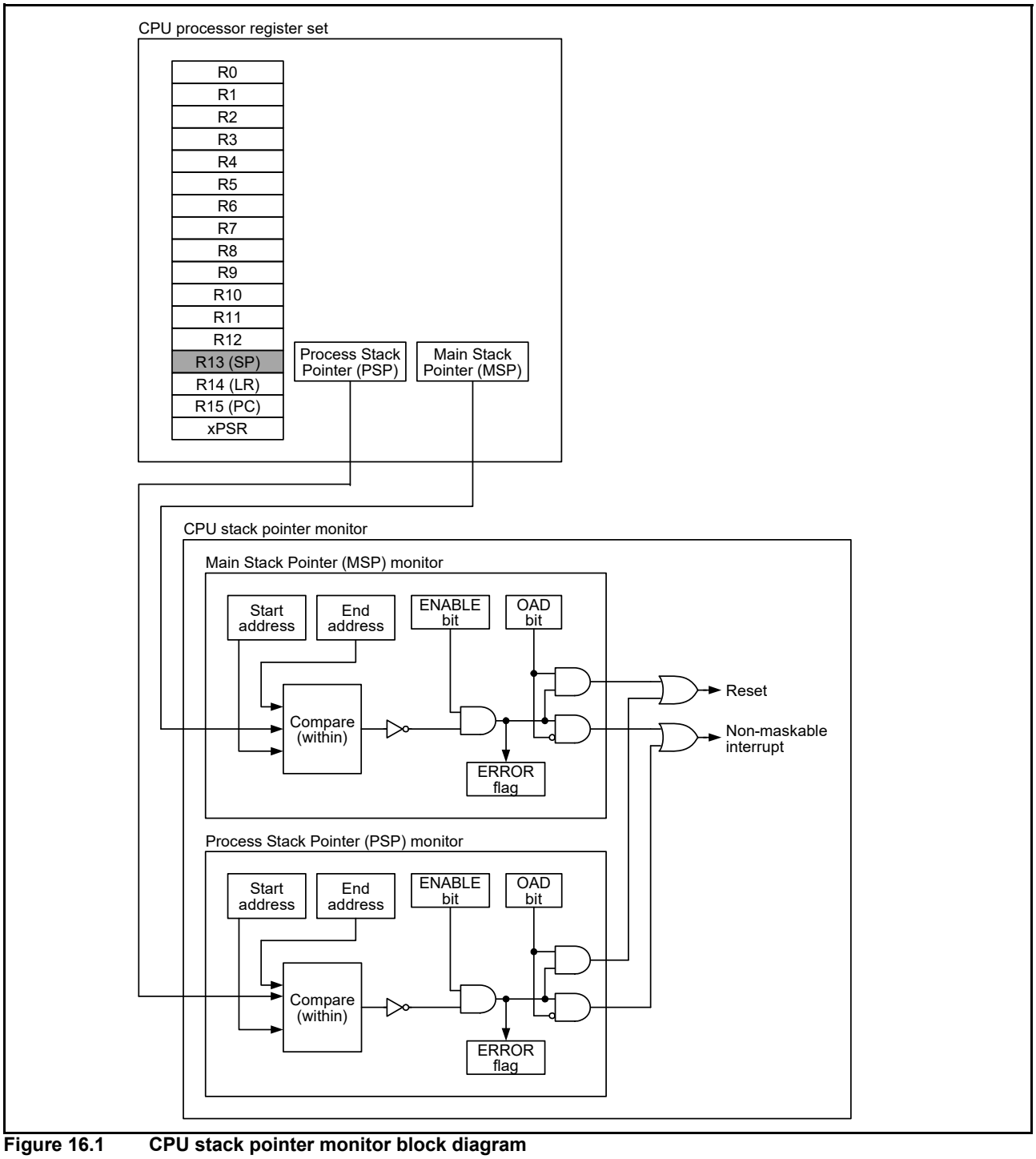


Figure 16.1 CPU stack pointer monitor block diagram

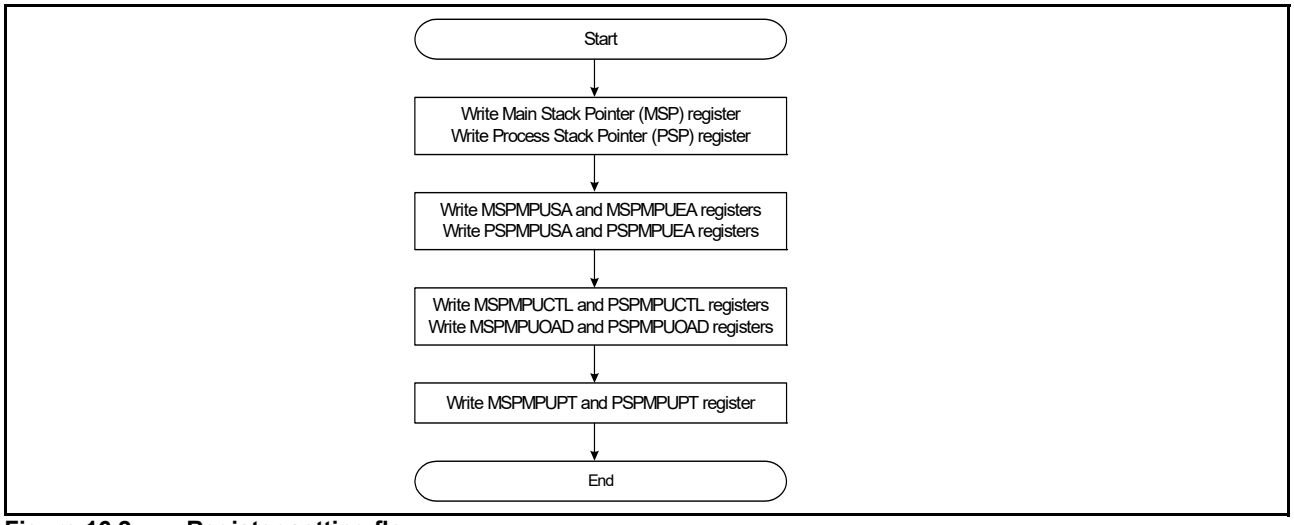


Figure 16.2 Register setting flow

16.2.1 Protection of Registers

Registers related to the CPU stack pointer monitor can be protected with the PROTECT bit.

16.2.2 Overflow/Underflow Error

If an overflow or underflow is detected, the CPU stack pointer monitor generates an overflow or underflow error. The memory protection error is selectable to a reset or a non-maskable interrupt in the OAD bit setting.

The status of the non-maskable interrupt is indicated in the ICU.NMISR.SPEST. For details, see section 14, Interrupt Controller Unit (ICU). The status of reset is indicated in the SYSTEM.RSTSR1.SPERF. For details, see section 6, Resets. When ICU.NMISR.SPEST indicates that a CPU stack pointer monitor interrupt occurred, check the ERROR bit in the 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 keeps the output when the stack pointer overflows or underflows. When a non-maskable interrupt flag is cleared, the stack pointer is set after ICU.NMICLR.SPECLR bit is 1. Then, write 0 to clear the ERROR bit in the MSPMPUCTL and PSPMPUCTL registers.

16.2.3 Register Descriptions

Note: Bus access must be stopped before writing to MPU registers.

16.2.3.1 Main Stack Pointer (MSP) Monitor Start Address Register (MSPMPUSA)

Address(es): SPMON.MSPMPUSA 4000 0D08h

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

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

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	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 should be 2000 0000h to 200F FFFCh, not including reserved areas.	R/W

The MSPMPUSA and MSPMPUEA registers specify the CPU stack region of SRAM (2000 0000h to 200F FFFFh, not including the reserved areas. For SRAM area to be covered, see Figure 4.1 Memory map).

16.2.3.2 Main Stack Pointer (MSP) Monitor End Address Register (MSPMPUEA)

Address(es): SPMON.MSPMPUEA 4000 0D0Ch

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSPMPUEA[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	MSPMPUEA[31:0]	Region End Address	Address where the region ends to use for region determination. The lower 2 bits should be 1. The value range should be 2000 0003h to 200F FFFFh, not including the reserved areas.	R/W

16.2.3.3 Process Stack Pointer (PSP) Monitor Start Address Register (PSPMPUSA)

Address(es): SPMON.PSPMPUSA 4000 0D18h

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

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

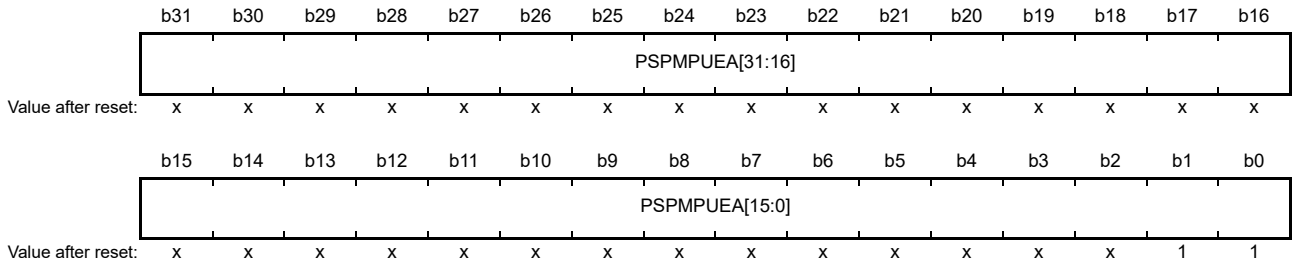
x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	PSPMPUSA[31:0]	Region Start Address	Address where the region starts to use for region determination. The lower 2 bits should be 0. The value range should be 2000 0000h to 200F FFFCh, not including reserved areas.	R/W

The PSPMPUSA and PSPMPUEA registers specify the CPU stack region of SRAM (2000 0000h to 200F FFFFh except reserved areas, not including the reserved areas. For SRAM area to be covered, see Figure 4.1 Memory map).

16.2.3.4 Process Stack Pointer (PSP) Monitor End Address Register (PSPMPUEA)

Address(es): SPMON.PSPMPUEA 4000 0D1Ch

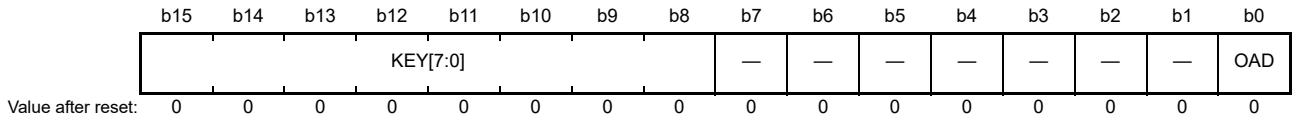


x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	PSPMPUEA[31:0]	Region End Address	Address where the region ends to use for region determination. The lower 2 bits should be 1. The value range should be 2000 0003h to 200F FFFFh, not including the reserved areas.	R/W

16.2.3.5 Stack Pointer Monitor Operation After Detection Register (MSPMPUOAD, PSPMPUOAD)

Address(es): SPMON.MSPMPUOAD 4000 0D00h, SPMON.PSPMPUOAD 4000 0D10h



Bit	Symbol	Bit name	Description	R/W
b0	OAD	Operation after Detection	0: Non-maskable interrupt 1: Reset.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits are used to enable or disable writing of the OAD bit.	R/(W)*1

Note 1. Write data is not retained.

OAD bit (Operation after Detection)

The OAD bit selects a reset or a non-maskable interrupt when a stack pointer underflow or overflow is detected by the CPU stack pointer monitor.

The main stack pointer monitor and the process stack pointer monitor each use the OAD bit to determine which signal is generated when a stack pointer underflow or overflow is detected. When writing to the OAD bit, write A5h simultaneously 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 OAD bit. When writing to the OAD bit, simultaneously write A5h to KEY[7:0]. When values other than A5h are written to the KEY[7:0] bits, the OAD bit is not updated. The KEY[7:0] bits are always read as 00h.

16.2.3.6 Stack Pointer Monitor Access Control Register (MSPMPUCTL, PSPMPUCTL)

Address(es): SPMON.MSPMPUCTL 4000 0D04h, SPMON.PSPMPUCTL 4000 0D14h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	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	Bit name	Description	R/W
b0	ENABLE	Stack Pointer Monitor Enable	0: Stack pointer monitor is disabled 1: Stack pointer monitor is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	ERROR	Stack Pointer Monitor Error Flag	0: Stack pointer has not overflowed or underflowed 1: Stack pointer has overflowed or underflowed.	R/W
b15 to b9	—	Reserved	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 bit (Stack Pointer Monitor Error Flag)

The ERROR bit indicates the state of the stack pointer monitor. Each stack point monitor has an independent ERROR bit.

[Setting condition]

- Overflow or underflow of stack pointer.

[Clearing conditions]

- 0 is written to this bit
- A reset other than the bus master MPU error reset, bus slave MPU error reset, and stack pointer error reset.

Note: Only 0 can be written to the ERROR bit.

16.2.3.7 Stack Pointer Monitor Protection Register (MSPMPUPT, PSPMPUPT)

Address(es): SPMON.MSPMPUPT 4000 0D06h, SPMON.PSPMPUPT 4000 0D16h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit name	Description	R/W
b0	PROTECT	Protection of Register	0: Stack Pointer Monitor register writes are possible 1: Stack Pointer Monitor register writes are protected. Read is possible.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits are used to 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 A5h to the KEY[7:0] bits, using halfword access.

KEY[7:0] bits (Key Code)

These bits enable or disable writing to the PROTECT bit.

When writing to the PROTECT bit, write A5h to KEY[7:0] simultaneously. When values other than A5h are written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0.

16.3 Arm MPU

The Arm MPU has 8 region memory protection units and provides full support for:

- Protection regions
- Overlapping protection regions, with ascending region priority:
  7 = highest priority
  0 = lowest priority.
- Access permissions
- Exporting memory attributes to the system.

Arm MPU mismatches and permission violations invoke the programmable-priority MemManage fault (HardFault) handler. For details, see section 16.7 2.

16.4 Bus Master MPU

The MCU incorporates a bus master MPU that monitors the addresses of bus master access to the overall address space (0000 0000h to FFFF FFFFh).

Access control information can be set for up to 16 regions and the bus master MPU monitors access to each region in association with this information. If access to the protected region is detected, the bus master MPU generates an internal reset or a non-maskable interrupt. For details information on error access, see 15.3.9 and 15.3.10 in section 15, Buses.

The supported access control information for the individual regions consists of permission to read and to write.

Table 16.4 lists the specifications of the bus master MPU.

Table 16.4 Bus master MPU specifications

Specifications	Description
Master groups	Bus master MPU group A: DMA bus
Protected regions	0000 0000h to FFFF FFFFh
Number of regions	Bus master MPU group A: 16 regions
Specifying addresses of individual regions	Setting the address where regions start and end
Setting to make memory protection effective or ineffective in individual regions	Settings effective or ineffective for the associated region
Access-control information settings for individual regions	Permission to read and to write
Operation after detection	Reset or non-maskable interrupts
Register protection	Register can be protected from illegal writes

Figure 16.3 shows a block diagram of the bus master MPU.

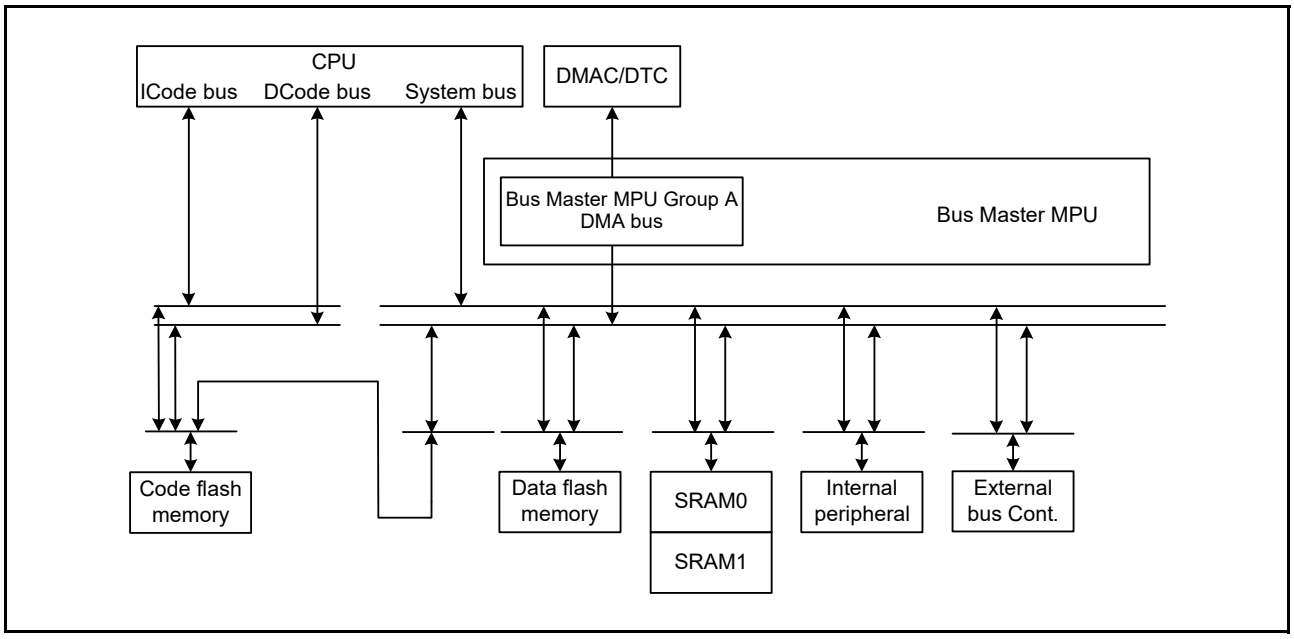


Figure 16.3 MPU bus master diagram

Figure 16.4 shows the MPU bus master group A.

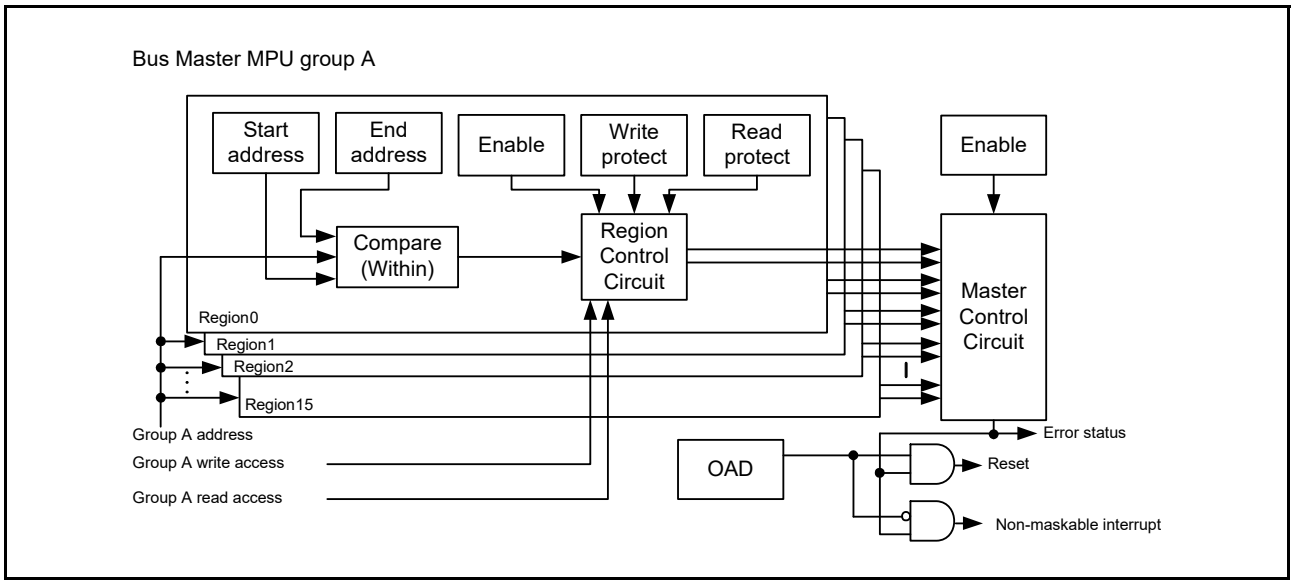


Figure 16.4 MPU bus master group A

16.4.1 Register Descriptions

Note: Stop bus access before processing register writes.

16.4.1.1 Group A Region n Start Address Register (MMPUSAn) (n = 0 to 15)

Address(es): MMPU.MMPUSA0 4000 0204h, MMPU.MMPUSA1 4000 0214h, MMPU.MMPUSA2 4000 0224h, MMPU.MMPUSA3 4000 0234h, MMPU.MMPUSA4 4000 0244h, MMPU.MMPUSA5 4000 0254h, MMPU.MMPUSA6 4000 0264h, MMPU.MMPUSA7 4000 0274h, MMPU.MMPUSA8 4000 0284h, MMPU.MMPUSA9 4000 0294h, MMPU.MMPUSA10 4000 02A4h, MMPU.MMPUSA11 4000 02B4h, MMPU.MMPUSA12 4000 02C4h, MMPU.MMPUSA13 4000 02D4h, MMPU.MMPUSA14 4000 02E4h, MMPU.MMPUSA15 4000 02F4h

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

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	MMPUSA[31:0]	Region Start Address	Address where the region starts to use for region determination. The lower 2 bits should be 0.	R/W

16.4.1.2 Group A Region n End Address Register (MMPUEAn) (n = 0 to 15)

Address(es): MMPU.MMPUEA0 4000 0208h, MMPU.MMPUEA1 4000 0218h, MMPU.MMPUEA2 4000 0228h, MMPU.MMPUEA3 4000 0238h, MMPU.MMPUEA4 4000 0248h, MMPU.MMPUEA5 4000 0258h, MMPU.MMPUEA6 4000 0268h, MMPU.MMPUEA7 4000 0278h, MMPU.MMPUEA8 4000 0288h, MMPU.MMPUEA9 4000 0298h, MMPU.MMPUEA10 4000 02A8h, MMPU.MMPUEA11 4000 02B8h, MMPU.MMPUEA12 4000 02C8h, MMPU.MMPUEA13 4000 02D8h, MMPU.MMPUEA14 4000 02E8h, MMPU.MMPUEA15 4000 02F8h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MMPUEA[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	MMPUEA[31:0]	Region End Address	Address where the region ends to use for region determination. The lower 2 bits should be 1.	R/W

16.4.1.3 Group A Region n Access Control Register (MMPUACAn) (n = 0 to 15)

Address(es): MMPU.MMPUACA0 4000 0200h, MMPU.MMPUACA1 4000 0210h, MMPU.MMPUACA2 4000 0220h, MMPU.MMPUACA3 4000 0230h, MMPU.MMPUACA4 4000 0240h, MMPU.MMPUACA5 4000 0250h, MMPU.MMPUACA6 4000 0260h, MMPU.MMPUACA7 4000 0270h, MMPU.MMPUACA8 4000 0280h, MMPU.MMPUACA9 4000 0290h, MMPU.MMPUACA10 4000 02A0h, MMPU.MMPUACA11 4000 02B0h, MMPU.MMPUACA12 4000 02C0h, MMPU.MMPUACA13 4000 02D0h, MMPU.MMPUACA14 4000 02E0h, MMPU.MMPUACA15 4000 02F0h

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

Bit	Symbol	Bit name	Description	R/W
b0	ENABLE	Region Enable	0: Group A region n unit disabled 1: Group A region n unit enabled.	R/W
b1	RP	Read Protection	0: Read permission 1: Read protection.	R/W
b2	WP	Write Protection	0: Write permission 1: Write protection.	R/W
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The group A region n unit each sets the ENABLE bit, the RP bit, and the WP bit each.

ENABLE bit (Region Enable)

The ENABLE bit enables or disables group A region n unit.

When the ENABLE bit is set to 1, the RP bit and the WP bit can be controlled for access permission or protection to the region that is set in MMPUSAn and MMPUEAn. When the ENABLE bit is set to 0, access to group A region n is without a region.

RP bit (Read Protection)

The RP bit enables or disables read protection of group A region n. When the ENABLE bit is set to 1, the RP bit is available.

WP bit (Write Protection)

The WP bit enables or disables write protection of group A region n. When the ENABLE bit is set to 1, the WP bit is available.

Table 16.5 Function of region control circuit

MMPUACAn. ENABLE	MMPUACAn. RP	MMPUACAn. WP	Access	Region	Output of Group A Region n unit
0	-	-	Read	-	Without region
			Write		Without region
1	0	0	Read	Within	Permission region
				Without	Without region
			Write	Within	Permission region
				Without	Without region
	0	1	Read	Within	Permission region
				Without	Without region
			Write	Within	Protection region
				Without	Without region
	1	0	Read	Within	Protection region
				Without	Without region
			Write	Within	Permission region
				Without	Without region
	1	1	Read	Within	Protection region
				Without	Without region
			Write	Within	Protection region
				Without	Without region

n = 0 to 15

Table 16.6 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 15 unit	Function of Group A
1	Protection region	*	*	Generate error
1	*	Protection region	*	Generate error
1	*	*	Protection region	Generate error
1	Without region	Without region	Without region	Generate error
Other case				No error

*: don't care

A master MPU error occurs on the following conditions:

- MMPUCTLA.ENABLE = 1, and output of one or more region n unit is to a protected region
- MMPUCTLA.ENABLE = 1, and output of all region n units is outside the region.

Other cases are for the permitted regions.

16.4.1.4 Bus Master MPU Control Register (MMPUCTLA)

Address(es): MMPU.MMPUCTLA 4000 0000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit name	Description	R/W
b0	ENABLE	Master Group Enable	0: Master Group A disabled 1: Master Group A enabled.	R/W
b1	OAD	Operation After Detection	0: Non-maskable interrupt 1: Reset.	R/W
b7 to b2	—	Reserved	These bits are read as 0.The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits are used to enable or disable writing of the OAD and ENABLE bit.	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 the ENABLE bit is set, write A5h in halfword access in KEY[7:0] simultaneously.

OAD bit (Operation After Detection)

The OAD bit generates either a reset or non-maskable interrupt when access to the protect region is detected by the bus master MPU.

When the OAD bit is set, write A5h in halfword access in KEY[7:0] simultaneously.

KEY[7:0] bits (Key Code)

These bits are used to enable or disable writing of the ENABLE and OAD bit.

When writing to the ENABLE and OAD bits, write A5h to KEY[7:0] simultaneously. When values other than A5h are written to the KEY[7:0] bits, the ENABLE and the OAD bits are not updated. The KEY[7:0] bits always read as 00h.

16.4.1.5 Group A Protection of Register (MMPUPTA)

Address(es): MMPU.MMPUPTA 4000 0102h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit name	Description	R/W
b0	PROTECT	Protection of register	0: All bus master MPU group A register writing is possible. 1: All bus master MPU group A register writing is protected. Read is possible.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits are used to enable or disable writing of the PROTECT bit.	R/(W)*1

Note 1. Write data is not retained.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writing 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, write A5h in halfword access in KEY[7:0] simultaneously.

KEY[7:0] bits (Key Code)

These bits enable or disable writing of the PROTECT bit. When writing to the PROTECT bit, write A5h to KEY[7:0] simultaneously. When values other than A5h are written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits always read as 00h.

16.4.2 Functions

(1) Memory Protection

The bus master MPU monitoring functions with access control information that is set for the individual access control regions. If access to the protected region is detected, the bus master MPU generates a memory protection error.

The bus master MPU can be set for up to 16 protection regions. A region where the permission and protection regions overlap is a protection region and a region with two overlapping permission regions is a protection region.

The bus master MPU has Group A. The memory protection checks the address of the bus for a unified master group and all accesses of a master group are protected by memory protection. The bus master MPU sets the permission for all the regions after reset. All regions are protected by setting MMPUCTLA.ENABLE to 1. Each region sets up a permission region on the protection region. If access to the protected region is detected, the bus master MPU generates an error.

Figure 16.5 shows the use case of a bus master MPU.

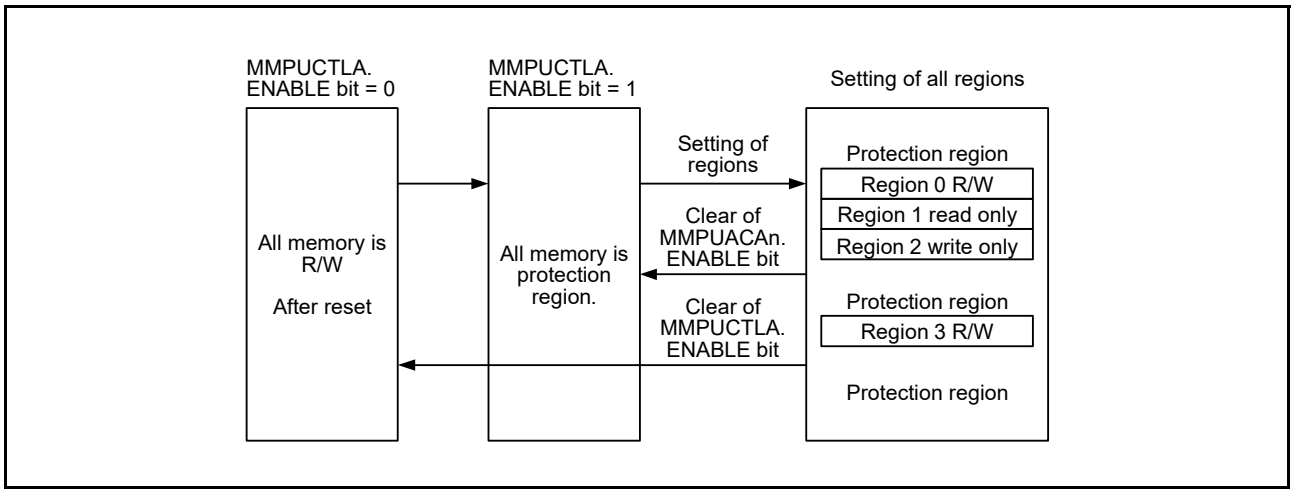


Figure 16.5 Use case of bus master MPU

Figure 16.6 shows the access permission or protection by the overlap of the bus master MPU region.

The control of the access permission or protection by the overlap of region is as follows.

1. Protection region when output of one or more region unit is the protection region.
2. Protection region when output of all region unit are without region.
3. Other cases are permission region.

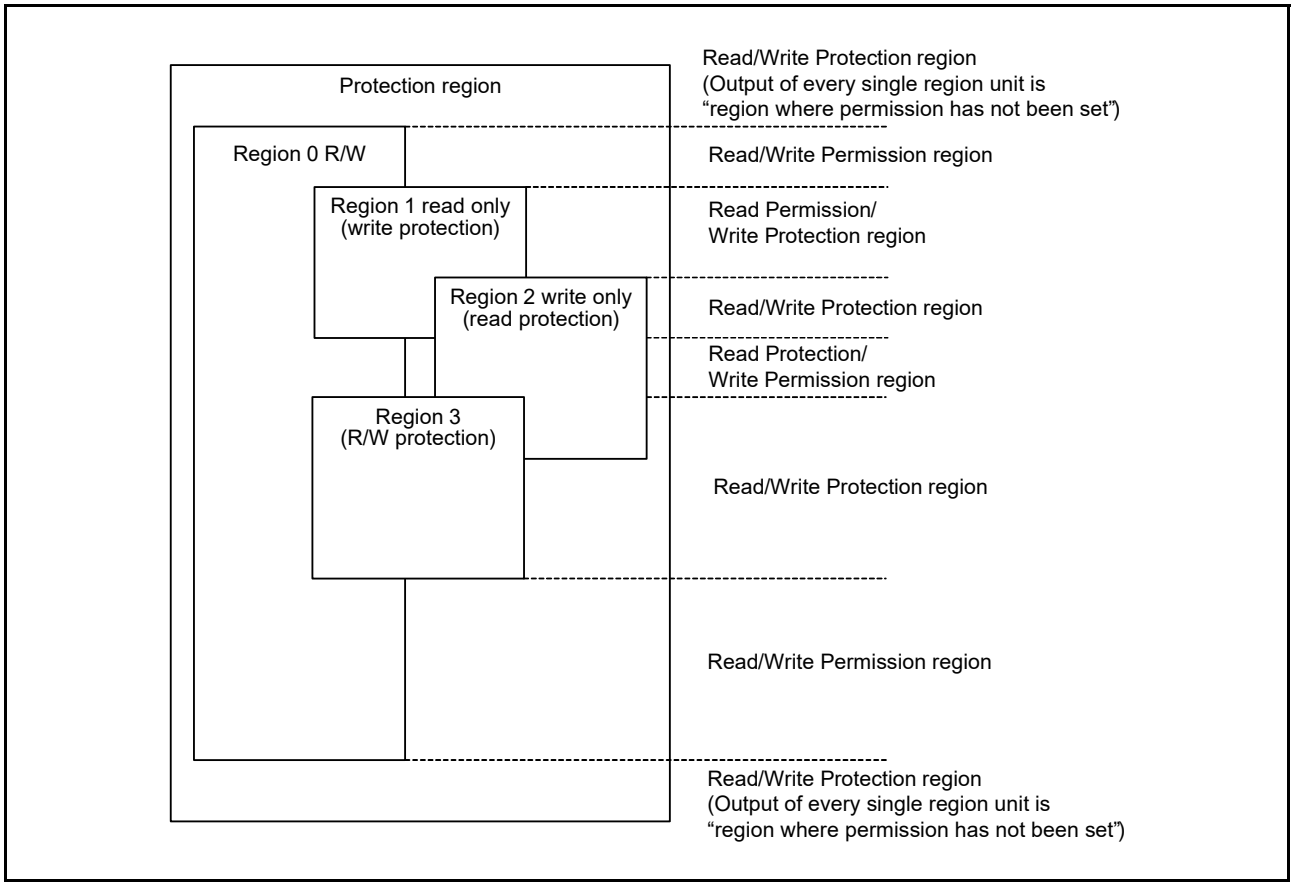


Figure 16.6 Access permission or protection by overlap of the bus master MPU region

Figure 16.7 shows the register setting flow after reset. During this register setting, stop the master except the CPU.

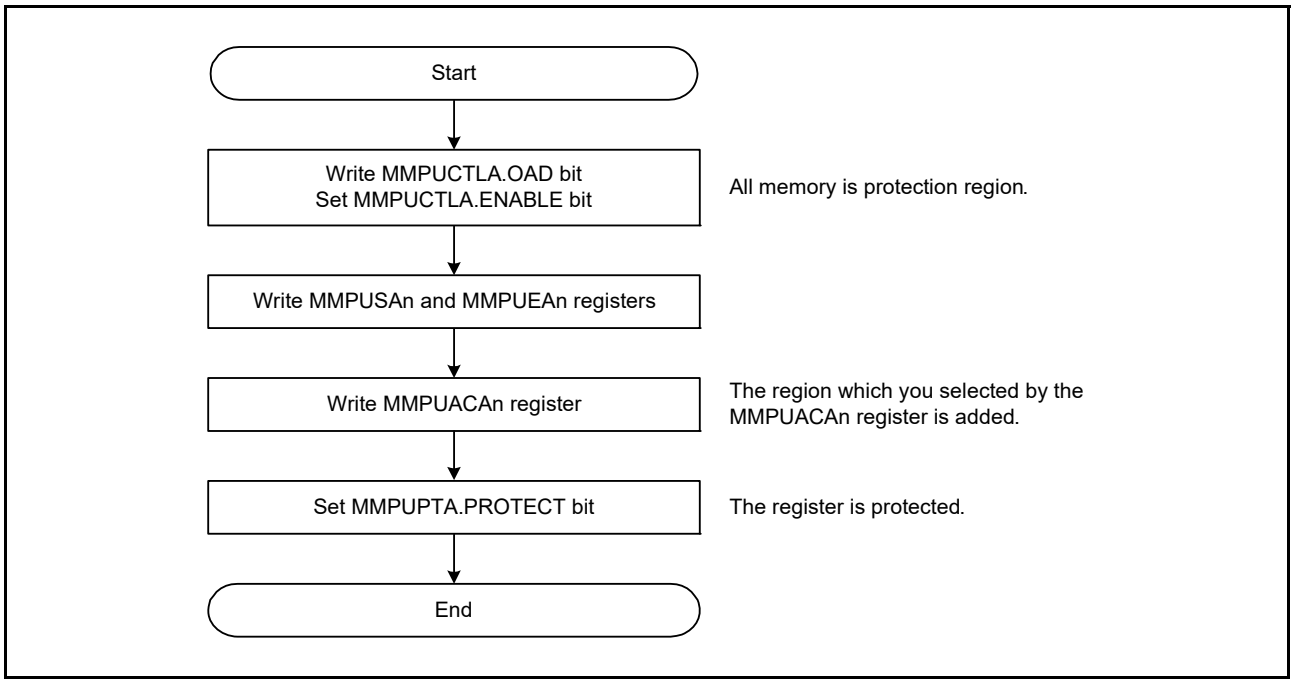


Figure 16.7 Register setting flow after reset

Figure 16.8 shows the register setting flow for region addition. During this register setting, stop the master except the CPU.

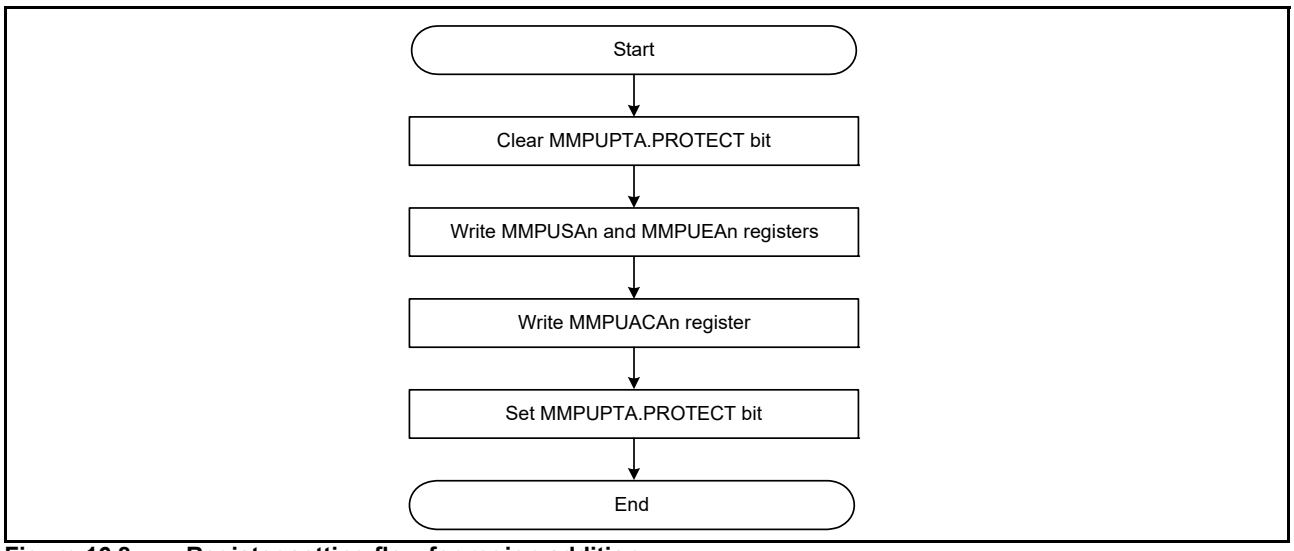


Figure 16.8 Register setting flow for region addition

16.4.2.1 Protection of Registers

Registers related to the bus master MPU can be protected with the PROTECT bit in the MMPUPTA register.

16.4.2.2 Memory Protection Error

If access to the protected region is detected, the bus master MPU generates an error. A memory protection error can select between a non-maskable interrupt or a reset by the OAD bit.

Status of non-maskable interrupt is indicated by ICU.NMISR.BUSMST. For details, see section 14, Interrupt Controller Unit (ICU). Status of reset is indicated by SYSTEM.RSTSR1.BUSMRF. For details, see section 6, Resets.

16.5 Bus Slave MPU

The MCU incorporates a bus slave MPU that checks access to the bus slave function such as flash or SRAM.

The bus slave function can be accessed from two bus master (CPU and bus master MPU group A). The bus slave MPU has a separate protection register for each bus master, and can protect access independently. If access to the protected region is detected, the bus slave MPU generates a reset or a non-maskable interrupt, including hold of bus error status, error access status, and bus error address in the I/O Register. For details bus error address, bus error status, and error access status, see 15.3.9 and 15.3.10 in section 15, Buses. The supported access control information for the individual regions consists of permission to read and to write.

Table 16.7 lists the specifications of the bus slave MPU and Figure 16.9 shows a block diagram of the bus slave MPU.

Table 16.7 Bus slave MPU specifications (1 of 2)

Specifications	Description
Protect bus master	Bus master MPU group A: DMA bus
Protect slave function	Memory bus 3: Code flash memory Memory bus 4: SRAM0 Memory bus 5: SRAM1 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, RTC, WDT, IWDT, IIC, CAN, SSI, ADC14, DAC12, and DOC) Internal peripheral bus 4: Connected to peripheral modules (SCI, IrDA, SPI, CRC, and SDHI) Internal peripheral bus 5: Connected to peripheral modules (KINT, AGT, USBFS, OPAMP, ACMPHS, ACMPLP, and CTSU) Internal peripheral bus 7: Connected to SecureIP (SCE5) Internal peripheral bus 9: Flash memory (in P/E) and data flash memory External bus (CS area): Connected to the external devices External bus (QSPI area): Connected to the external SPI devices

Table 16.7 Bus slave MPU specifications (2 of 2)

Specifications	Description
Access-control information settings for individual regions	Permission to read, permission to write
Operation after detection	Reset, non-maskable interrupt, or exception
Register protection	Register can be protected from illegal writes

The bus slave MPU is located in each bus slave side and controls the permission or protection of access from each bus master to each bus slave.

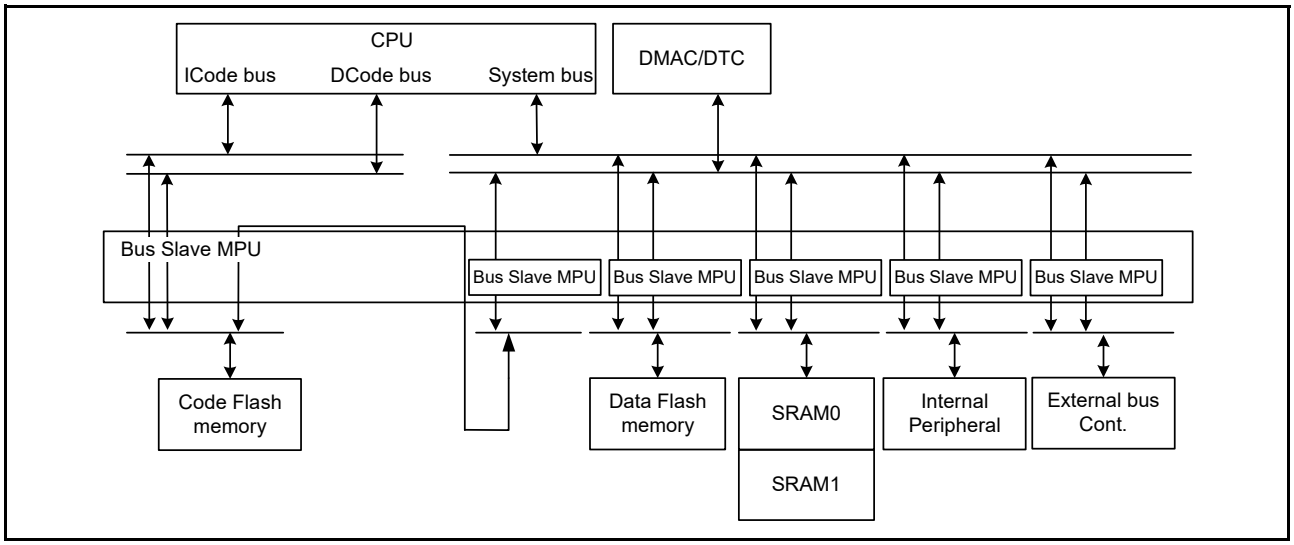


Figure 16.9 Block diagram of the bus slave MPU

16.5.1 Register Descriptions

Note: Bus access must be stopped before writing to MPU registers.

16.5.1.1 Access Control Register for Memory bus 3 (SMPUMBIU)

Address(es): SMPU.SMPUMBIU 4000 0C10h

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

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	RPGRPA	Master Group A Read Protection	0: Master group A read of memory protection disabled 1: Master group A read of memory protection enabled.	R/W
b3	WPGRPA	Master Group A Write Protection	0: Master group A write of memory protection disabled 1: Master group A write of memory protection enabled.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RPGRPA bit (Master Group A Read Protection)

The RPGRPA bit enables or disables memory protection for Master Group A Read Memory Bus 3.

WPGRPA bit (Master Group A Write Protection)

WPGRPA bit enables or disables memory protection for Master Group A Write Memory Bus 3.

16.5.1.2 Access Control Register for Internal peripheral bus 9 (SMPUFBIU)

Address(es): SMPU.SMPUFBIU 4000 0C14h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	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	Bit name	Description	R/W
b0	RPCPU	CPU Read protection	0: CPU read of memory protection disabled 1: CPU read of memory protection enabled.	R/W
b1	WPCPU	CPU Write protection	0: CPU write of memory protection disabled 1: CPU write of memory protection enabled.	R/W
b2	RPGRPA	Master Group A Read protection	0: Master group A read of memory protection disabled 1: Master group A read of memory protection enabled.	R/W
b3	WPGRPA	Master Group A Write protection	0: Master group A write of memory protection disabled 1: Master group A write of memory protection enabled.	R/W
b15 to b4	—	Reserved	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 Read Internal Peripheral Bus 9.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for CPU Write Internal Peripheral Bus 9.

RPGRPA bit (Master Group A Read protection)

The RPGRPA bit enables or disables memory protection for Master Group A Read Internal Peripheral Bus 9.

WPGRPA bit (Master Group A Write protection)

The WPGRPA bit enables or disables memory protection for Master Group A Write Internal Peripheral Bus 9.

16.5.1.3 Access Control Register for Memory bus 4 (SMPUSRAM0)

Address(es): SMPU.SMPUSRAM0 4000 0C18h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	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	Bit name	Description	R/W
b0	RPCPU	CPU Read protection	0: CPU read of memory protection disabled 1: CPU read of memory protection enabled.	R/W
b1	WPCPU	CPU Write protection	0: CPU write of memory protection disabled 1: CPU write of memory protection enabled.	R/W
b2	RPGRPA	Master Group A Read protection	0: Master group A read of memory protection disabled 1: Master group A read of memory protection enabled.	R/W
b3	WPGRPA	Master Group A Write protection	0: Master group A write of memory protection disabled 1: Master group A write of memory protection enabled.	R/W
b15 to b4	—	Reserved	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 Read Memory Bus 4.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for CPU Write Memory Bus 4.

RPGRPA bit (Master Group A Read protection)

The RPGRPA bit enables or disables memory protection for Master Group A Read Memory Bus 4.

WPGRPA bit (Master Group A Write protection)

The WPGRPA bit enables or disables memory protection for Master group A Write Memory Bus 4.

16.5.1.4 Access Control Register for Memory bus 5 (SMPUSRAM1)

Address(es): SMPU.SMPUSRAM1 4000 0C1Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	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	Bit name	Description	R/W
b0	RPCPU	CPU Read protection	0: CPU read of memory protection disabled 1: CPU read of memory protection enabled.	R/W
b1	WPCPU	CPU Write protection	0: CPU write of memory protection disabled 1: CPU write of memory protection enabled.	R/W
b2	RPGRPA	Master Group A Read protection	0: Master group A read of memory protection disabled 1: Master group A read of memory protection enabled.	R/W
b3	WPGRPA	Master Group A Write protection	0: Master group A write of memory protection disabled 1: Master group A write of memory protection enabled.	R/W
b15 to b4	—	Reserved	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 Read Memory Bus 5.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for CPU Write Memory Bus 5.

RPGRPA bit (Master Group A Read protection)

The RPGRPA bit enables or disables memory protection for Master Group A Read Memory Bus 5.

WPGRPA bit (Master Group A Write protection)

The WPGRPA bit enables or disables memory protection for Master Group A Write Memory Bus 5.

16.5.1.5 Access Control Register for Internal peripheral bus 1 (SMPUP0BIU)

Address(es): SMPU.SMPUP0BIU 4000 0C20h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	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	Bit name	Description	R/W
b0	RPCPU	CPU Read protection	0: CPU read of memory protection disabled 1: CPU read of memory protection enabled.	R/W
b1	WPCPU	CPU Write protection	0: CPU write of memory protection disabled 1: CPU write of memory protection enabled.	R/W
b2	RPGRPA	Master Group A Read protection	0: Master group A read of memory protection disabled 1: Master group A read of memory protection enabled.	R/W
b3	WPGRPA	Master Group A Write protection	0: Master group A write of memory protection disabled 1: Master group A write of memory protection enabled.	R/W
b15 to b4	—	Reserved	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 Read Internal Peripheral Bus 1.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for CPU Write Internal Peripheral Bus 1.

RPGRPA bit (Master Group A Read protection)

The RPGRPA bit enables or disables memory protection for Master Group A Read Internal Peripheral Bus 1.

WPGRPA bit (Master Group A Write protection)

The WPGRPA bit enables or disables memory protection for Master Group A Write Internal Peripheral Bus 1.

16.5.1.6 Access Control Register for Internal peripheral bus 3 (SMPUP2BIU)

Address(es): SMPU.SMPUP2BIU 4000 0C24h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	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	Bit name	Description	R/W
b0	RPCPU	CPU Read protection	0: CPU read of memory protection disabled 1: CPU read of memory protection enabled.	R/W
b1	WPCPU	CPU Write protection	0: CPU write of memory protection disabled 1: CPU write of memory protection enabled.	R/W
b2	RPGRPA	Master Group A Read protection	0: Master group A read of memory protection disabled 1: Master group A read of memory protection enabled.	R/W
b3	WPGRPA	Master Group A Write protection	0: Master group A write of memory protection disabled 1: Master group A write of memory protection enabled.	R/W
b15 to b4	—	Reserved	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 read Internal Peripheral Bus 3, Internal Peripheral Bus 4, and Internal Peripheral Bus 5.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for CPU Write Internal Peripheral Bus 3, Internal Peripheral Bus 4, and Internal Peripheral Bus 5.

RPGRPA bit (Master Group A Read protection)

The RPGRPA bit enables or disables memory protection for Master Group A Read Internal Peripheral Bus 3, Internal Peripheral Bus 4, and Internal Peripheral Bus 5.

WPGRPA bit (Master Group A Write protection)

The WPGRPA bit enables or disables memory protection for Master Group A Write Internal Peripheral Bus 3, Internal Peripheral Bus 4, and Internal peripheral Bus 5.

16.5.1.7 Access Control Register for Internal peripheral bus 7 (SMPUP6BIU)

Address(es): SMPU.SMPUP6BIU 4000 0C28h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	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	Bit name	Description	R/W
b0	RPCPU	CPU Read protection	0: CPU read of memory protection disabled 1: CPU read of memory protection enabled.	R/W
b1	WPCPU	CPU Write protection	0: CPU write of memory protection disabled 1: CPU write of memory protection enabled.	R/W
b2	RPGRPA	Master Group A Read protection	0: Master group A read of memory protection disabled 1: Master group A read of memory protection enabled.	R/W
b3	WPGRPA	Master Group A Write protection	0: Master group A write of memory protection disabled 1: Master group A write of memory protection enabled.	R/W
b15 to b4	—	Reserved	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 Read Internal Peripheral Bus 7.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for CPU Write Internal Peripheral Bus 7.

RPGRPA bit (Master Group A Read protection)

The RPGRPA bit enables or disables memory protection for Master Group A Read Internal Peripheral Bus 7.

WPGRPA bit (Master Group A Write protection)

The WPGRPA bit enables or disables memory protection for Master Group A Write Internal Peripheral Bus 7.

16.5.1.8 Access Control Register for CS area (SMPUEXBIU)

Address(es): SMPU.SMPUEXBIU 4000 0C30h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	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	Bit name	Description	R/W
b0	RPCPU	CPU Read protection	0: CPU read of memory protection disabled 1: CPU read of memory protection enabled.	R/W
b1	WPCPU	CPU Write protection	0: CPU write of memory protection disabled 1: CPU write of memory protection enabled.	R/W
b2	RPGRPA	Master Group A Read protection	0: Master group A read of memory protection disabled 1: Master group A read of memory protection enabled.	R/W
b3	WPGRPA	Master Group A Write protection	0: Master group A write of memory protection disabled 1: Master group A write of memory protection enabled.	R/W
b15 to b4	—	Reserved	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 Read CS Area.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for CPU Write CS Area.

RPGRPA bit (Master Group A Read protection)

The RPGRPA bit enables or disables memory protection for Master Group A Read CS Area.

WPGRPA bit (Master Group A Write protection)

The WPGRPA bit enables or disables memory protection for Master Group A Write CS Area.

16.5.1.9 Access Control Register for QSPI area (SMPUEXBIU2)

Address(es): SMPU.SMPUEXBIU2 4000 0C34h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	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	Bit name	Description	R/W
b0	RPCPU	CPU Read protection	0: CPU read of memory protection disabled 1: CPU read of memory protection enabled.	R/W
b1	WPCPU	CPU Write protection	0: CPU write of memory protection disabled 1: CPU write of memory protection enabled.	R/W
b2	RPGRPA	Master Group A Read protection	0: Master group A read of memory protection disabled 1: Master group A read of memory protection enabled.	R/W
b3	WPGRPA	Master Group A Write protection	0: Master group A write of memory protection disabled 1: Master group A write of memory protection enabled.	R/W
b15 to b4	—	Reserved	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 Read QSPI Area.

WPCPU bit (CPU Write protection)

The WPCPU bit enables or disables memory protection for CPU Write QSPI Area.

RPGRPA bit (Master Group A Read protection)

The RPGRPA bit enables or disables memory protection for Master Group A Read QSPI Area.

WPGRPA bit (Master Group A Write protection)

The WPGRPA bit enables or disables memory protection for Master Group A Write QSPI Area.

16.5.1.10 Slave MPU Control Register (SMPUCTL)

Address(es): SMPU.SMPUCTL 4000 0C00h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit name	Description	R/W
b0	OAD	Operation after detection	0: Non-maskable interrupt 1: Reset.	R/W
b1	PROTECT	Protection of register	0: All bus slave register writes are possible 1: All bus slave register writes are protected. Read is possible.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits are used to enable or disable writing of the OAD and PROTECT bit.	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, write A5h in halfword access in KEY[7:0] simultaneously.

PROTECT bit (Protection of register)

The PROTECT bit enables or disables writing to the associated registers to be protected. The following registers are protected by SMPUCTL.PROTECT:

- SMPUMBIU
- SMPUFBIU
- SMPUSRAM0
- SMPUSRAM1
- SMPUP0BIU
- SMPUP2BIU
- SMPUP6BIU
- SMPUEXBIU
- SMPUEXBIU2.

When the PROTECT bit is set, write A5h in halfword access in KEY[7:0] simultaneously.

KEY[7:0] bits (Key Code)

These bits enable or disable writing of the OAD and PROTECT bits.

When writing to the OAD and PROTECT bits, write A5h to KEY[7:0] simultaneously. When values other than A5h are written to the KEY[7:0] bits, the OAD and the PROTECT bits are not updated. The KEY[7:0] bits always read as 00h.

16.5.2 Functions

16.5.2.1 Memory Protection

The bus slave MPU monitoring functions with access control information that is set for the individual access control register 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 register (SMPUMBIU, SMPUFBIU, SMPUSRAM0, SMPUSRAM1, SMPUP0BIU, SMPUP2BIU, SMPUP6BIU, SMPUEXBIU, and SMPUEXBIU2).

16.5.2.2 Protection of Registers

Registers related to the bus slave MPU can be protected with the PROTECT bit in the SMPUCTL register.

16.5.2.3 Memory Protection Error

If access to the protected region is detected, the bus slave MPU generates a memory protection error. A memory protection error can select between a non-maskable interrupt or a reset by the OAD bit.

Status of non-maskable interrupt is indicated by ICU.NMISR.BUSSST. For details, see section 14, Interrupt Controller Unit (ICU). Status of reset is indicated by SYSTEM.RSTSR1.BUSSRF. For details, see section 6, Resets.

16.6 Security MPU

The MCU incorporates a Security MPU and has secure regions. A secure region can be protected from access by a non-secure program. Access to a protected region by a non-secure program is not possible. In addition, the Security MPU provides one guard function that includes the code flash.

Table 16.8 lists the specifications of the Security MPU.

Table 16.8 Specifications of the Security MPU

Specifications	Description
Guard function	Code flash
Region to be covered by memory protection	0000 0000h to 000F FFFFh (code flash memory)
Number of regions	Program Counter = 2 regions, Data Access = 1 region
Specifying addresses of individual regions	Setting the address where regions start and end
Setting to make memory protection effective or ineffective in individual regions	Settings effective or ineffective for the associated region

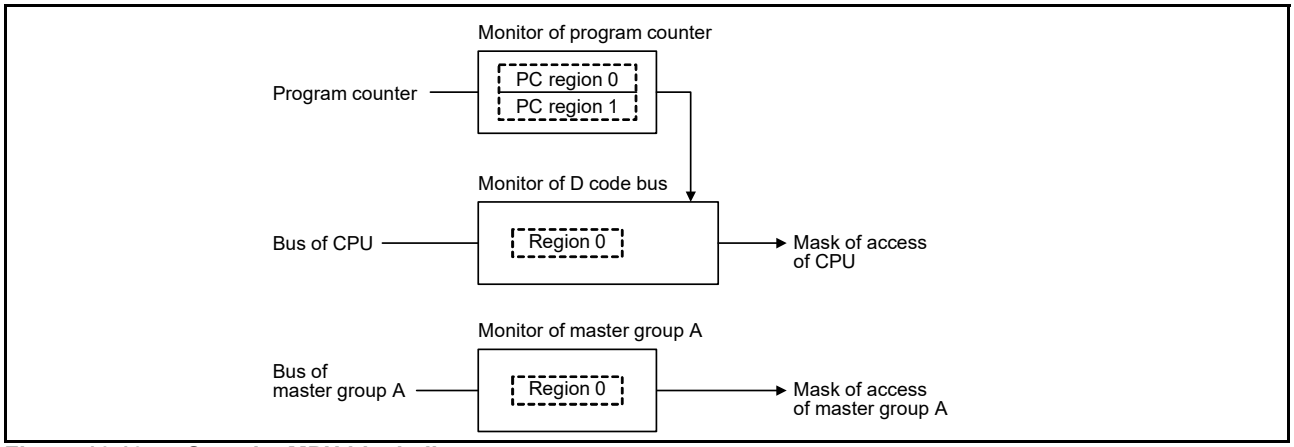


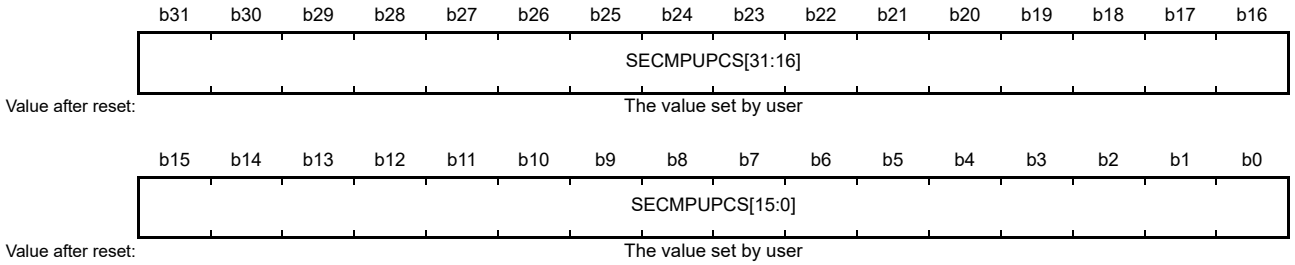
Figure 16.10 Security MPU block diagram

16.6.1 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 flash.

16.6.1.1 Security MPU Program Counter Start Address Register (SECMPUPCSn) (n = 0, 1)

Address(es): SECMPUPCS0 0000 0408h, SECMPUPCS1 0000 0410h



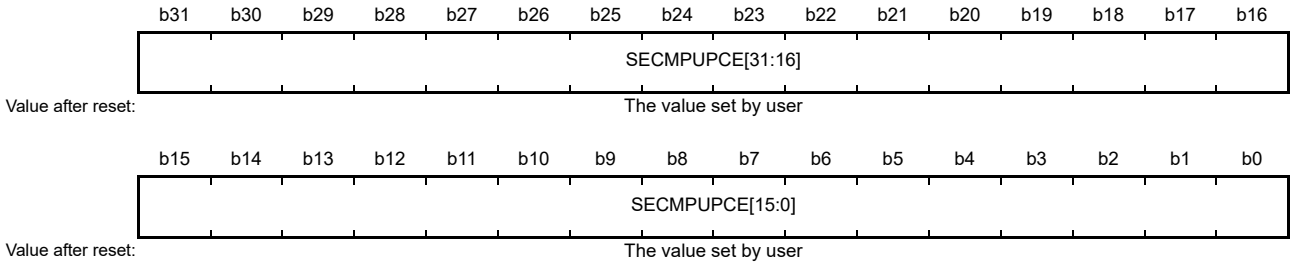
Bit	Symbol	Bit name	Description	R/W
b31 to b0	SECMPUPCS[31:0]	Region Start Address	Address where the region starts, for use in region determination. The lower 2 bits are read as 0. The value range should be 0000 0000h to 000F FFFCh.	R

The SECMPUPCSn and SECMPUPCEn registers specify the security fetch region of the flash (0000 0000h to 000F FFFFh). 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).

Address space of greater than 12 bytes is required between the last instruction of the non-secure program and the first instruction of the secure program.

16.6.1.2 Security MPU Program Counter End Address Register (SECMPUPCEn) (n = 0, 1)

Address(es): SECMPUPCE0 0000 040Ch, SECMPUPCE1 0000 0414h



Bit	Symbol	Bit name	Description	R/W
b31 to b0	SECMPUPCE[31:0]	Region End Address	Address where the region ends, for use in region determination. The lower 2 bits are read as 1. The value range should be 0000 0003h to 000F FFFFh.	R

16.6.1.3 Security MPU Region 0 Start Address Register (SECMPUS0)

Address(es): SECMPUS0 0000 0418h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	SECMPUS0[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	The value set by user							

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SECMPUS0[15:0]															
Value after reset:	The value set by user															

Bit	Symbol	Bit name	Description	R/W
b23 to b0	SECMPUS0[23:0]	Region Start Address	Address where the region starts, for use in region determination. The lower 2 bits are read as 0. The value range should be 0000 0000h to 00FF FFFCh.	R
b31 to b24	—	Reserved	These bits are read as 0.	R

The SECMPUS0 and SECMPUE0 registers specify the security program and the flash data (0000 0000h to 000F FFFFh). 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. Setting of the vector table area is prohibited.

16.6.1.4 Security MPU Region 0 End Address Register (SECMPUE0)

Address(es): SECMPUE0 0000 041Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	SECMPUE0[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	The value set by user							

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SECMPUE0[15:0]															
Value after reset:	The value set by user															

Bit	Symbol	Bit name	Description	R/W
b23 to b0	SECMPUE0[23:0]	Region End Address	Address where the region ends, for use in region determination. The lower 2 bits are read as 1. The value range should be 0000 0003h to 00FF FFFFh.	R
b31 to b24	—	Reserved	These bits are read as 0.	R

16.6.1.5 Security MPU Access Control Register (SECMPUAC)

Address(es): SECMPUAC 0000 0438h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DISPC1	DISPC0	—	—	—	—	—	—	—	DIS0
Value after reset:	1	1	1	1	1	1	The value set by user		1	1	1	1	1	1	1	The value set by user

Bit	Symbol	Bit name	Description	R/W
b0	DIS0	Region 0 Disable	0: Security MPU Region 0 enabled 1: Security MPU Region 0 disabled.	R
b7 to b1	—	Reserved	These bits are read as 1.	R
b8	DISPC0	PC Region 0 Disable	0: Security MPU PC Region 0 enabled 1: Security MPU PC Region 0 disabled.	R
b9	DISPC1	PC Region 1 Disable	0: Security MPU PC Region 1 enabled 1: Security MPU PC Region 1 disabled.	R
b15 to b10	—	Reserved	These bits are read as 1.	R

Note: When flash memory is erased, Security MPU is disabled.
Note: For the setting method of enabling and disabling of the security MPU, see section 16.6.2, Memory Protection.

DIS0 bit (Region 0 Disable)

The DIS0 bit enables or disables Security MPU Region 0. If Security MPU Region 0 is enabled, the code flash region within the limits set up by SECMPUS0 and SECMPUE0 is the security data.

DISPC0 bit (PC Region 0 Disable)

The DISPC0 bit enables or disables Security MPU PC Region 0. If Security MPU PC Region 0 is enabled, th flash region within the limits set up by SECMPUPCS0 and SECMPUPCE0 is the security program.

DISPC1 bit (PC Region 1 Disable)

The DISPC1 bit enables or disables Security MPU PC Region 1. If Security MPU PC Region 1 is enabled, the flash region within the limits set up by SECMPUPCS1 and SECMPUPCE1 is the security program.

16.6.2 Memory Protection

The Security MPU protects the secure memory (flash) from being accessed by programs other than a secure program. If access to the 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) must be set to 0 and DIS0 in the Security MPU Access Control Register (SECMPUAC) must be set to 0. When the security MPU is disabled, all bits in DISPC0, DISPC1, and DIS0 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 (DMAC, DTC).

Secure data can be accessed in the following condition:

- Secure data can be accessed from a secure program.

Note: Secure program:
Flash region within the limits set up by SECMPUPCS0 and SECMPUPCE0
Flash region within the limits set up by SECMPUPCS1 and SECMPUPCE1
Non-secure program: All regions without the secure program
Secure data: Flash region within the limits set up by SECMPUS0 and SECMPUE0

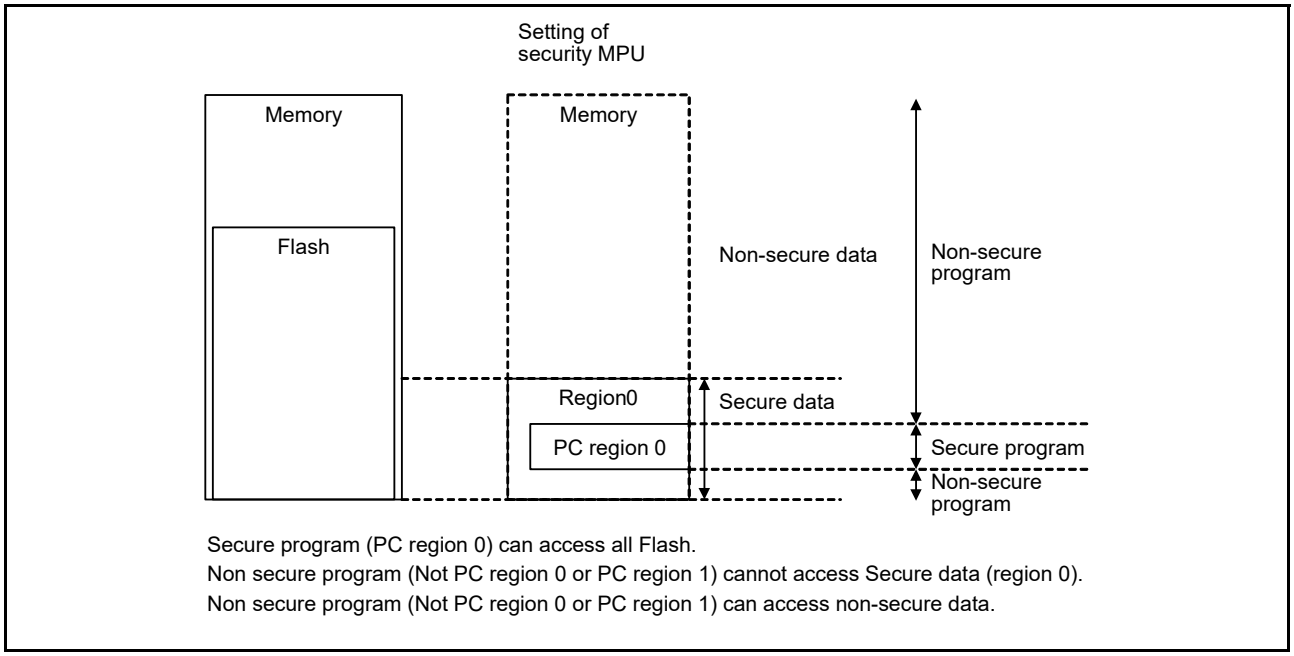


Figure 16.11 Use case of Security MPU

16.6.3 Usage Notes

When using the debugger, access to secure data is prohibited.

16.7 References

1. *ARM®v7-M Architecture Reference Manual* (ARM DDI 0403D)
2. *ARM® Cortex®-M4 Processor Technical Reference Manual* (ARM DDI 0439D)
3. *ARM® Cortex®-M4 Devices Generic User Guide* (ARM DUI 0553A)

17. DMA Controller (DMAC)

17.1 Overview

The 4-channel DMA Controller (DMAC) can transfer data without intervention from the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address. Table 17.1 lists the DMAC specifications and Figure 17.1 shows the block diagram.

Table 17.1 DMAC specifications

Parameter		Description
Number of channels		4 channels (DMACm, where m = 0 to 3)
Transfer space		4 GB (0000 0000h to FFFF FFFFh excluding reserved areas)
Maximum transfer volume		64M data units (maximum number of transfers in block transfer mode: 1,024 data units × 65,536 blocks)
DMA activation source		Selectable for each channel individually to: • Software trigger • Interrupt requests from peripheral modules or trigger from external interrupt input pins.*1
Channel priority		Channel 0 > Channel 1 > Channel 2 > Channel 3 (Channel 0: highest)
Transfer data	Single data	Bit length: 8, 16, 32 bits
	Block size	Number of data: 1 to 1,024
Transfer mode	Normal transfer mode	• One data transfer by one DMA transfer request • Selectable free running mode (total number of data transfers is not specified).
	Repeat transfer mode	• One data transfer by one DMA transfer request • Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination • Maximum settable repeat size: 1,024.
	Block transfer mode	• One data block transfer by one DMA transfer request • Maximum settable block size: 1,024 data
Selective functions	Extended repeat area function	• Allows data to be transferred by repeating the address values in the specified range, with the upper bit values in the transfer address register remaining fixed • Area of 2 bytes to 128 MB individually selectable as the extended repeat area for transfer source and destination.
Interrupt request (DMACm_INT)	Transfer end interrupt	Generated on completion of transferring data volume specified by the transfer counter.
	Transfer escape end interrupt	Generated when: • The repeat size of data transfer is complete • The source address of the extended repeat area overflows • The destination address of the extended repeat area overflows.
Event link activation (DMACm_INT)		An event link request is generated after each data transfer (for block transfer, after each block is transferred)
Module-stop function		Allows a module-stop state to be set

Note 1. For details on DMAC activation sources, see Table 14.3, Interrupt Vector Table in section 14, Interrupt Controller Unit (ICU).

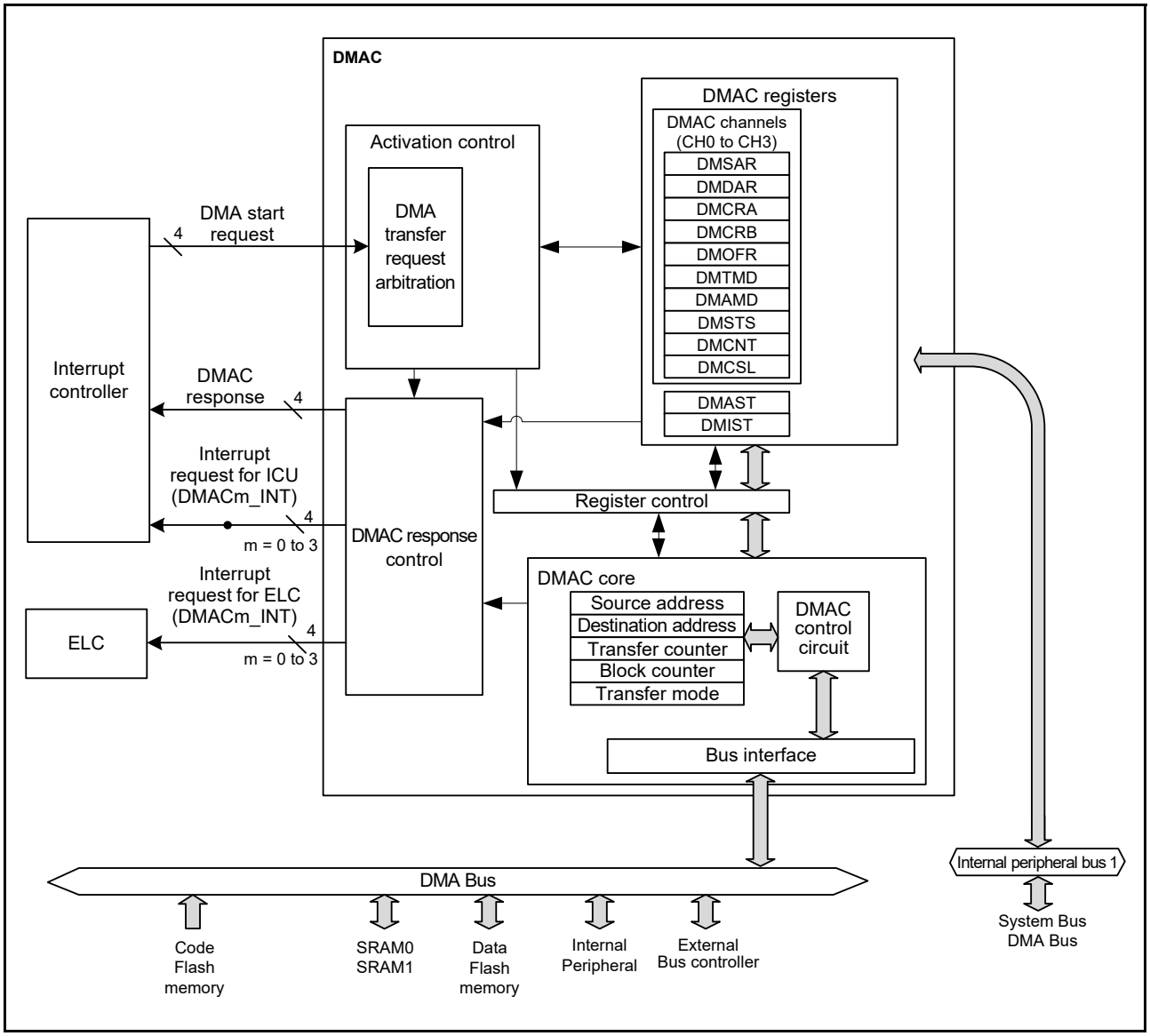


Figure 17.1 DMAC block diagram

17.2 Register Descriptions

17.2.1 DMA Source Address Register (DMSAR)

Address(es): DMAC0.DMSAR 4000 5000h, DMAC1.DMSAR 4000 5040h, DMAC2.DMSAR 4000 5080h, DMAC3.DMSAR 4000 50C0h

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

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

Bit	Description	Setting range	R/W
b31 to b0	Specifies the transfer source start address	0000 0000h to FFFF FFFFh (4 GB)	R/W

Set DMSAR while DMAC activation is disabled (the DMST bit in DMAST = 0) or DMA transfer is disabled (the DTE bit in DMCNT = 0).

Note: Address alignment in this register must match the transfer data size value selected in the SZ bit in DMTMD.

17.2.2 DMA Destination Address Register (DMDAR)

Address(es): DMAC0.DMDAR 4000 5004h, DMAC1.DMDAR 4000 5044h, DMAC2.DMDAR 4000 5084h, DMAC3.DMDAR 4000 50C4h

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

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

Bit	Description	Setting range	R/W
b31 to b0	Specifies the transfer destination start address	0000 0000h to FFFF FFFFh (4 GB)	R/W

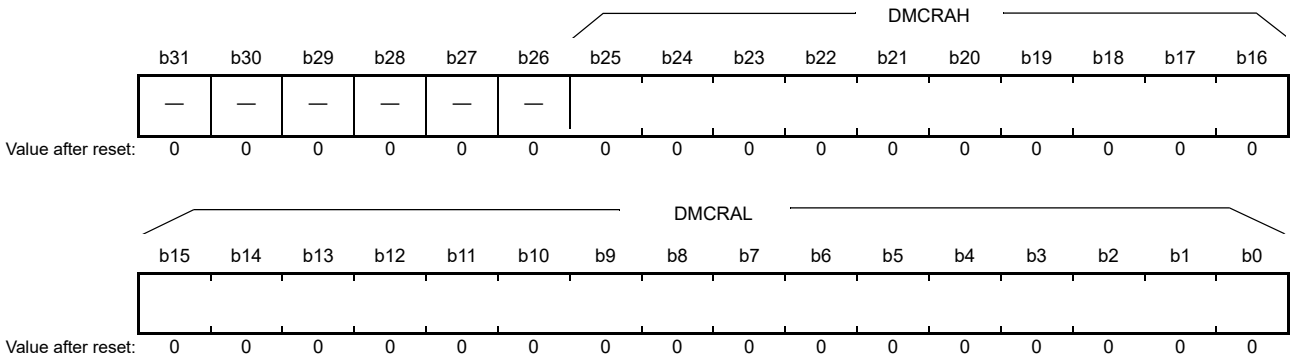
Set DMDAR while DMAC activation is disabled (the DMST bit in DMAST = 0) or DMA transfer is disabled (the DTE bit in DMCNT = 0).

Note: Address alignment in this register must match the transfer data size value selected in the SZ bit in DMTMD.

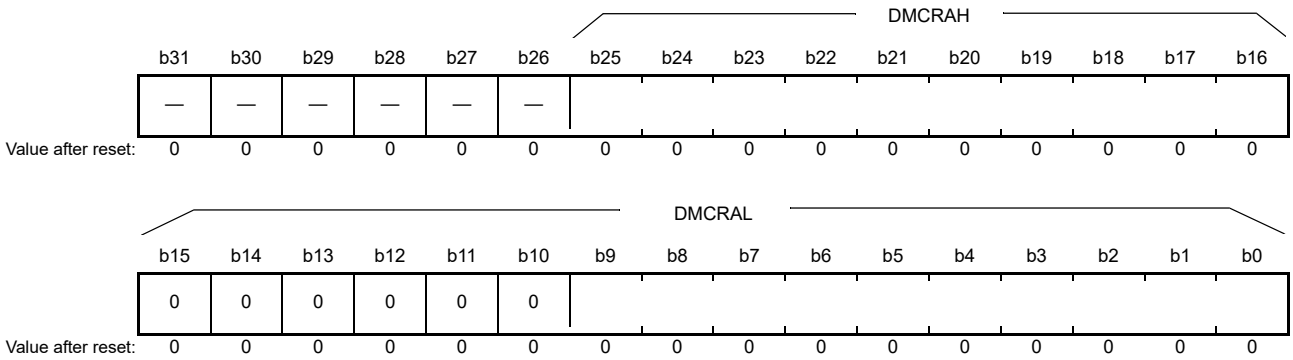
17.2.3 DMA Transfer Count Register (DMCRA)

Address(es): DMAC0.DMCRA 4000 5008h, DMAC1.DMCRA 4000 5048h, DMAC2.DMCRA 4000 5088h, DMAC3.DMCRA 4000 50C8h

- Normal transfer mode



- Repeat transfer mode, block transfer mode



Symbol	Bit name	Description	R/W
DMCRAL	Lower bits of transfer count	Specifies the number of transfer operations	R/W
DMCRAH	Upper bits of transfer count		R/W

Note: In repeat and block transfer modes, set the same value for DMCRAH and DMCRAL.

(1) Normal Transfer Mode (MD[1:0] bits in DMACm.DMTMD = 00b)

In normal transfer mode, DMCRAL functions as a 16-bit transfer counter. The number of transfer operations is one when the setting is 0001h, and 65,535 when it is FFFFh. The value is decremented by one each time data is transferred. A setting of 0000h indicates an unspecified number of transfer operations. Data transfer is performed with the transfer counter stopped, that is, in free running mode. Do not use DMCRAH in normal transfer mode. Write 0000h to DMCRAH.

(2) Repeat Transfer Mode (MD[1:0] bits in DMACm.DMTMD = 01b)

In repeat transfer mode, DMCRAH specifies the repeat size and DMCRAL functions as a 10-bit transfer counter. The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In this mode, a value in the range of 000h to 3FFh (1 to 1024) can be set for DMCRAH and DMCRAL. Setting bits [15:10] in DMCRAL is invalid. Write 0 to these bits. The value in DMCRAL is decremented by one each time data is transferred until it reaches 000h, at which time the value in DMCRAH is loaded into DMCRAL.

(3) Block Transfer Mode (MD[1:0] bits in DMACm.DMTMD = 10b)

In block transfer mode, DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter. The block size is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In this mode, a value in the range of 000h to 3FFh can be set for DMCRAH and DMCRAL.
Setting bits [15:10] in DMCRAL is invalid. Write 0 to these bits. The value in DMCRAL is decremented by one each time data is transferred until it reaches 000h, at which time the value in DMCRAH is loaded into DMCRAL.

17.2.4 DMA Block Transfer Count Register (DMCRB)

Address(es): DMAC0.DMCRB 4000 500Ch, DMAC1.DMCRB 4000 504Ch, DMAC2.DMCRB 4000 508Ch, DMAC3.DMCRB 4000 50CCh

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

Bit	Description	Setting range	R/W
b15 to b0	Specifies the number of block transfer operations or repeat transfer operations	0001h to FFFFh (1 to 65,535) 0000h (65,536).	R/W

DMCRB specifies the number of operations in block and repeat transfer modes. The number of transfer operations is one when the setting is 0001h, 65,535 when it is FFFFh, and 65,536 when it is 0000h. In repeat transfer mode, the value is decremented by one when the final data of one repeat size is transferred. In block transfer mode, the value is decremented by one when the final data of one block size is transferred. Do not use DMCRB in normal transfer mode as the setting is invalid.

17.2.5 DMA Transfer Mode Register (DMTMD)

Address(es): DMAC0.DMTMD 4000 5010h, DMAC1.DMTMD 4000 5050h, DMAC2.DMTMD 4000 5090h, DMAC3.DMTMD 4000 50D0h

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

Bit	Symbol	Bit name	Description	R/W
b1, b0	DCTG[1:0]	Transfer Request Source Select	b1 b0 0 0: Software 0 1: Interrupts*1 from peripheral modules or external interrupt input pins 1 0: Setting prohibited 1 1: Setting prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	SZ[1:0]	Transfer Data Size Select	b9 b8 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: Setting prohibited.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13, b12	DTS[1:0]	Repeat Area Select	b13 b12 0 0: The destination is specified as the repeat area or block area 0 1: The source is specified as the repeat area or block area 1 0: The repeat area or block area is not specified 1 1: Setting prohibited.	R/W
b15, b14	MD[1:0]	Transfer Mode Select	b15 b14 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Setting prohibited.	R/W

Note 1. To select the DMAC activation source, use the DELSRn registers of the ICU. For details on DMAC activation sources, see Table 14.4, Event table in section 14, Interrupt Controller Unit (ICU).

DTS[1:0] bits (Repeat Area Select)

The DTS[1:0] bits select either the source or destination as the repeat area in repeat transfer mode and the block area in block transfer mode. In normal transfer mode, these bit settings are invalid.

17.2.6 DMA Interrupt Setting Register (DMINT)

Address(es): DMAC0.DMINT 4000 5013h, DMAC1.DMINT 4000 5053h, DMAC2.DMINT 4000 5093h, DMAC3.DMINT 4000 50D3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable	0: Disabled 1: Enabled.	R/W
b1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable	0: Disabled 1: Enabled.	R/W
b2	RPTIE	Repeat Size End Interrupt Enable	0: Disabled 1: Enabled.	R/W
b3	ESIE	Transfer Escape End Interrupt Enable	0: Disabled 1: Enabled.	R/W
b4	DTIE	Transfer End Interrupt Enable	0: Disabled 1: Enabled.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0	R/W

DARIE bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow occurs on the destination address when this bit is set to 1, the DTE bit in DMCNT is set to 0. At the same time, the ESIF flag in DMSTS sets to 1 to indicate an interrupt triggered by an extended repeat area overflow on the destination address.

When block transfer mode is used with the extended repeat area function, an interrupt occurs after completion of a 1-block size transfer. When the DTE bit is set to 1 in DMACm.DMCNT of the channel associated with the stopped transfer, the transfer resumes from the state it was in when the transfer stopped.

If the extended repeat area is not specified for the destination address, this bit is ignored.

SARIE bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow occurs on the source address while this bit is set to 1, the DTE bit in DMCNT is set to 0. At the same time, the ESIF flag in DMSTS sets to 1 to indicate an interrupt request triggered by an extended repeat area overflow on the source address.

When block transfer mode is used with the extended repeat area function, an interrupt occurs after completion of a 1-block size transfer. When the DTE bit is set to 1 in DMACm.DMCNT of the channel associated with the stopped transfer, the transfer resumes from the state it was in when the transfer stopped. When the extended repeat area is not specified for the source address, this bit is ignored.

RPTIE bit (Repeat Size End Interrupt Enable)

When this bit is set to 1 in repeat transfer mode, the DTE bit in DMCNT is set to 0 after completion of a 1-repeat size data transfer. At the same time, the ESIF flag in DMSTS sets to 1 indicate that the repeat size end interrupt request occurred. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in DMTMD are 10b (= repeat area or block area is not specified).

When this bit is set to 1 in block transfer mode, the DTE bit in DMCNT is set to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the ESIF flag in DMSTS sets to 1 to indicate that the repeat size end interrupt request occurred. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in DMTMD are 10b (= repeat area or block area is not specified).

ESIE bit (Transfer Escape End Interrupt Enable)

This bit enables the transfer escape end interrupt requests (repeat size end interrupt request and extended repeat area overflow interrupt request) that occur during DMA transfer. The interrupt occurs when this bit is 1 and the ESIF flag in DMSTS is set to 1. The clear the transfer escape end interrupt, clear this bit or the ESIF flag in DMSTS to 0.

DTIE bit (Transfer End Interrupt Enable)

This bit enables the transfer end interrupt request that occurs on completion of a specified number of data transfers. The interrupt occurs when this bit is 1 and the DTIF bit in DMSTS is set to 1. To clear the transfer end interrupt, clear this bit or the DTIF bit in DMSTS to 0.

17.2.7 DMA Address Mode Register (DMAMD)

Address(es): DMAC0.DMAMD 4000 5014h, DMAC1.DMAMD 4000 5054h, DMAC2.DMAMD 4000 5094h, DMAC3.DMAMD 4000 50D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SM[1:0]		—	SARA[4:0]					DM[1:0]		—	DARA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	DARA[4:0]	Destination Address Extended Repeat Area	Specifies the extended repeat area on the destination address. For details on the settings, see Table 17.2.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	DM[1:0]	Destination Address Update Mode	b7 b6 0 0: Destination address is fixed 0 1: Offset addition 1 0: Destination address is incremented 1 1: Destination address is decremented.	R/W
b12 to b8	SARA[4:0]	Source Address Extended Repeat Area	Specifies the extended repeat area on the source address. For details on the settings, see Table 17.2.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15, b14	SM[1:0]	Source Address Update Mode	b15 b14 0 0: Source address is fixed 0 1: Offset addition 1 0: Destination address is incremented 1 1: Destination address is decremented.	R/W

DARA[4:0] bits (Destination Address Extended Repeat Area)

These bits specify the extended repeat area on the destination address. The extended repeat area function is realized through an update of the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 MB. The start address of the extended repeat area is set when the lower address overflows the extended repeat area on an address increment. Similarly, the end address of the extended repeat area is set when the lower address underflows the extended repeat area on an address decrement.

Do not specify the extended repeat area on the destination address when a repeat area or block area is specified as the transfer destination. When repeat or block transfer is selected, and when DMACm.DMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat or block area), write 00000b in the DARA[4:0] bits.

To request an interrupt when an overflow or underflow occurs in the extended repeat area, set the DARIE bit in DMINT to 1. Table 17.2 lists the extended repeat areas associated with each setting.

DM[1:0] bits (Destination Address Update Mode)

These bits select the update mode for the destination address.

- When increment is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the destination address is incremented by 1, 2, and 4, respectively
- When decrement is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the destination address is decremented by 1, 2, and 4, respectively
- When offset addition is selected, the offset specified in the DMACm.DMOFR register is added to the address.

SARA[4:0] bits (Source Address Extended Repeat Area)

These bits specify the extended repeat area on the source address. The extended repeat area function is realized through an update of the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 MB. The start address of the extended repeat area is set when the

lower address overflows the extended repeat area on an address increment. Similarly, the end address of the extended repeat area is set when the lower address underflows the extended repeat area on an address decrement.

Do not specify the extended repeat area on the source address when the repeat or block area is specified as a transfer source. When repeat or block transfer is selected, and when DMACm.DMTMD.DTS[1:0] = 01b (the transfer source is specified as the repeat area or block area), write 00000b in the SARA[4:0] bits.

To request an interrupt when an overflow or underflow occurs in the extended repeat area, set the SARIE bit in DMINT to 1. Table 17.2 lists the extended repeat areas associated with each setting.

SM[1:0] (Source Address Update Mode)

These bits select the update mode for the source address.

- When increment is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the source address is incremented by 1, 2, and 4, respectively
- When decrement is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the source address is decremented by 1, 2, and 4, respectively
- When offset addition is selected, the offset specified in the DMACm.DMOFR register is added to the address.

Table 17.2 SARA[4:0] or DARA[4:0] settings and corresponding repeat areas

SARA4 to SARA0 or DARA4 to DARA0	Extended repeat area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 KB specified as extended repeat area by the lower 10 bits of the address
01011b	2 KB specified as extended repeat area by the lower 11 bits of the address
01100b	4 KB specified as extended repeat area by the lower 12 bits of the address
01101b	8 KB specified as extended repeat area by the lower 13 bits of the address
01110b	16 KB specified as extended repeat area by the lower 14 bits of the address
01111b	32 KB specified as extended repeat area by the lower 15 bits of the address
10000b	64 KB specified as extended repeat area by the lower 16 bits of the address
10001b	128 KB specified as extended repeat area by the lower 17 bits of the address
10010b	256 KB specified as extended repeat area by the lower 18 bits of the address
10011b	512 KB specified as extended repeat area by the lower 19 bits of the address
10100b	1 MB specified as extended repeat area by the lower 20 bits of the address
10101b	2 MB specified as extended repeat area by the lower 21 bits of the address
10110b	4 MB specified as extended repeat area by the lower 22 bits of the address
10111b	8 MB specified as extended repeat area by the lower 23 bits of the address
11000b	16 MB specified as extended repeat area by the lower 24 bits of the address
11001b	32 MB specified as extended repeat area by the lower 25 bits of the address
11010b	64 MB specified as extended repeat area by the lower 26 bits of the address
11011b	128 MB specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited

17.2.8 DMA Offset Register (DMOFR)

Address(es): DMAC0.DMOFR 4000 5018h, DMAC1.DMOFR 4000 5058h, DMAC2.DMOFR 4000 5098h, DMAC3.DMOFR 4000 50D8h

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

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

Bit	Description	Setting range	R/W
b31 to b0	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination	0000 0000h to 00FF FFFFh (0 bytes to (16 MB – 1 byte)) FF00 0000h to FFFF FFFFh (-16 MB to -1 byte)	R/W

Only write to this register while the DMAC operation is stopped or DMA transfer is disabled, not during data transfer. Setting bits [31:25] is invalid. The value in bit [24] is extended to bits [31:25]. Reading DMOFR returns the extended value.

17.2.9 DMA Transfer Enable Register (DMCNT)

Address(es): DMAC0.DMCNT 4000 501Ch, DMAC1.DMCNT 4000 505Ch, DMAC2.DMCNT 4000 509Ch, DMAC3.DMCNT 4000 50DCh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DTE	DMA Transfer Enable	0: Disables DMA transfer 1: Enables DMA transfer.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTE bit (DMA Transfer Enable)

The DTE bit enables DMA transfer. To enable DMA transfer, set the DMST bit in DMAST to 1 to enable DMAC activation, and then set the DTE bit to 1 to enable DMA transfer for the associated channel.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit
- When the specified total volume of data transfer is complete
- When DMA transfer is stopped by a repeat size end interrupt
- When DMA transfer is stopped by an extended repeat area overflow interrupt.

17.2.10 DMA Software Start Register (DMREQ)

Address(es): DMAC0.DMREQ 4000 501Dh, DMAC1.DMREQ 4000 505Dh, DMAC2.DMREQ 4000 509Dh, DMAC3.DMREQ 4000 50DDh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CLRS	—	—	—	SWREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SWREQ	DMA Software Start	0: DMA transfer is not requested 1: DMA transfer is requested.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CLRS	DMA Software Start Bit Auto Clear Select	0: SWREQ bit is cleared after DMA transfer is started by software 1: SWREQ bit is not cleared after DMA transfer is started by software.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SWREQ bit (DMA Software Start)

Writing 1 to this bit generates a DMA transfer request. After DMA transfer starts in response, it is set to 0 if the CLRS bit is set to 0. It does not clear if CLRS bit is 1. The DMA transfer request can be re-issued after the transfer is complete.

Note: Setting this bit is valid and DMA transfer by software is enabled only when the DCTG[1:0] bits in DMTMD are set to 00b, specifying software as the DMA activation source. Setting this bit is invalid when the DCTG[1:0] bits in DMTMD are set to any value other than 00b.

To start DMA transfer by software with the CLRS bit set to 0, ensure that the SWREQ bit is 0, and then write 1 to the SWREQ bit.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When a DMA transfer request by software is accepted and DMA transfer is started with the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software)
- When 0 is written to this bit.

CLRS bit (DMA Software Start Bit Auto Clear Select)

When an SWREQ setting of 1 triggers a transfer request, this bit specifies whether to clear the SWREQ bit to 0 after DMA transfer starts in response. When set to 0, SWREQ is set to 0 after DMA transfer starts. When set to 1, SWREQ does not clear to 0. The DMA transfer request can be re-issued after the transfer is complete.

17.2.11 DMA Status Register (DMSTS)

Address(es): DMAC0.DMSTS 4000 501Eh, DMAC1.DMSTS 4000 505Eh, DMAC2.DMSTS 4000 509Eh, DMAC3.DMSTS 4000 50DEh

	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	DTIF	—	—	—	ESIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ESIF	Transfer Escape End Interrupt Flag	0: No interrupt 1: Interrupt occurred.	R/W*1
b3 to b1	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

Bit	Symbol	Bit name	Description	R/W
b4	DTIF	Transfer End Interrupt Flag	0: No interrupt 1: Interrupt occurred.	R/W*1
b6, b5	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	ACT	DMA Active Flag	0: DMAC operation suspended 1: DMAC operating.	R

Note 1. Only 0 can be written, to clear the flag.

ESIF flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt occurred.

[Setting conditions]

- In repeat transfer mode, when one repeat size data transfer completes with the RPTIE bit in DMINT set to 1
- In block transfer mode, when one block data transfer completes with the RPTIE bit in DMINT set to 1
- When an extended repeat area overflow on the source address occurs with the SARIE bit in DMINT is set to 1, and the SARA[4:0] bits in DMAMD set to any value other than 00000b (extended repeat area is specified on the transfer source address)
- When an extended repeat area overflow on the destination address occurs with the DARIE bit in DMINT set to 1 and the DARA[4:0] bits in DMAMD set to any value other than 00000b (extended repeat area is specified on the transfer destination address).

[Clearing conditions]

- When 0 is written to this bit
- When 1 is written to the DTE bit in DMCNT.

DTIF flag (Transfer End Interrupt Flag)

This flag indicates that the transfer end interrupt occurred.

[Setting conditions]

- In normal transfer mode, when the specified number of unit transfers completes (the value of DMCRAL becomes 0 on completion of transfer)
- In repeat transfer mode, when the specified number of repeat transfer operations completes (the value of DMCRB becomes 0 on completion of transfer)
- In block transfer mode, when the specified number of blocks is transferred (the value of DMCRB becomes 0 on completion of transfer).

[Clearing conditions]

- When 0 is written to this bit
- When 1 is written to the DTE bit in DMCNT.

ACT flag (DMA Active Flag)

This flag indicates whether the DMAC is in the idle or active state.

[Setting condition]

- When the DMAC starts a data transfer.

[Clearing condition]

- When the data transfer in response to one transfer request completes.

17.2.12 DMACA Module Activation Register (DMAST)

Address(es): DMA.DMAST 4000 5200h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DMST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DMST	DMAC Operation Enable	0: Disabled 1: Enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DMST bit (DMAC Operation Enable)

Setting this bit to 1 enables DMAC activation for all channels. When this bit is set to 1 (DMAC activation is enabled), and 1 is written to the DMACm.DMCNT.DTE bit (DMA transfer is enabled) for multiple channels, all of the associated channels can be placed in the transfer request ready state at the same time.

When the DMST bit is set to 0 during DMA transfer, DMA transfer is suspended after the current data transfer corresponding to a single transfer request is complete. To resume DMA transfer, set the DMST bit to 1 again.

[Setting condition]

- When 1 is written to this bit.

[Clearing condition]

- When 0 is written to this bit.

17.3 Operation

17.3.1 Transfer Mode

(1) Normal Transfer Mode

In normal transfer mode, one data unit is transferred for one transfer request. You can specify the number of transfer operations, up to a maximum of 65,535, in DMACm.DMCRAL. When these bits are set to 0000h, no number of operations is specified and data transfer is performed with the transfer counter stopped (free running mode). Except in free running mode, a transfer end interrupt request can be generated after completion of the specified number of transfer operations. Setting DMACm.DMCRB is invalid in normal transfer mode.

Table 17.3 summarizes the register update operation in normal transfer mode.

Table 17.3 Register update operation in normal transfer mode

Register	Function	Update operation after completion of a transfer for one transfer request
DMACm.DMSAR	Transfer source address	Increment, decrement, fixed, or offset addition
DMACm.DMDAR	Transfer destination address	Increment, decrement, fixed, or offset addition
DMACm.DMCRAL	Transfer count	Decremented by one or not updated (in free running mode)
DMACm.DMCRAH	-	Not updated (not used in normal transfer mode)
DMACm.DMCRB	-	Not updated (not used in normal transfer mode)

Figure 17.2 shows the operation in normal transfer mode.

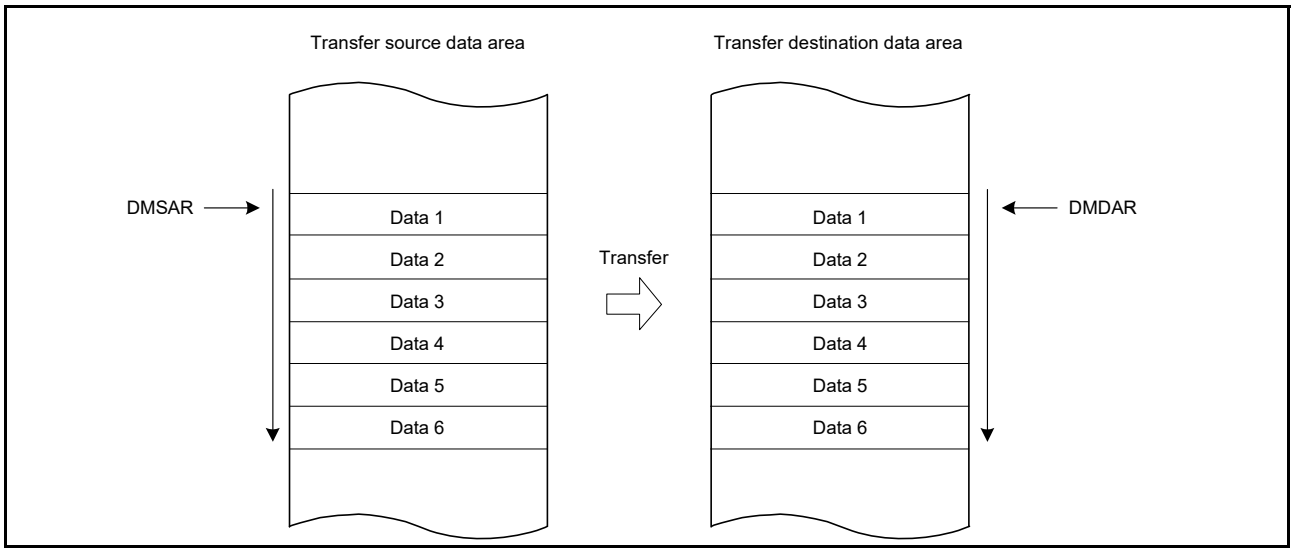


Figure 17.2 Operation in normal transfer mode

(2) Repeat Transfer Mode

In repeat transfer mode, one data unit is transferred for one transfer request. You can set the total data transfer size up to a maximum of 64M data units (1K data units × 64K repeat transfers). To do this, set a maximum of 1K data units as the total repeat transfer size in DMACm.DMCRA, and set a maximum of 64K as the number of repeat transfers in DMACm.DMCRB.

You can specify either the transfer source or destination as a repeat area. When transfer of the repeat size data is complete, the address of the specified repeat area (DMSAR or DMDAR in DMACm) returns to the transfer start address. In this mode, when all data of the specified repeat size is transferred, you can stop DMA transfer and request a repeat size end interrupt. To resume DMA transfer, write 1 to the DTE bit in DMACm.DMCNT during repeat size end interrupt handling.

A transfer end interrupt request can be generated after completion of the specified number of repeat transfers.

Table 17.4 summarizes the register update operation in repeat transfer mode, and Figure 17.3 shows the operation in repeat transfer mode.

Table 17.4 Register update operation in repeat transfer mode

		Update operation after completion of a transfer for one transfer request	
Register	**Function**	**When DMACm.DMCRAL is not 1**	**When DMACm.DMCRAL is 1 (transfer of the last repeat size data unit)**
DMACm.DMSAR	Transfer source address	Increment/decrement/fixed/offset addition	• DMACm.DMTMD.DTS[1:0] = 00b Increment,decrement,fixed, or offset addition • DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR • DMACm.DMTMD.DTS[1:0] = 10b Increment, decrement, fixed, or offset addition
DMACm.DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition	• DMACm.DMTMD.DTS[1:0] = 00b Initial value of DMACm.DMDAR • DMACm.DMTMD.DTS[1:0] = 01b Increment, decrement, fixed, offset addition • DMACm.DMTMD.DTS[1:0] = 10b Increment, decrement, fixed, or offset addition
DMACm.DMCRAH	Repeat size	Not updated	Not updated
DMACm.DMCRAL	Transfer count	Decremented by one	DMACm.DMCRAH
DMACm.DMCRB	Count of repeat transfer operations	Not updated	Decremented by one

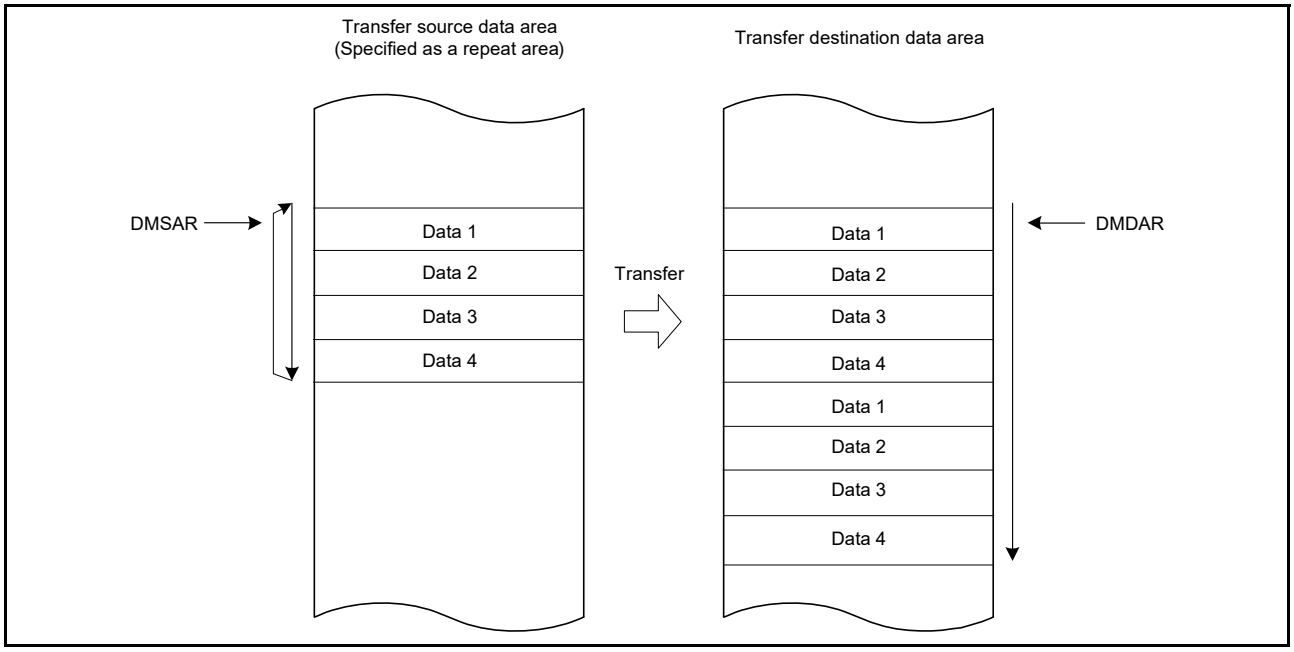


Figure 17.3 Operation in repeat transfer mode

(3) Block Transfer Mode

In block transfer mode, a single data block is transferred for one transfer request. You can set the total data transfer size up to a maximum of 64M data units (1K data units × 64K block transfers). To do this, set a maximum of 1K data units as the total block transfer size in DMACm.DMCRA, and set a maximum of 64K as the number of block transfers in DMACm.DMCRB.

You can specify either the transfer source or destination as a block area. When transfer of a single data block is complete, the address of the specified block area (DMSAR or DMDAR in DMACm) returns to the transfer start address. In this mode, when all data in a single block is transferred, you can stop DMA transfer and request a repeat size end interrupt. To resume DMA transfer, write 1 to the DTE bit in DMACm.DMCNT during repeat size end interrupt handling.

A transfer end interrupt request can be generated after completion of the specified number of block transfers.Table 17.5 summarizes the register update operation in block transfer mode, and Figure 17.4 shows the operation in block transfer mode.

Table 17.5 Register update operation in block transfer mode

Register	Function	Update operation after completion of single-block transfer for one transfer request
DMACm.DMSAR	Transfer source address	• DMACm.DMTMD.DTS[1:0] = 00b Increment, decrement, fixed, or offset addition • DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR • DMACm.DMTMD.DTS[1:0] = 10b Increment, decrement, fixed, or offset addition.
DMACm.DMDAR	Transfer destination address	• DMACm.DMTMD.DTS[1:0] = 00b Initial value of DMACm.DMDAR • DMACm.DMTMD.DTS[1:0] = 01b Increment, decrement,fixed, or offset addition • DMACm.DMTMD.DTS[1:0] = 10b Increment, decrement,fixed, or offset addition.
DMACm.DMCRAH	Block size	Not updated
DMACm.DMCRAL	Transfer count	DMACm.DMCRAH
DMACm.DMCRB	Count of block transfer operations	Decremented by one

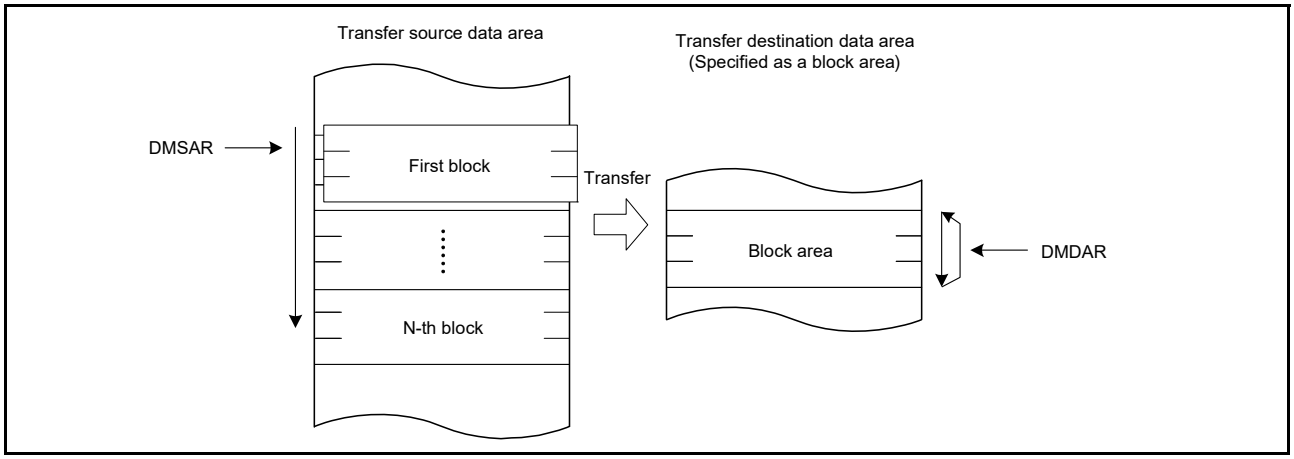


Figure 17.4 Operation in block transfer mode

17.3.2 Extended Repeat Area Function

The DMAC supports extended repeat areas on the transfer source and destination addresses, specified separately in the transfer source address register (DMSAR) and transfer destination address register (DMDAR) of DMACm. When this function is set, the address registers repeatedly indicate the addresses of the specified extended repeat areas. The extended repeat area on the source address is specified by the SARA[4:0] bits in DMACm.DMAMD.

The extended repeat area on the destination address is specified by the DARA[4:0] bits in DMACm.DMAMD. You can specify different sizes for the source and destination. However, you must not specify a transfer source or destination that is set as the repeat or block area as the extended repeat area.

When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an extended repeat area overflow interrupt can be requested. When an overflow occurs in the extended repeat area on the transfer source while the SARIE bit in DMACm.DMINT is set to 1, the ESIF flag in DMACm.DMSTS is set to 1 and the DTE bit in DMACm.DMCNT is set to 0 to stop DMA transfer. At this point, if the ESIE bit in DMACm.DMINT is set to 1, an interrupt by an extended repeat area overflow is requested. When the DARIE bit in DMINT of DMACm is set to 1, the destination address register becomes a target for the function. To resume DMA transfer, write 1 to the DTE bit in DMACm.DMCNT during interrupt handling.

Figure 17.5 shows an example of the extended repeat area operation.

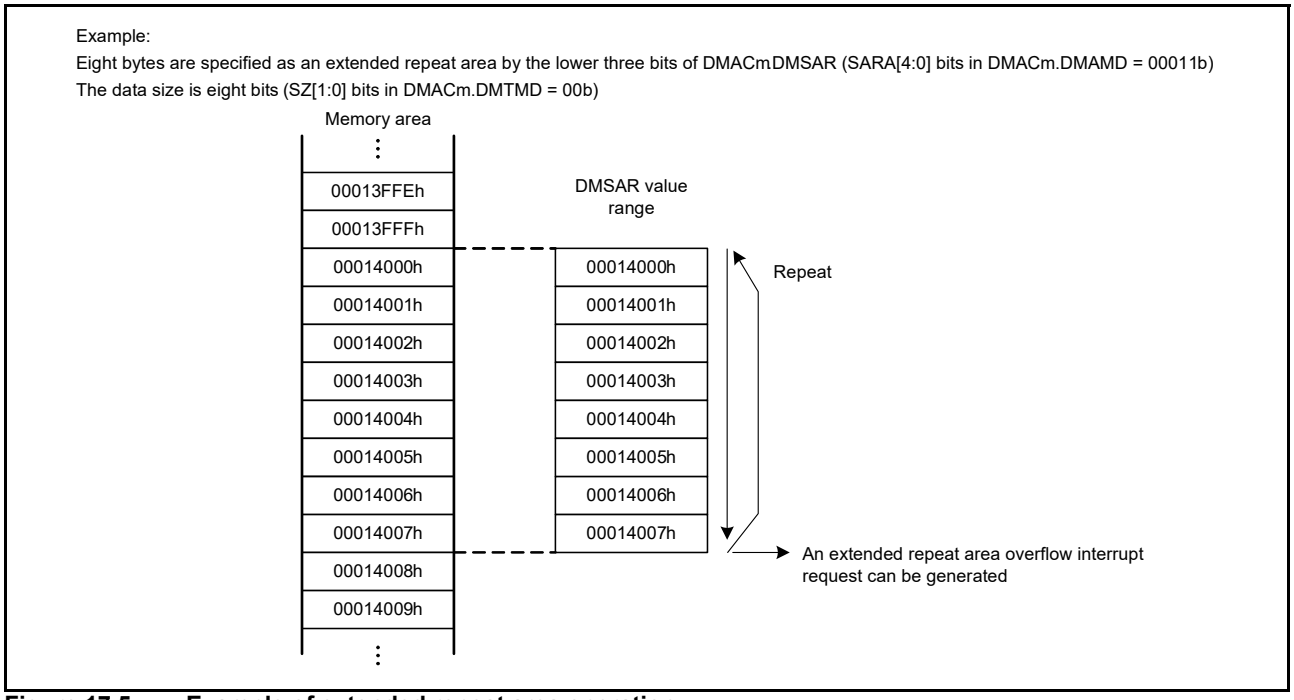


Figure 17.5 Example of extended repeat area operation

When an interrupt by an extended repeat area overflow is used in block transfer mode, the following point should be taken into consideration:

- When a transfer is stopped by an extended repeat area overflow interrupt, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the overflow interrupt is suspended until transfer of the block is complete, and the transfer overruns.

Figure 17.6 shows an example of using the extended repeat area function in block transfer mode.

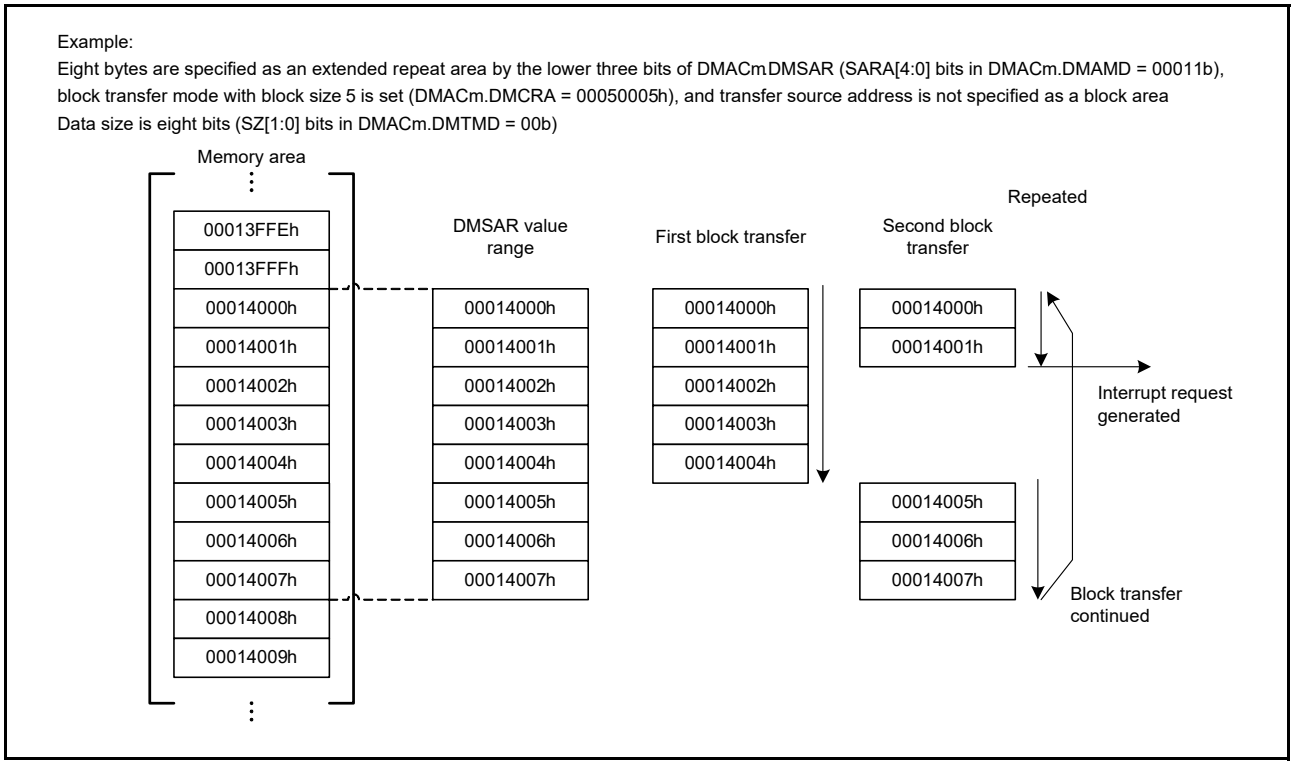


Figure 17.6 Example of extended repeat area function in block transfer mode

17.3.3 Address Update Function Using Offset

The source and destination addresses can be updated by fixing, incrementing, decrementing, or adding an offset. When offset addition is selected, the offset specified in the DMA Offset Register (DMACm.DMOFR) is added to the address every time the DMAC performs one data transfer. This function performs a data transfer when addresses are allocated to separated areas. You can also subtract an offset by setting a negative value in DMACm.DMOFR. The negative value must be in 2's complement.

Table 17.6 shows the address update method in each address update mode.

Table 17.6 Address update method in each address update mode

Address update mode	Settings of DMACm.DMAMD.SM[1:0] and DMACm.DMAMD.DM[1:0] for address update modes	Address update method (for different SZ[1:0] Settings in DMACm.DMTMD)		
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b
Address fixed	00b	Fixed		
Offset addition	01b	+DMACm.DMOFR*1		
Increment	10b	+1	+2	+4
Decrement	11b	-1	-2	-4

Note 1. When setting a negative value in the DMA Offset Register, the value must be 2's complement, obtained by the following formula:
2's complement of a negative offset value = ~ (offset) + 1 (~: bit inversion)

(1) Basic Transfer Using Offset Addition

Figure 17.7 shows an example of address updating using offset addition.

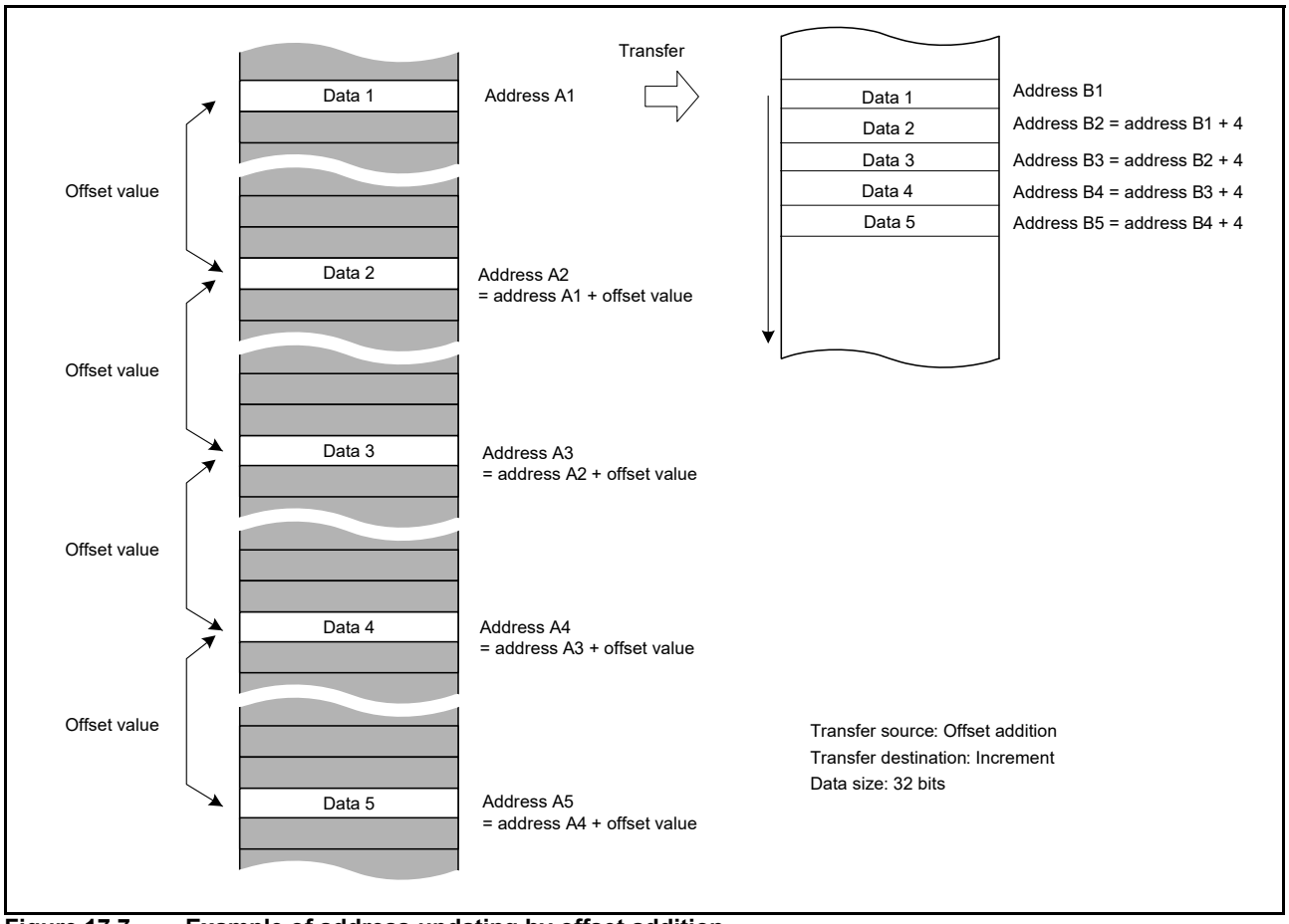


Figure 17.7 Example of address updating by offset addition

In Figure 17.7:

- The transfer data is 32 bits long
- Offset addition is set as the transfer source address update mode
- Increment is set as the transfer destination address update mode.

The second and subsequent data units are each read from the source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to continuous locations on the destination.

(2) Example of XY Conversion Using Offset Addition

Figure 17.8 shows the XY conversion using offset addition in repeat transfer mode. The settings are as follows:

- DMAC0.DMAMD — Transfer source address update mode: offset addition
- DMAC0.DMAMD — Transfer destination address update mode: destination address is incremented
- DMAC0.DMTMD — Transfer data size select: 32 bits
- DMAC0.DMTMD — Transfer mode select: repeat transfer
- DMAC0.DMTMD — Repeat area select: the source is specified as the repeat area
- DMAC0.DMOFR — Offset address: 10h
- DMAC0.DMCRA — Repeat size: 4h
- DMAC0.DMINT — The repeat size end interrupt is enabled.

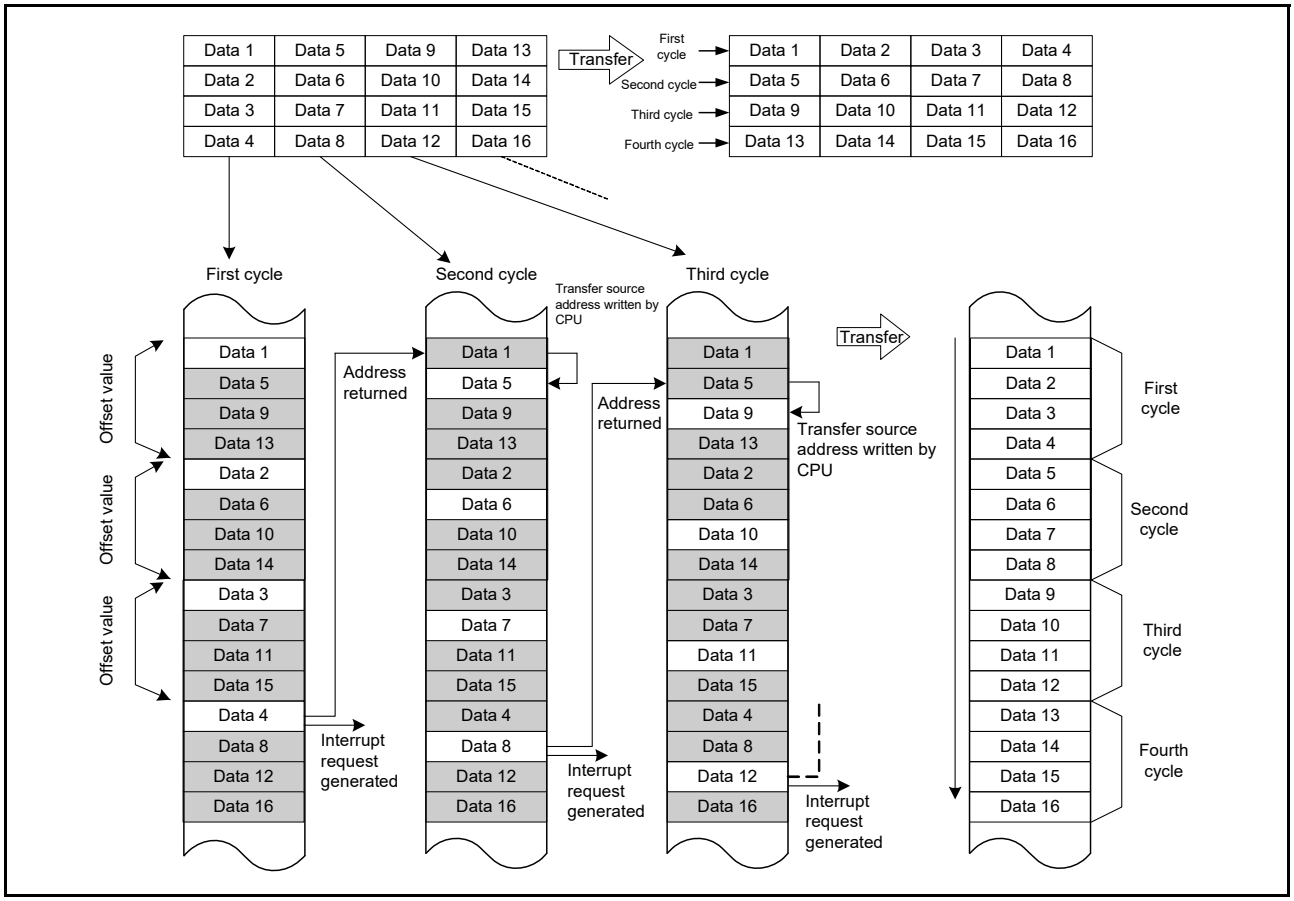


Figure 17.8 XY conversion operation using offset addition in repeat transfer mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to continuous destination addresses. When data 4 is transferred:

- The repeat size of the transfers is complete
- The transfer source address returns to the transfer start address (the address of data 1 on the transfer source)
- A repeat size end interrupt is requested.

During the time this interrupt pauses the transfer, perform the following:

- DMAC0.DMSAR — Rewrite the DMA transfer source address to the address of data 5 (in this example, the data 1 address + 4)
- DMAC0.DMCNT — Set the DTE bit to 1.

The DMA transfer resumes from the state when the DMA transfer was stopped. The same operations are repeated until the transfer source data is transposed to the destination area (XY conversion).

Figure 17.9 shows a flowchart of the XY conversion.

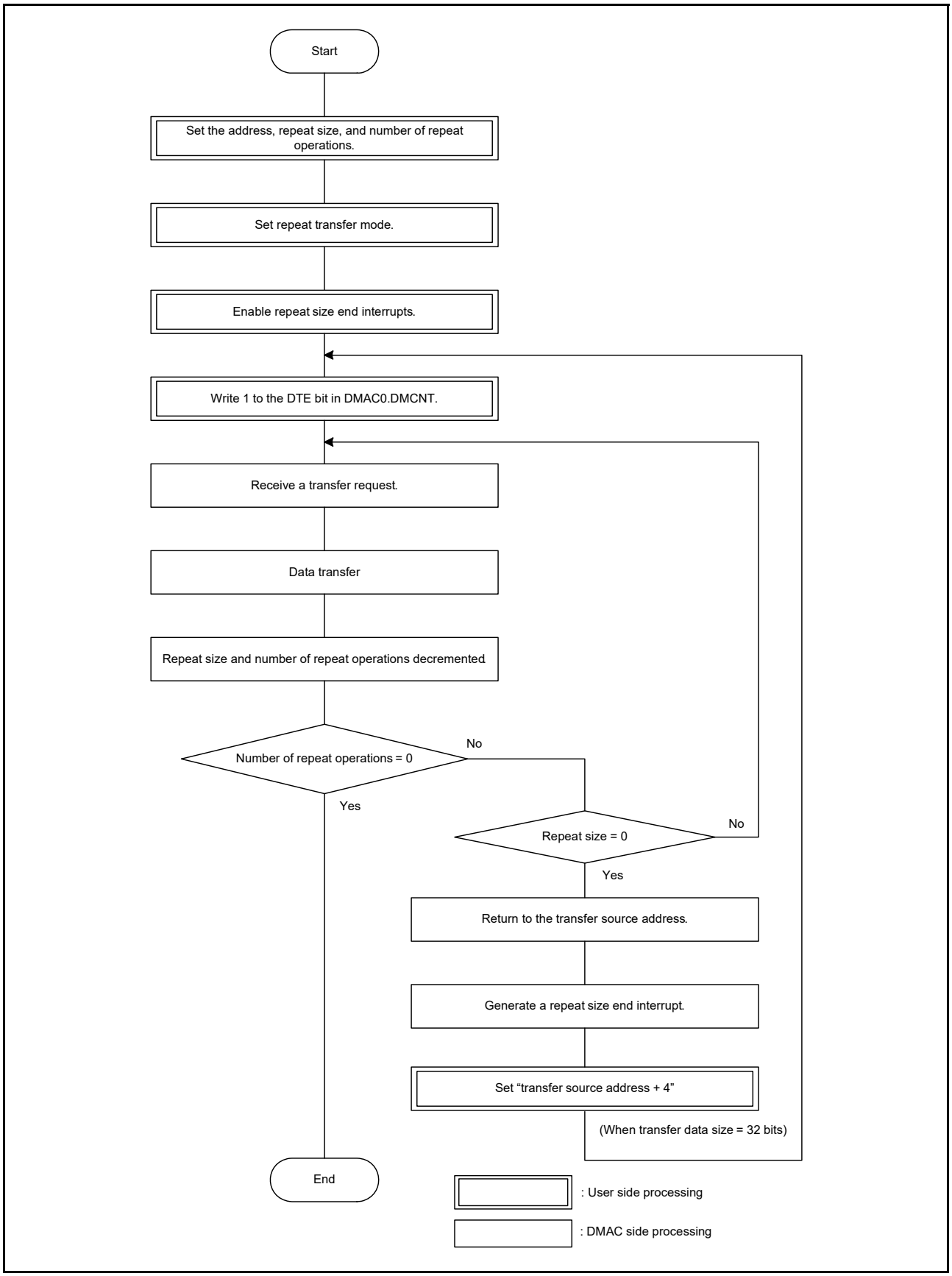


Figure 17.9 XY conversion flowchart using offset addition in repeat transfer mode

17.3.4 Activation Sources

Software, the interrupt requests from the peripheral modules, and external interrupt requests can all be specified as DMAC activation sources. Set the DCTG[1:0] bits in DMACm.DMTMD to select the activation source.

(1) DMAC Activation by Software

To start DMA transfer through software:

1. Set the DCTG[1:0] bits in DMACm.DMTMD to 00b.
2. Set the DTE bit in DMACm.DMCNT to 1 (enable DMA transfer).
3. Set the DMST bit in DMAST to 1 (enable DMAC activation).
4. Set the SWREQ bit in DMACm.DMREQ to 1 (request DMA).

When the DMAC is activated by software while the CLRS bit in DMACm.DMREQ is 0, the SWREQ bit in DMACm.DMREQ is set to 0 after data transfer starts in response to a DMA transfer request. When the DMAC is activated by software while the CLRS bit is 1, SWREQ does not clear to 0 after data transfer starts. A DMA transfer request is issued again after completion of a transfer.

(2) DMAC Activation through Interrupt Requests from On-Chip Peripheral Modules or External Interrupt Requests

You can specify interrupt requests from on-chip peripheral modules and external interrupt requests as DMAC activation sources. The activation source can be selected individually for each channel in ICU.DELSRn.DELS[8:0] (n = 0 to 3).

To start DMAC transfer through an interrupt request from an on-chip peripheral module or an external interrupt request:

1. Set the DCTG[1:0] bits in DMACm.DMTMD to 01b (select interrupts from the peripheral modules and the external interrupt pins).
2. Set the DTE bit in DMACm.DMCNT to 1 (enable DMA transfer).
3. Set ICU.DELSRn.DSEL to the event number (select the DMAC Event Link).
4. Set the DMST bit in DMAST to 1 (enable DMAC activation).

For interrupt requests specified as DMAC activation sources, see Table 14.3, Interrupt Vector Table in section 14, Interrupt Controller Unit (ICU).

17.3.5 Operation Timing

The following timing charts show the number of execution cycles of the minimum.

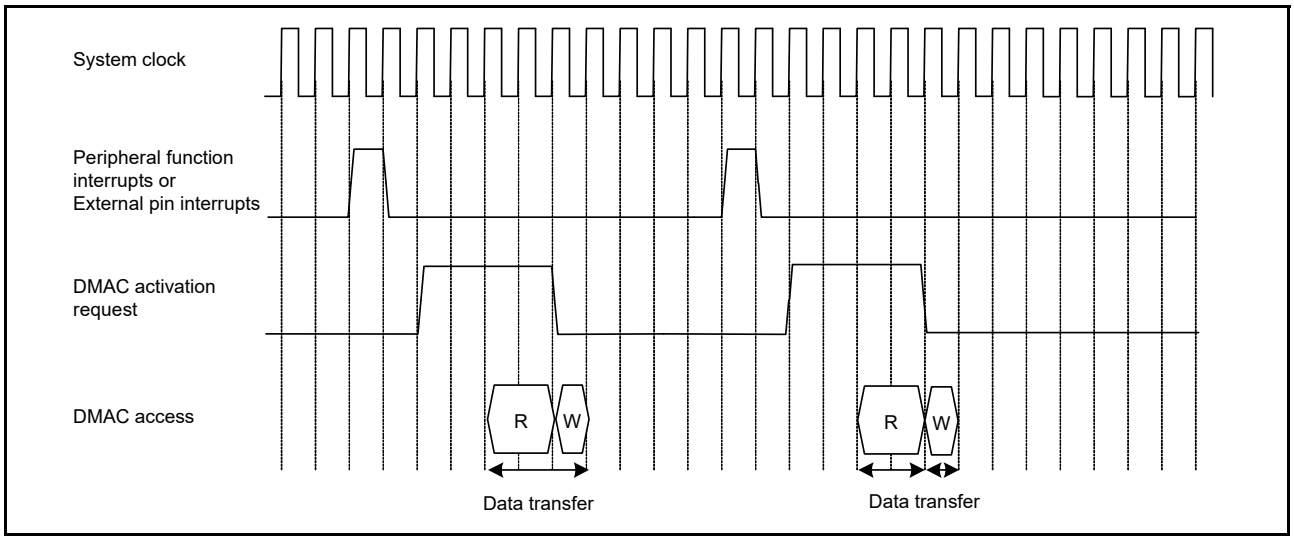


Figure 17.10 DMAC operation timing example (1) (DMA activation by interrupt from peripheral module/external interrupt input pin, normal transfer mode, repeat transfer mode)

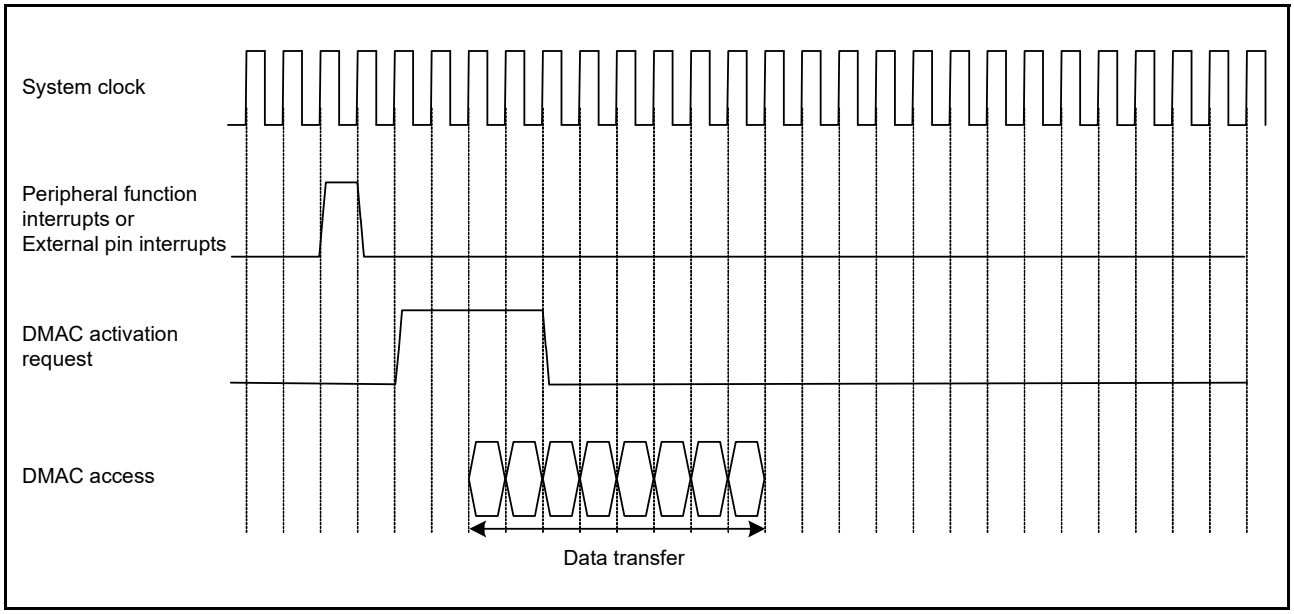


Figure 17.11 DMAC operation timing example (2) (DMA activation by interrupt from peripheral module/external interrupt input pin, block transfer mode, block size = 4)

17.3.6 Execution Cycles of DMAC

Table 17.7 lists the execution cycles in one DMAC data transfer operation.

Table 17.7 DMAC execution cycles

Transfer Mode	Data Transfer (Read)	Data Transfer (Write)
Normal	Cr+1	Cw
Repeat	Cr+1	Cw
Block*1	P × Cr	P × Cw

Note 1. This is the case when the block size is 2 or more. When the block size is 1, normal transfer cycle applies.
Note 2. P = Block size (DMCRAH register setting).
Note 3. Cr = Read destination access cycle.
Note 4. Cw = Data write destination access cycle.

Cr and Cw depend on the access destination. For the number of cycles for each access destination, see section 47, SRAM, section 48, Flash Memory, and section 15, Buses. The frequency ratio of the system clock and the peripheral clock is also taken into consideration.

The unit for +1 in the Data Transfer (Read) column is one system clock cycle, ICLK. For the operation example, see section 17.3.5, Operation Timing.

The DMAC response time is the time from when the DMAC activation source is detected until the DMAC transfer starts. Table 17.7 does not include the time until the DMAC data transfer starts after the DMAC activation source becomes active.

17.3.7 Activating DMAC

Figure 17.12 shows the register setting procedure.

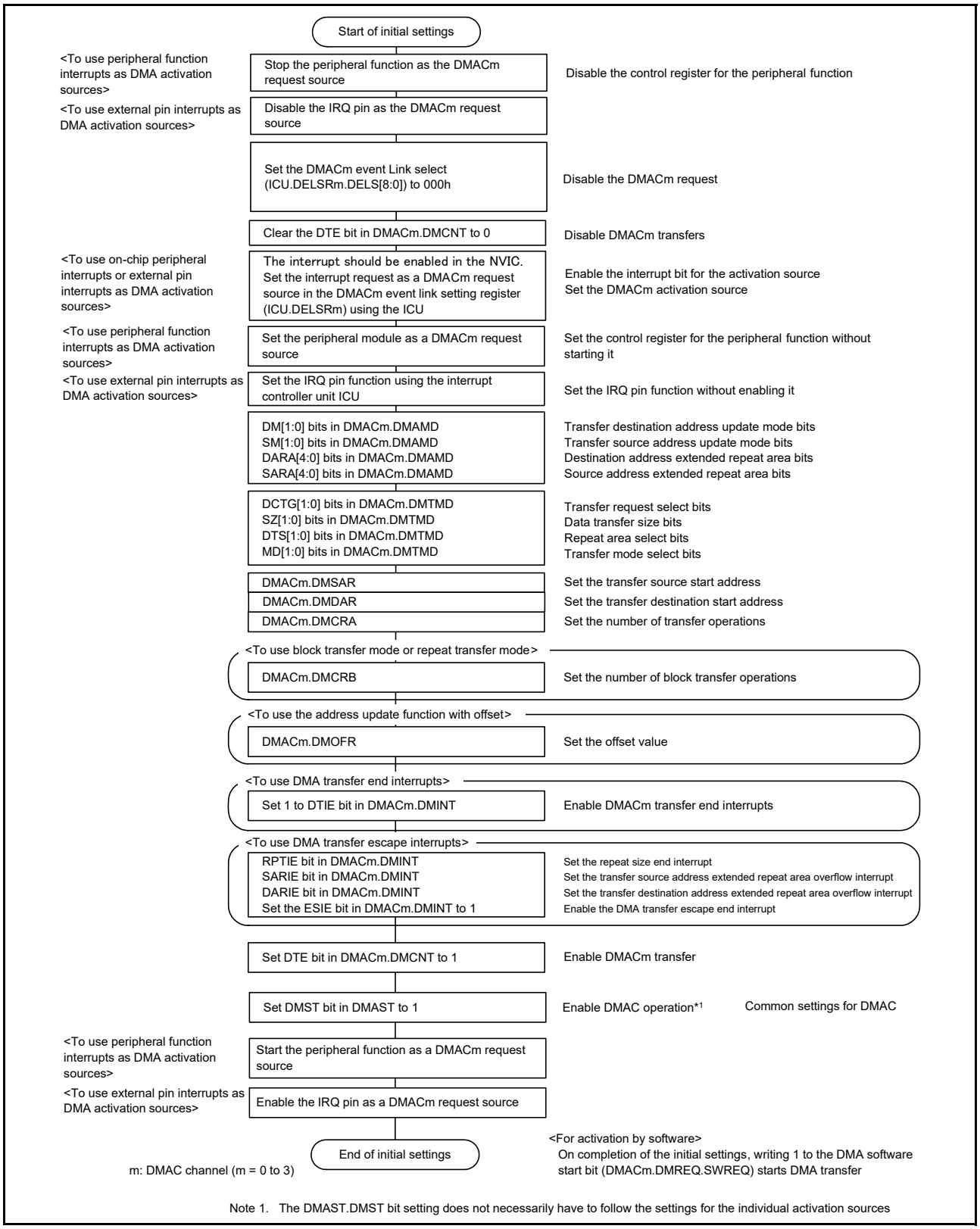


Figure 17.12 Register setting procedure

17.3.8 Starting DMA Transfer

To enable a DMA transfer of channel m, set the DTE bit in DMACm.DMCNT to 1 (DMA transfer enabled) and set the DMST bit in DMAST to 1 (DMAC start enabled). New activation requests are not accepted during the transfer of another DMAC channel or DTC. When the proceeding transfer is complete, channel arbitration selects the DMA transfer request of the highest priority channel, and DMA transfer of that channel starts. When DMA transfer starts, the ACT bit in DMACm.DMSTS is set to 1 (the DMAC is in the active state).

17.3.9 Registers during DMA Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated changes according to the other settings and the transfer state. The registers to be updated are DMSAR, DMDAR, DMCRA, DMCRB, DMCNT, and DMACm.DMSTS, described in the following sections. For details on register update operation in each transfer mode, see Table 17.3 to Table 17.5.

(1) DMA Source Address Register (DMACm.DMSAR)

After the data for one transfer request is transferred, the contents of DMSAR are updated to the address to be accessed by the next transfer request.

(2) DMA Destination Address Register (DMACm.DMDAR)

After the data for one transfer request is transferred, the contents of DMDAR are updated to the address to be accessed by the next transfer request.

(3) DMA Transfer Count Register (DMACm.DMCRA)

After the data for one transfer request is transferred, the count value is updated. The update operation depends on the transfer mode selected.

(4) DMA Block Transfer Count Register (DMACm.DMCRB)

After the data for one transfer request is transferred, the count value is updated. The update operation depends on the transfer mode selected.

(5) DMA Transfer Enable Bit (DMACm.DMCNT.DTE)

The DMACm.DMCNT.DTE bit enables or disables data transfer through register write access. It is automatically set to 0 by the DMAC based on the DMA transfer state.

The conditions for clearing this bit by the DMAC are as follows:

- When the specified total volume of data transfer is complete
- When DMA transfer is stopped by a repeat size end interrupt
- When DMA transfer is stopped by an extended repeat area overflow interrupt.

Writing to the registers for channels whose associated DMACm.DMCNT.DTE bit is set to 1 is prohibited except for DMACm.DMCNT. Writes are only possible after the bit is set to 0.

(6) DMA Active Flag (DMACm.DMSTS.ACT)

The ACT bit in DMSTS of DMACm indicates whether the DMACm is in the idle or active state. This flag is set to 1 when the DMAC starts data transfer, and is set to 0 when data transfer for one transfer request is complete. Even when DMA transfer is stopped by write of 0 to the DTE bit in DMACm.DMCNT, this flag remains 1 until DMA transfer is complete.

(7) Transfer End Interrupt Flag (DMACm.DMSTS.DTIF)

The DTIF flag in DMACm.DMSTS is set to 1 after DMA transfer of the total transfer size is complete. When both this flag and the DTIE bit in DMACm.DMINT are 1, a transfer end interrupt is requested. This flag is set to 1 when the DMA transfer bus cycle is complete and the ACT flag in DMACm.DMSTS is set to 0, indicating the DMA transfer end. The flag automatically is set to 0 when the DTE bit in DMACm.DMCNT is set to 1 during interrupt handling.

(8) Transfer Escape End Interrupt Flag (DMACm.DMSTS.ESIF)

The ESIF flag in DMACm.DMSTS is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this bit and the ESIE bit in DMACm.DMINT are 1, a transfer escape end interrupt is requested. This flag is set to 1 when the bus cycle of the DMA transfer that caused the interrupt request is complete and the ACT flag in DMACm.DMSTS is set to 0, indicating the DMA transfer end. The flag automatically is set to 0 when the DTE bit in DMACm.DMCNT is set to 1 during interrupt handling.

You must set the interrupt control register before sending an interrupt request from the DMAC to the CPU or the DTC. For more information, see section 14, Interrupt Controller Unit (ICU).

17.3.10 Channel Priority

When multiple DMA transfer requests occur, the DMAC determines the priority of channels that have DMA transfer requests.

The priority is fixed as channel 0 > channel 1 > channel 2 > channel 3 (channel 0 is the highest).

When a DMA transfer request occurs during data transfer, channel arbitration starts after the final data unit is transferred, and DMA transfer of the highest-priority channel starts.

17.4 Ending DMA Transfer

The operation for ending a DMA transfer depends on the transfer end conditions. When a DMA transfer ends, the DTE bit in DMCNT and the ACT flag in DMACm.DMSTS change from 1 to 0.

17.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In Normal Transfer Mode (DMACm.DMTMD.MD[1:0] = 00b)

When the value of DMACm.DMCRAL changes from 1 to 0, DMA transfer ends on the associated channel, the DTE bit in DMACm.DMCNT is set to 0, and the DTIF bit in DMACm.DMSTS is set to 1. If the DTIE bit in DMACm.DMINT is 1 at this time, a transfer end interrupt request is sent to the CPU or the DTC.

(2) In Repeat Transfer Mode (DMACm.DMTMD.MD[1:0] = 01b)

When the value of DMACm.DMCRB changes from 1 to 0, DMA transfer ends on the associated channel, the DTE bit in DMACm.DMCNT is set to 0, and the DTIF bit in DMACm.DMSTS is set to 1. If the DTIE bit in DMACm.DMINT is 1 at this time, an interrupt request is sent to the CPU or the DTC.

(3) In Block Transfer Mode (DMACm.DMTMD.MD[1:0] = 10b)

When the value of DMACm.DMCRB changes from 1 to 0, DMA transfer ends on the associated channel, the DTE bit in DMACm.DMCNT is set to 0, and the DTIF bit in DMACm.DMSTS is set to 1. If the DTIE bit in DMACm.DMINT is 1 at this time, an interrupt request is sent to the CPU or the DTC.

You must set the interrupt control register before sending an interrupt request from the DMAC to the CPU or the DTC. For more information, see section 14, Interrupt Controller Unit (ICU).

17.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, if the RPTIE bit in DMACm.DMINT is 1, a repeat size end interrupt is requested when transfer of a single repeat size of data is complete. The DTE bit in DMACm.DMCNT is set to 0 and the ESIF flag in DMACm.DMSTS is set to 1. If the ESIE bit in DMACm.DMINT is 1 at this time, an interrupt request is sent to the CPU or the DTC. To resume the transfer, write 1 to the DTE bit in DMACm.DMCNT.

A repeat size end interrupt can also be requested in block transfer mode. When transfer of a single block size of data is complete, the interrupt is requested in the same way as in repeat transfer mode.

You must set the interrupt control register before sending an interrupt request from the DMAC to the CPU or the DTC. For more information, see section 14, Interrupt Controller Unit (ICU).

17.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the SARIE or DARIE bit in DMACm.DMINT is 1, an extended repeat area overflow interrupt is requested. The DMA transfer is terminated, the DTE bit in DMACm.DMCNT is set to 0, and the ESIF flag in DMACm.DMSTS is set to 1. If the ESIE bit in DMACm.DMINT is 1 at this time, an interrupt request is sent to the CPU or the DTC.

If this interrupt is requested during a read cycle, the subsequent write cycle is performed. In block transfer mode, if the interrupt is requested during a one-block transfer, the remaining data in the block is transferred before transfer stops.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For more information, see section 14, Interrupt Controller Unit (ICU).

17.4.4 Precautions in the end of DMA transfer

A DMA activation request source might occur in the next request after a DMA transfer completes. If this happens, the DMA transfer starts and the DMA activation request is held in DMAC. To prevent this, stop the DMA activation requests by clearing the DELSRn.DSELS[8:0] bits in the ICU to 0.

When a DMA activation request occurs after the last round of the DMA transfer is generated, clear the DMA activation request with a DMA dummy transfer. See Figure 17.13.

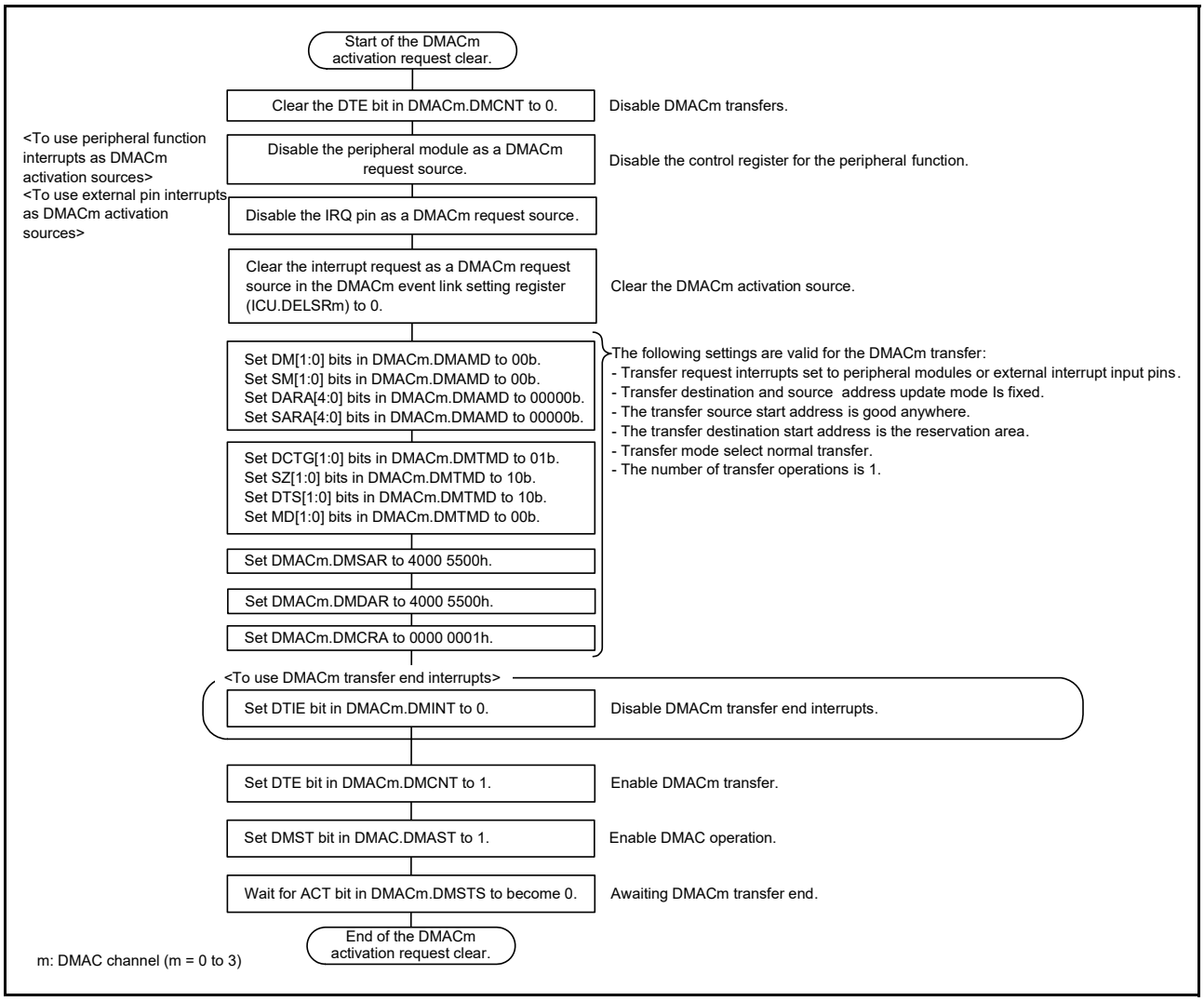


Figure 17.13 Example of register setting procedure to clear the DMA activation interrupt

17.5 Interrupts

Each DMAC channel can output an interrupt request (DMACm_INT) to the CPU or DTC after transfer for one request is complete. When the transfer destination is the external bus, an interrupt request is generated after completion of a data write to the write buffer, and not to the actual transfer destination.

Table 17.8 lists the interrupt sources and their associated status flags and enable bits. Figure 17.14 shows the schematic logic diagram of the interrupt outputs (DMAC0 to DMAC3). Figure 17.15 shows the DMAC interrupt handling routine for resuming and terminating DMA transfers.

Table 17.8 Association between interrupt sources, interrupt status flags, and interrupt enable bits

Interrupt sources		Interrupt enable bits	Interrupt status flags	Request output enable bits
Transfer end		-	DMACm.DMSTS.DTIF	DMACm.DMINT.DTIE
Escape transfer end	Repeat size end	DMACm.DMINT.RPTIE	DMACm.DMSTS.ESIF	DMACm.DMINT.ESIE
	Source address extended repeat area overflow	DMACm.DMINT.SARIE		
	Destination address extended repeat area overflow	DMACm.DMINT.DARIE		

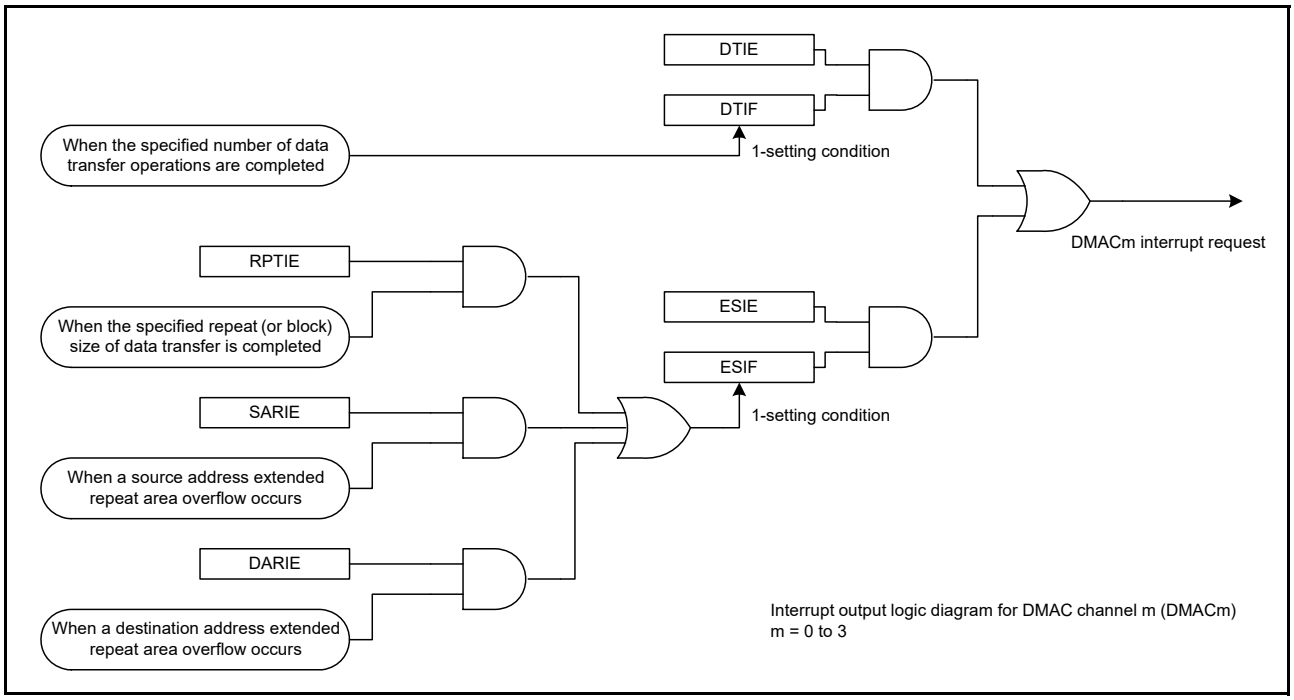


Figure 17.14 Schematic logic diagram of interrupt outputs (DMAC0 to DMAC3)

Different procedures are used for canceling an interrupt to restart a DMA transfer in the following cases:

- When Terminating a DMA Transfer
- When Continuing a DMA Transfer.

(1) When Terminating a DMA Transfer

Write 0 to the DTIF bit in DMACm.DMSTS to clear a transfer end interrupt, and to the ESIF bit in DMACm.DMSTS to clear a repeat size interrupt or an extended repeat area overflow interrupt. DMACm remains in the stopped state. When starting another DMA transfer, set the appropriate registers and set the DTE bit in DMACm.DMCNT to 1 (DMA transfer enabled).

(2) When Continuing a DMA Transfer

Write 1 to the DTE bit in DMACm.DMCNT. The ESIF bit in DMSTS of DMACm is automatically set to 0 (interrupt source cleared), and the DMA transfer resumes.

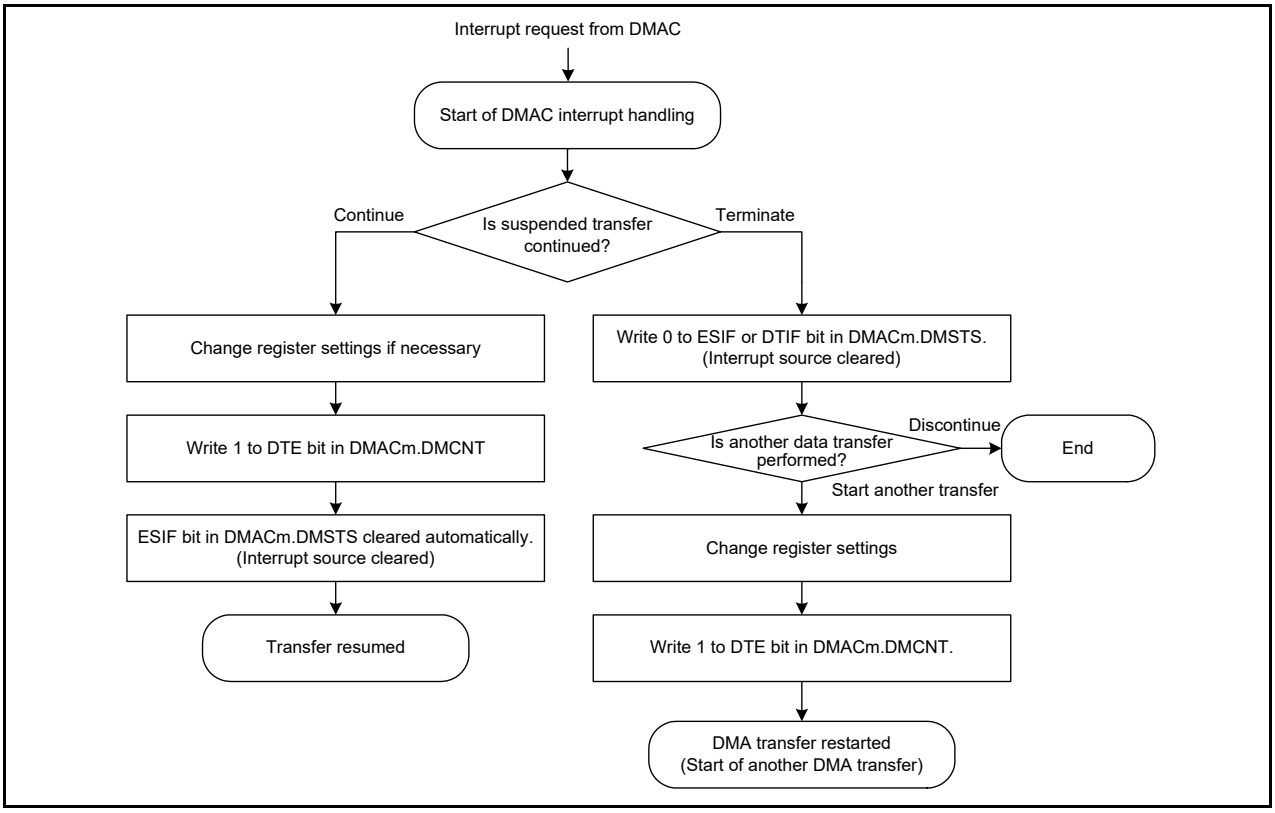


Figure 17.15 DMAC interrupt handling routine to resume/terminate a DMA transfer

17.6 Event Link

Each DMAC channel outputs an event link request signal (DMACm_INT) every time it completes a data transfer, or a block transfer in block transfer mode. When the transfer destination is the external bus, the signal is generated when writing to the write buffer is accepted. For more information, see section 19, Event Link Controller (ELC).

17.7 Low Power Consumption Function

Before entering the module-stop state or Software Standby mode, you must first clear the DMST bit in DMAST to 0 (DMAC suspended), and use the following settings.

(1) Module-Stop Function

Writing 1 to the MSTPA22 bit (transition to the module-stop state) in MSTPCRA enables the module-stop function of the DMAC. If a DMA transfer is in progress at the time 1 is written to the MSTPA22 bit, the transition to the module-stop state continues after DMA transfer ends. Access to the DMAC registers is prohibited while the MSTPA22 bit is 1. Writing 0 to the MSTPA22 bit releases the DMAC from the module-stop state.

(2) Software Standby mode

Use the settings described in section 11.7.1, Transition to Software Standby Mode.

If DMA transfer operations are in progress when the WFI instruction is executed, the DMA transfer completes before the transition to Software Standby mode.

(3) Note on Low Power Consumption Function

For information on the WFI instruction and register settings, see section 11.9.6, Timing of WFI Instruction.

To perform DMA transfer after returning from low power consumption mode, set the DMST bit in DMAST 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 DMAC startup request, specify the CPU as the interrupt request destination, as described in section 14.4.2, Selecting Interrupt Request Destinations, and then execute the WFI instruction.

17.8 Usage Notes

17.8.1 DMA Transfer to External Devices

In a DMA transfer to an external device, the ACT bit in DMACm.DMSTS must be set to 0 (DMAC transfer suspended) from the beginning of the final data write to the end of the external bus access.

17.8.2 Access to Registers during DMA Transfer

Do not write to the following registers of DMACm while the ACT bit in DMSTS of the associated channel is set to 1 (DMAC active state) or the DTE bit in DMCNT of the associated channel is set to 1 (DMA transfer enabled):

- DMSAR
- DMDAR
- DMCRA
- DMCRB
- DMTMD
- DMINT
- DMAMD
- DMOFR.

17.8.3 DMA Transfer to Reserved Areas

DMA transfer to reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on reserved areas, see section 4, Address Space.

17.8.4 Setting of DMAC Event Link Setting Register of the Interrupt Controller Unit (ICU.DELSRn)

Before setting the DMAC event link setting register (ICU.DELSRn), make sure the DMA transfer enable bit (DMACm.DMCNT.DTE) is set to 0, disabling DMA transfer. Additionally, ensure that the DTC activation enable register (ICU.IELSRn.DTCE) associated with the event number set in the ICU.DELSRn register is not set to 1. For details on ICU.IELSRn.DTCE and ICU.DELSRn, see section 14, Interrupt Controller Unit (ICU).

17.8.5 Suspending or Restarting DMA Activation

To suspend a DMA activation request, write 0 to the DMAC Event Link select (ICU.DELSRn.DELS[8:0]). To restart the DMA transfer, write the event number to the ICU.DELSRn.DELS[8:0] bit with the settings shown in section 17.3.7, Activating DMAC.

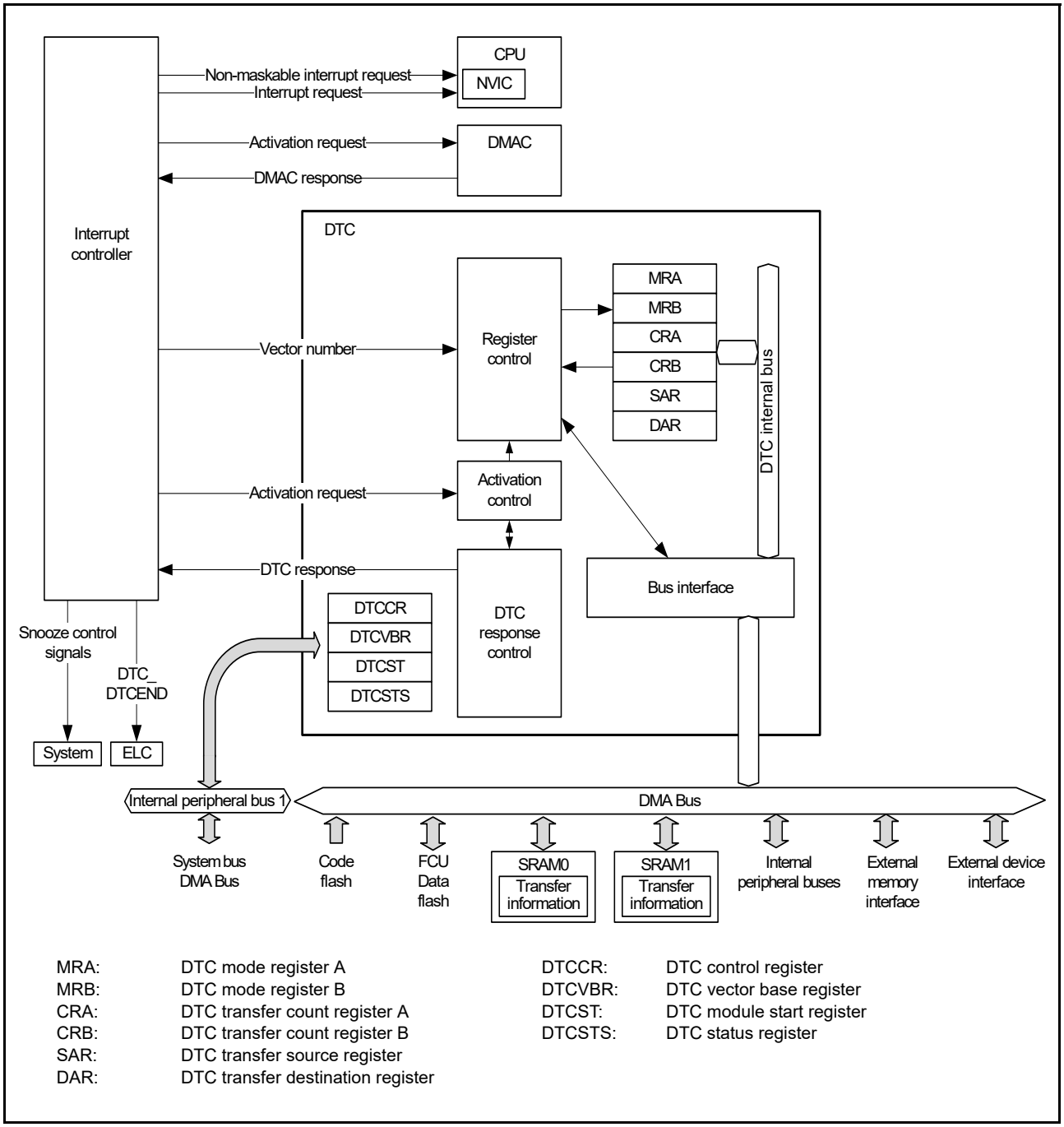
18. Data Transfer Controller (DTC)

18.1 Overview

The Data Transfer Controller (DTC) performs data transfers when activated by an interrupt request. Table 18.1 lists the DTC specifications and Figure 18.1 shows the block diagram.

Table 18.1 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 transfer 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 the DTC activation request from the ICU) • Multiple data units can be transferred on a single activation source (chain transfer) • Chain transfers selectable to either "executed," when the counter is 0, or "always executed."
Transfer space	• 4 GB area from 0000 0000h to FFFF FFFFh except 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	Transfer information read skip can be executed
Write-back skip	When the transfer source or destination address is specified as fixed, a write-back skip is executed
Module stop function	Module stop state can be set



MRA: DTC mode register A
MRB: DTC mode register B
CRA: DTC transfer count register A
CRB: DTC transfer count register B
SAR: DTC transfer source register
DAR: DTC transfer destination register

DTCCR: DTC control register
DTCVBR: DTC vector base register
DTCST: DTC module start register
DTCSTS: DTC status register

Figure 18.1 DTC block diagram

See section 14.1, Overview in section 14, Interrupt Controller Unit (ICU) for the connections between the DTC and NVIC (in the CPU).

18.2 Register Descriptions

The 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.

18.2.1 DTC Mode Register A (MRA)

Address(es): (inaccessible directly from the CPU. See section 18.3.1)

	b7	b6	b5	b4	b3	b2	b1	b0
	MD[1:0]		SZ[1:0]		SM[1:0]		—	—
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as undefined. The write value should be 0.	—
b3, b2	SM[1:0]	Transfer Source Address Addressing Mode	b3 b2 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] bits = 00b, +2 when SZ[1:0] bits = 01b, +4 when SZ[1:0] bits = 10b) 1 1: SAR value is decremented after data transfer. (-1 when SZ[1:0] bits = 00b, -2 when SZ[1:0] bits = 01b, -4 when SZ[1:0] bits = 10b)	—
b5, b4	SZ[1:0]	DTC Data Transfer Size	b5 b4 0 0: Byte (8-bit) transfer 0 1: Halfword (16-bit) transfer 1 0: Word (32-bit) transfer 1 1: Setting prohibited.	—
b7, b6	MD[1:0]	DTC Transfer Mode Select	b7 b6 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited.	—

The MRA cannot be accessed directly from the CPU. The CPU can access the SRAM area (transfer information (n) start address + 03h) and DTC transfer it automatically from and to MRA register. See section 18.3.1, Allocating Transfer Information and DTC Vector Table.

18.2.2 DTC Mode Register B (MRB)

Address(es): (inaccessible directly from the CPU. See section 18.3.1)

	b7	b6	b5	b4	b3	b2	b1	b0
	CHNE	CHNS	DISEL	DTS	DM[1:0]		—	—
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as undefined. The write value should be 0.	—
b3, b2	DM[1:0]	Transfer Destination Address Addressing Mode	b3 b2 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] bits = 00b, +2 when SZ[1:0] bits = 01b, +4 when SZ[1:0] bits = 10b) 1 1: DAR value is decremented after data transfer (-1 when MRA.SZ[1:0] bits = 00b, -2 when SZ[1:0] bits = 01b, -4 when SZ[1:0] bits = 10b)	—
b4	DTS	DTC Transfer Mode Select	0: Select transfer destination as repeat or block area 1: Select transfer source as repeat or block area.	—
b5	DISEL	DTC Interrupt Select	0: An interrupt request to the CPU is generated when specified data transfer is complete 1: An interrupt request to the CPU is generated each time DTC data transfer is performed.	—
b6	CHNS	DTC Chain Transfer Select	0: Chain transfer is continuous 1: Chain transfer occurs only when the transfer counter is changed from 1 to 0 or 1 to CRAH.	—
b7	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. The CPU can access the SRAM area (transfer information (n) start address + 02h) and DTC transfer it automatically from and to the MRB register. See section 18.3.1, Allocating Transfer Information and DTC Vector Table.

DTS bit (DTC Transfer Mode Select)

The DTS bit selects either the transfer source or transfer destination as the repeat area or block area in repeat or block transfer mode.

CHNS bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition. When the CHNE bit is 0, the CHNS setting is ignored. For details on the conditions for chain transfer, see Table 18.3, Chain transfer conditions.

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 of chain transfer, see section 18.4.6, Chain Transfer.

18.2.3 DTC Transfer Source Register (SAR)

Address(es): (inaccessible directly from the CPU. See section 18.3.1)

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

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

x: Undefined

The SAR sets the transfer source start address. It cannot be accessed directly from the CPU. The CPU can access the SRAM area (transfer information (n) start address + 04h) and DTC transfer it automatically from and to the SAR register. See section 18.3.1, Allocating Transfer Information and DTC Vector Table.

Note: Misalignment is prohibited at the DTC transfer.
Bit [0] should be 0 when MRA.SZ[1:0] = 01b, and bit [1] and bit [0] must be 0 when MRA.SZ[1:0] = 10b.

18.2.4 DTC Transfer Destination Register (DAR)

Address(es): (inaccessible directly from the CPU. See section 18.3.1)

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

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

x: Undefined

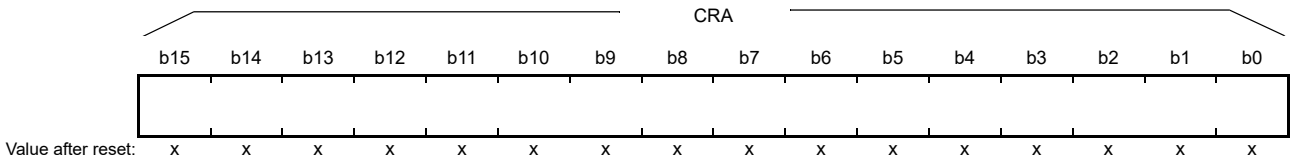
The DAR sets the transfer destination start address. It cannot be accessed directly from the CPU. The CPU can access the SRAM area (transfer information (n) start address + 08h) and DTC transfer it automatically from and to the DAR register. See section 18.3.1, Allocating Transfer Information and DTC Vector Table.

Note: 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.

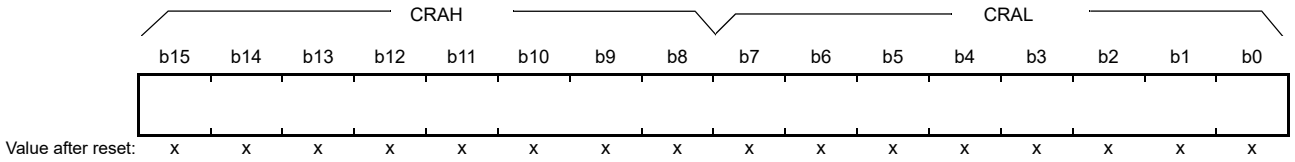
18.2.5 DTC Transfer Count Register A (CRA)

Address(es): (inaccessible directly from the CPU. See section 18.3.1)

- Normal transfer mode



- Repeat transfer mode/block transfer mode



x: Undefined

Symbol	Register name	Description	R/W
CRAL	Transfer Counter A Lower Register	Set transfer count	—
CRAH	Transfer Counter A Upper Register		—

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.

CRA cannot be accessed directly from the CPU. The CPU can access the SRAM area (transfer information (n) start address + 0Eh) and DTC transfer it automatically from and to the CRA register. See section 18.3.1, Allocating Transfer Information and DTC Vector Table.

(1) Normal Transfer mode (MRA.MD[1:0] bits = 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 0001h, FFFFh, and 0000h, respectively. The CRA value is decremented (-1) on each data transfer.

(2) Repeat Transfer mode (MRA.MD[1:0] bits = 01b)

In repeat transfer mode, the CRAH register hold 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 01h, FFh, and 00h, respectively. The CRAL value is decremented (-1) on each data transfer. When it reaches 00h, the CRAH value is transferred to CRAL.

(3) Block Transfer mode (MRA.MD[1:0] bits = 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 01h, FFh, and 00h, respectively. The CRAL value is decremented (-1) on each data transfer. When it reaches 00h, the CRAH value is transferred to CRAL.

18.2.6 DTC Transfer Count Register B (CRB)

Address(es): (inaccessible directly from the CPU. See section 18.3.1)

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

x: Undefined

The CRB sets the block transfer count for block transfer mode. The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, 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. The CPU can access the SRAM area (transfer information (n) start address + 0Ch) and DTC transfer it automatically from and to the CRB register. See section 18.3.1, Allocating Transfer Information and DTC Vector Table.

18.2.7 DTC Control Register (DTCCR)

Address(es): DTC.DTCCR 4000 5400h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	RRS	—	—	—	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b4	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
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RRS bit (DTC Transfer Information Read Skip Enable)

The RRS 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 value 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.

18.2.8 DTC Vector Base Register (DTCVBR)

Address(es): DTC.DTCVBR 4000 5404h

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

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

Bit	Bit name	Description	R/W
b31 to b0	DTC Vector Base Address	Set the DTC Vector Base Address (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 0000 0000h to FFFF FFFFh (4-GB) in 1-KB units.

18.2.9 DTC Module Start Register (DTCST)

Address(es): DTC.DTCST 4000 540Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTCST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DTCST	DTC Module Start	0: DTC module stop 1: DTC module start.	R/W
b7 to b1	—	Reserved	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 is complete.

DTCST must be set to 0 before making a transition to one of the following:

- Module stop state
- Software Standby mode without Snooze mode transition.

For details on these transitions, see section 18.10, Module-Stop Function, and section 11, Low Power Modes.

18.2.10 DTC Status Register (DTCSTS)

Address(es): DTC.DTCSTS 4000 540Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit name	Description	R/W
b7 to b0	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 (the value of the ACT flag is 1).	R
b14 to b8	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b15	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, these 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 value of the ACT flag is 1, indicating a DTC transfer is in progress, and invalid if the value of the ACT flag is 0, indicating no current DTC transfer is in progress.

ACT flag (DTC Active Flag)

This 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.

18.3 Activation Sources

The DTC is activated by an interrupt request. Setting the ICU.IELSRn.DTCE bit of the ICU to 1 enables activation of the DTC by the associated interrupt. The number of selector output *n* set in ICU.IELSR is defined as the interrupt vector number, where n = 0 to 63. For an enabled interrupt, the specific DTC interrupt source associated with each interrupt vector number n is selected by ICU.IELSRn.IELS[8:0], where n = 0 to 63.

For the setup of ICU.IELSRn.IELS[8:0] (n = 0 to 63), see Table 14.4, Event table in section 14, Interrupt Controller Unit (ICU). For activation by software, see section 19.2.2, Event Link Software Event Generation Register n (ELSEGRn) (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 transfer for that single request is complete, regardless of the priority of the requests. When multiple activation requests are generated during a DMAC or DTC transfer, a request with the highest priority on completion of the transfer is accepted. When multiple activation requests are generated while the DTC module start bit (DTCST.DTCST) is 0, the DTC accepts the request with the highest priority at the time when the bit is subsequently set to 1. The small interrupt vector number is high 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 bit of the activation source is set to 0 at the start of the data transfer.

18.3.1 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 18.2 shows the association between the DTC vector table and transfer information. Figure 18.3 shows the allocation of transfer information in the SRAM area.

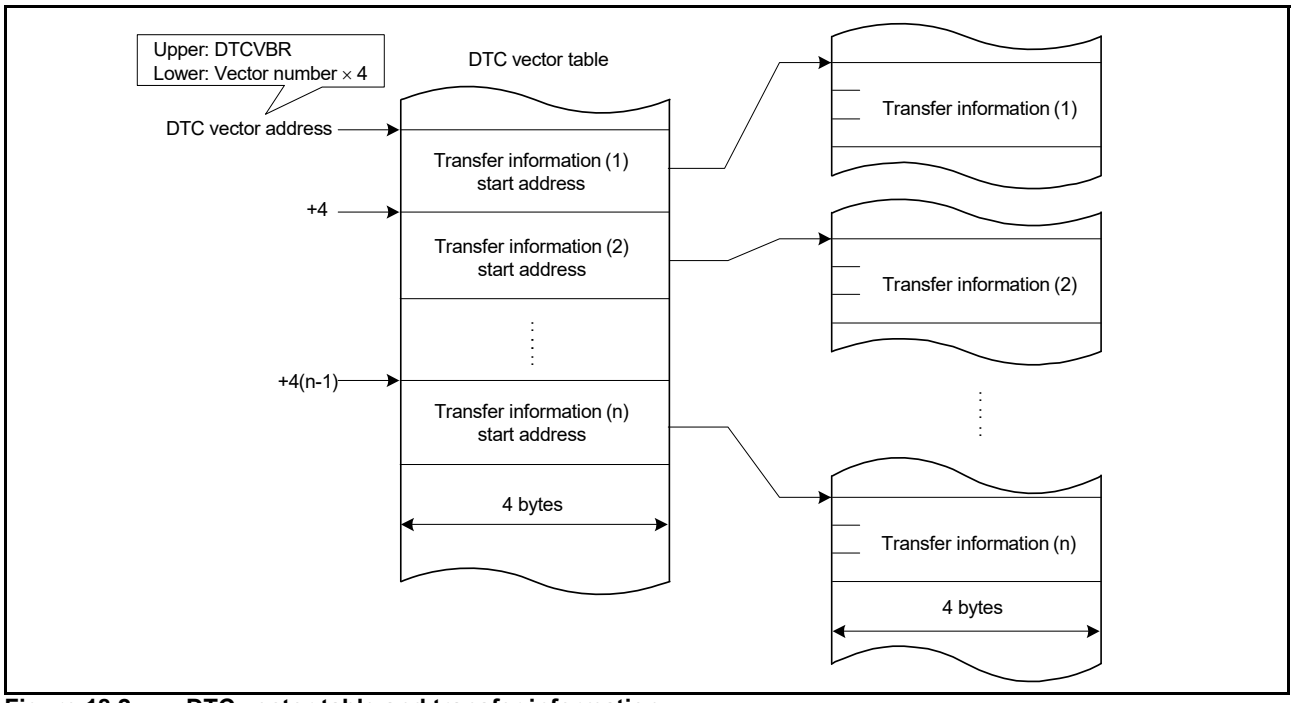


Figure 18.2 DTC vector table and transfer information

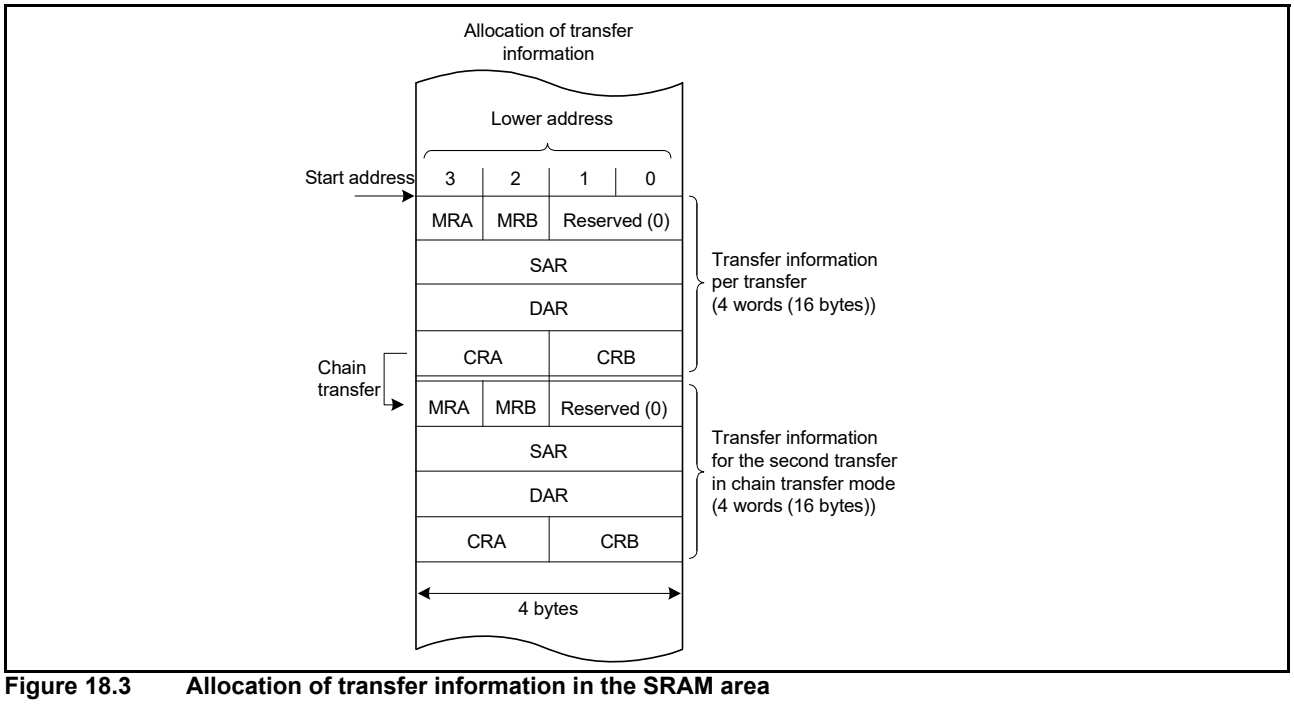


Figure 18.3 Allocation of transfer information in the SRAM area

18.4 Operation

The DTC transfers data in accordance with 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 then reads the transfer information from the transfer information store address referenced by the DTC vector and transfers 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.

There are three transfer modes:

- 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 individually after the data transfer.

Table 18.2 describes the DTC transfer modes.

Table 18.2 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) / 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 18.4 shows the operation flowchart of the DTC. Table 18.3 lists the chain transfer conditions. The combination of control information for the second and subsequent transfers are omitted in this table.

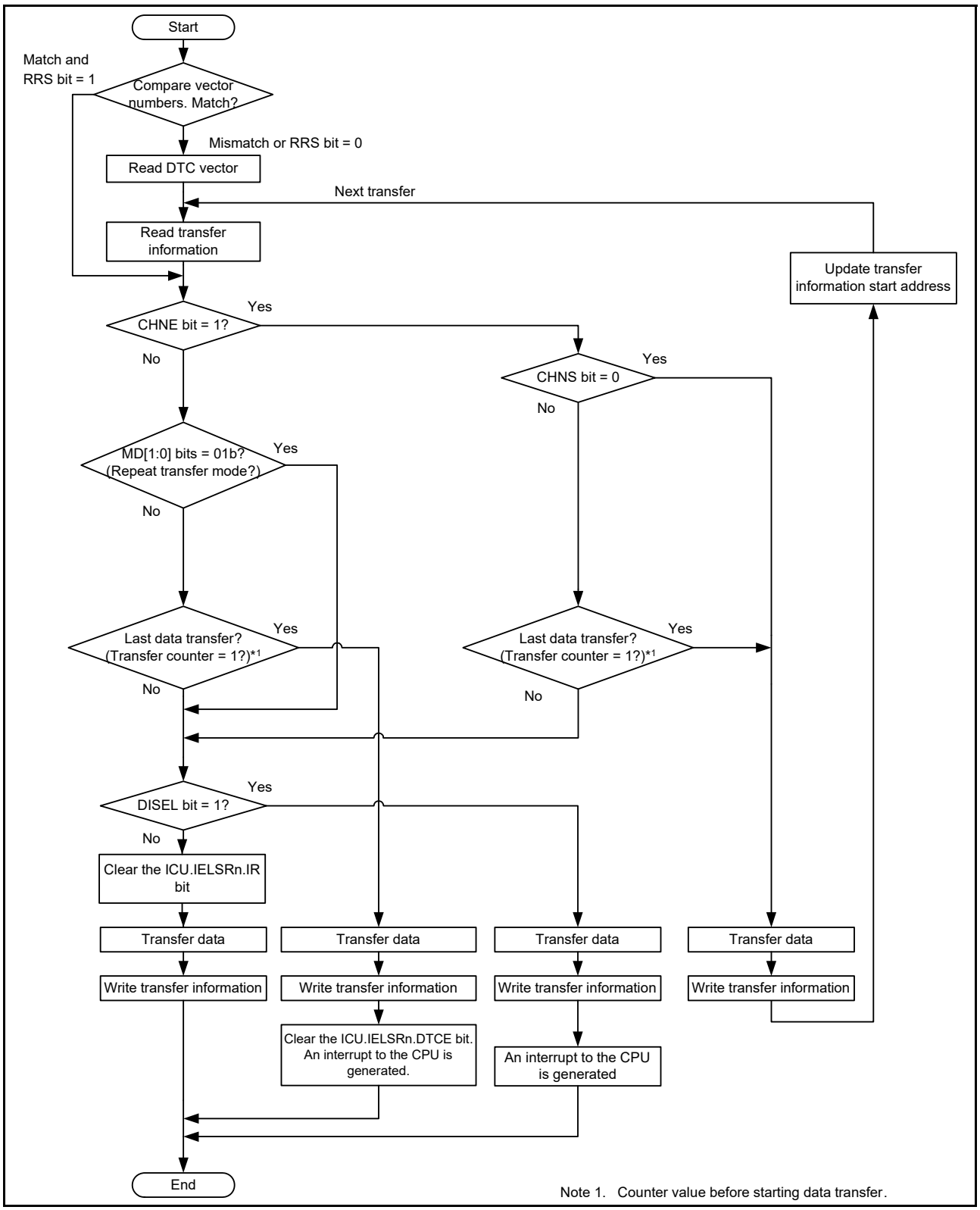


Figure 18.4 DTC operation flow

Table 18.3 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 counters used depend 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 the two operations above.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The conditions for the combination of the second transfer and CHNE bit = 1 is omitted.

18.4.1 Transfer Information Read Skip Function

Reading of vector addresses and transfer information can be skipped through the setting in the DTCCR.RRS bit. When a DTC activation request is generated, the current DTC vector number is compared to 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, or when the transfer counter (CRB register) becomes 0 during the previous block transfer, transfer information is read regardless of the value of the RRS bit. Figure 18.12 shows an example of a transfer information read skip.

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.

18.4.2 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 18.4 lists the transfer information write-back skip conditions and associated registers. The CRA and CRB registers are written back, and the write-back of the MRA and MRB registers is skipped.

Table 18.4 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		

18.4.3 Normal Transfer Mode

This 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 transfer destination addresses can be individually set to increment, decrement, or remain fixed. This mode enables an interrupt request to the CPU to be generated at the end of a specified-count transfer.

Table 18.5 lists register functions in normal transfer mode, and Figure 18.5 shows the memory map for normal transfer mode.

Table 18.5 Register functions in normal transfer mode

Register	Description	Value written back by writing transfer information
SAR	Transfer source address	Increment/decrement/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.

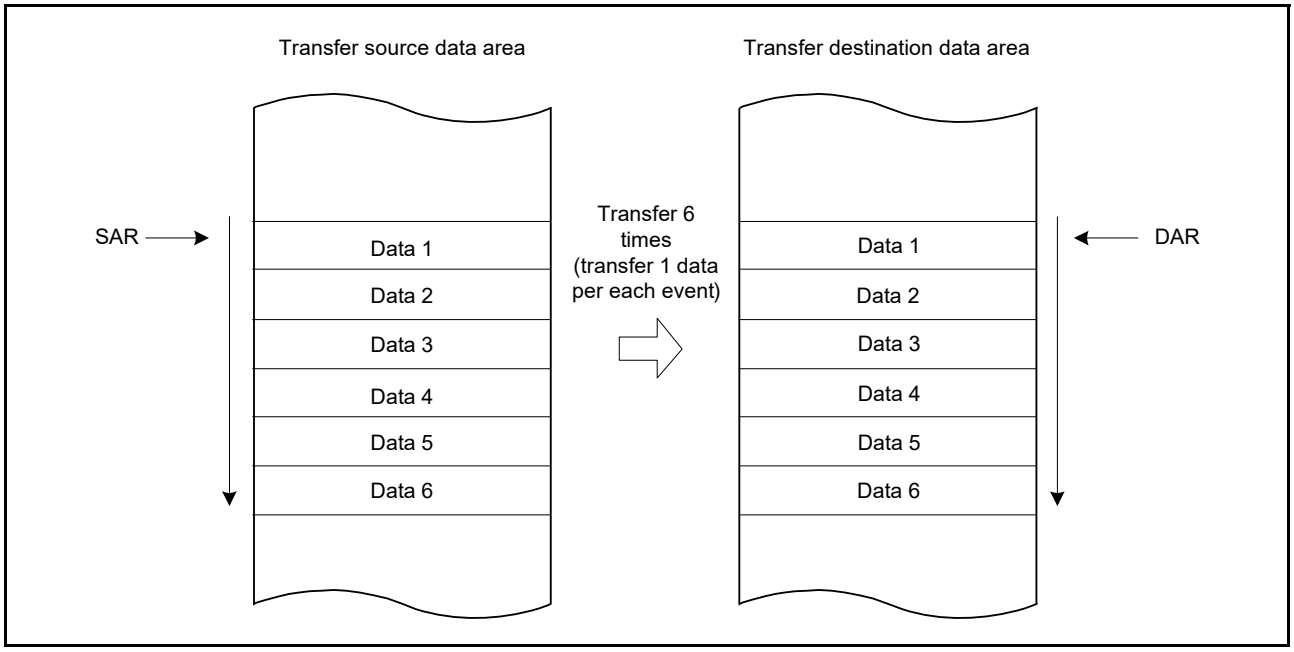


Figure 18.5 Memory map of normal transfer mode (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRA=0006h)

18.4.4 Repeat Transfer Mode

This 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-count transfer 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 00h 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 become 00h, 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 is complete.

Table 18.6 lists the register functions in repeat transfer mode, and Figure 18.6 shows the memory map for repeat transfer mode.

Table 18.6 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/fixed*1 (When the MRB.DTS bit is 1) SAR register initial value
DAR	Transfer destination address	Increment/decrement/fixed*1	(When the MRB.DTS bit is 0) DAR register initial value (When the MRB.DTS bit is 1) Increment/decrement/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.

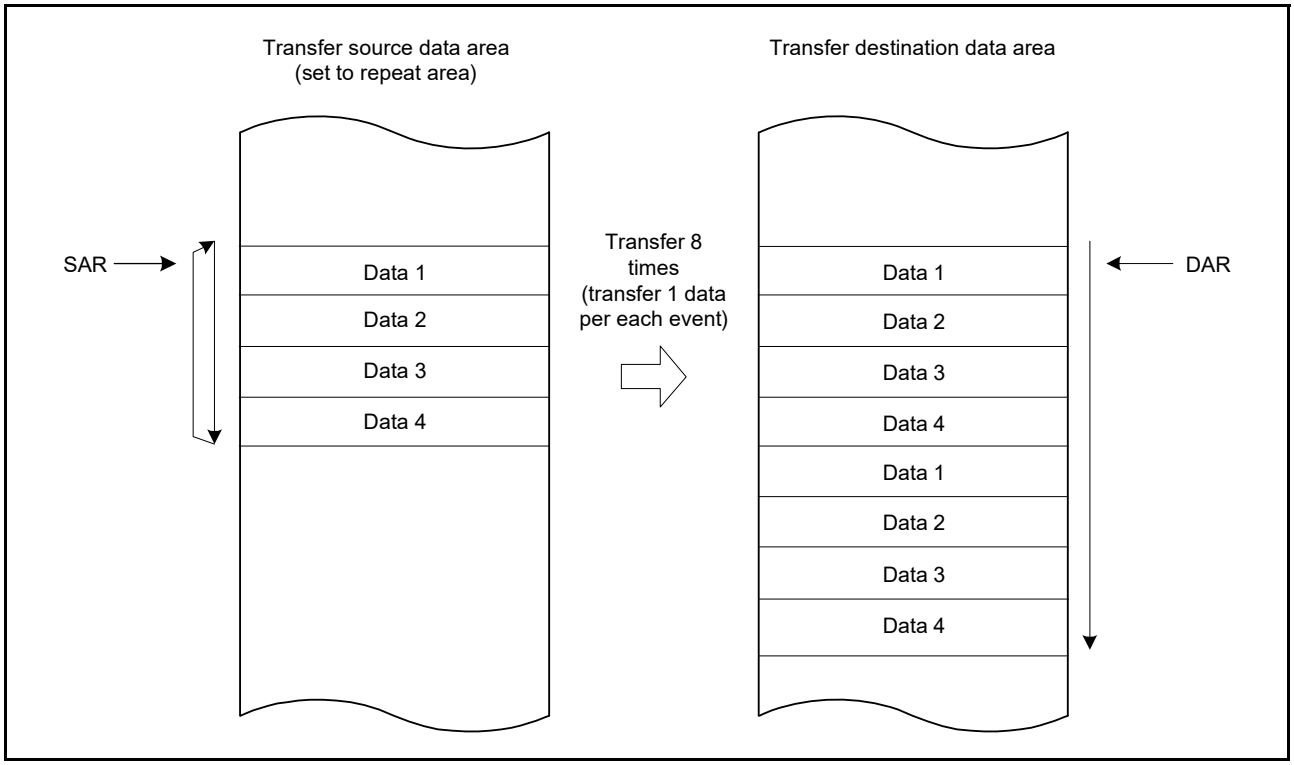


Figure 18.6 Memory map of repeat transfer mode - transfer source: repeat area (MRA.SM[1:0] = 10b, MRB.DM[1:0] =10b, CRAH=04h)

18.4.5 Block Transfer Mode

This 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 bit = 1 or the DAR register when the DTS bit = 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 18.7 lists register functions in block transfer mode, and Figure 18.7 shows the memory map for block transfer mode.

Table 18.7 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/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/fixed*1
CRAH	Retains block size	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB - 1

Note 1. Write-back is skipped in address-fixed mode.

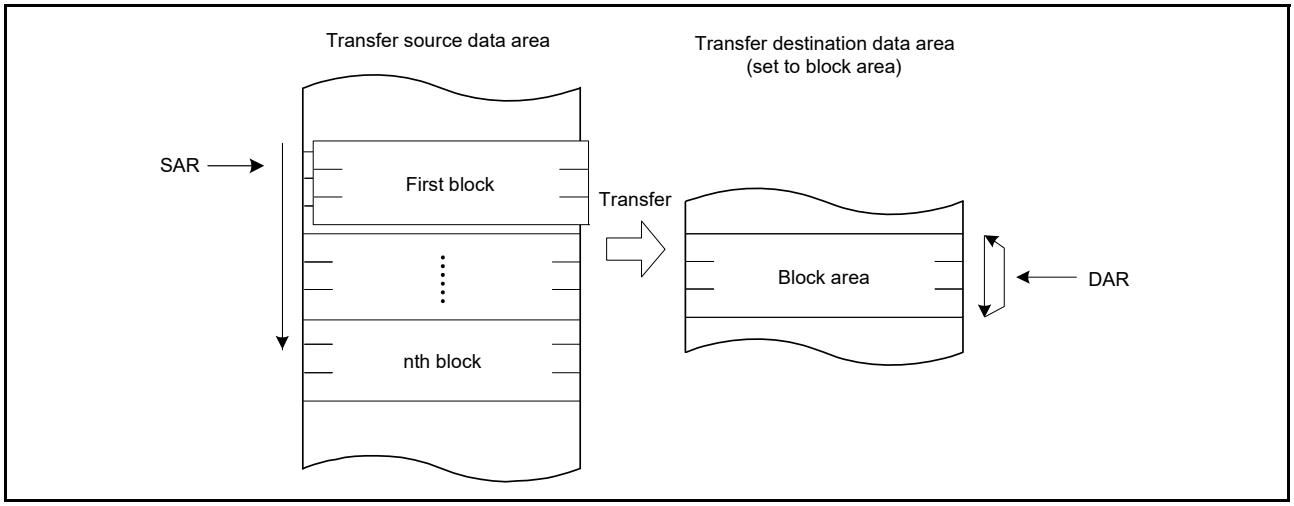


Figure 18.7 Memory map of block transfer mode

18.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single activation source. If 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 bit of the activation source.

The SAR, DAR, CRA, CRB, MRA, and MRB registers can be set independently of each other to define data transfer. Figure 18.8 shows a chain transfer operation.

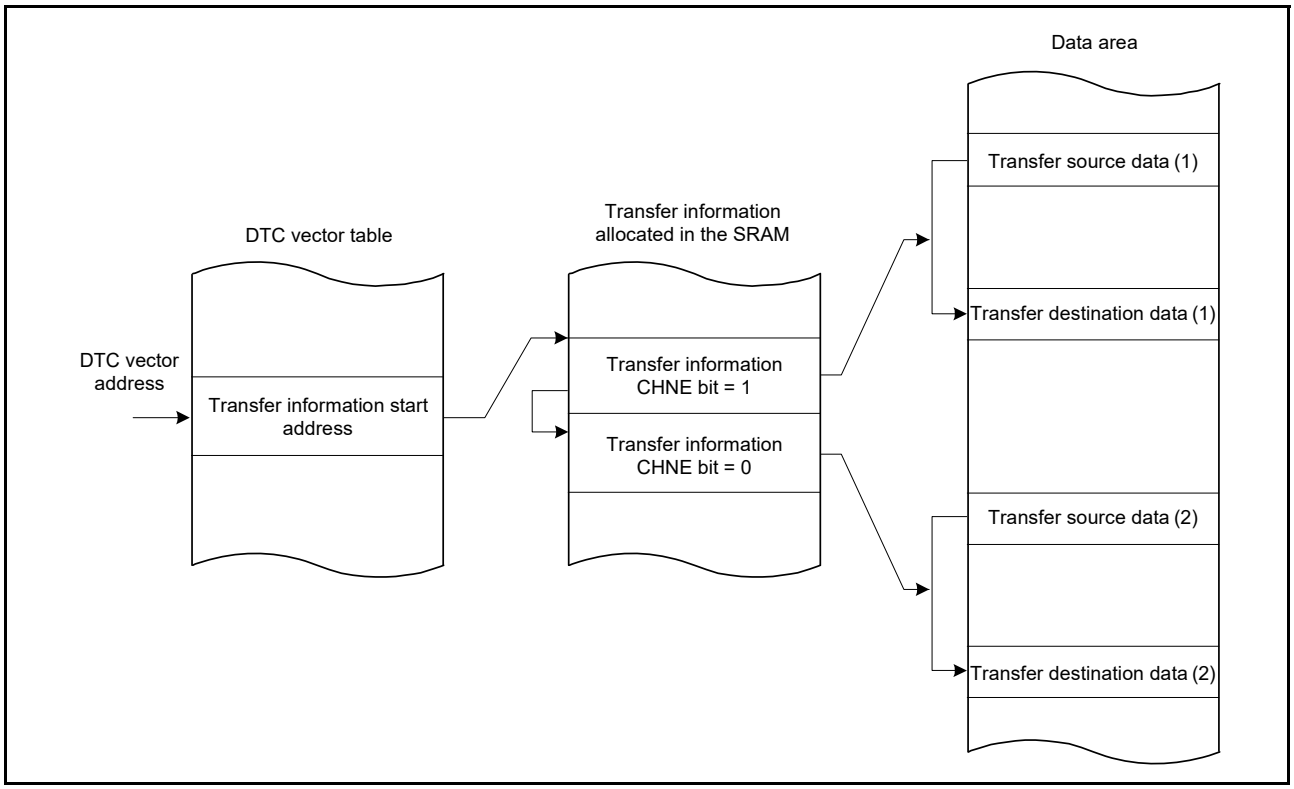


Figure 18.8 Chain transfer operation

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 18.3, Chain transfer conditions.

18.4.7 Operation Timing

Figure 18.9 to Figure 18.12 are timing diagrams that show the minimum number of execution cycles.

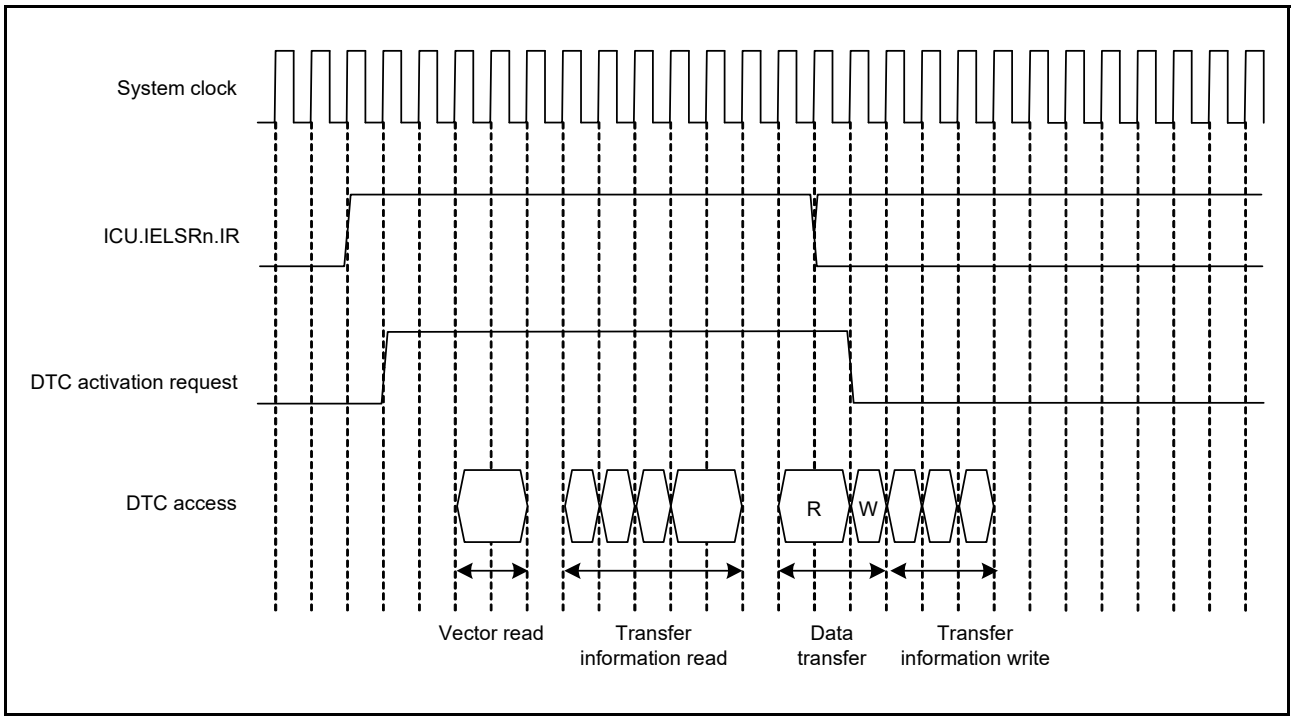


Figure 18.9 Example (1) of DTC operation timing (normal transfer mode, repeat transfer mode)

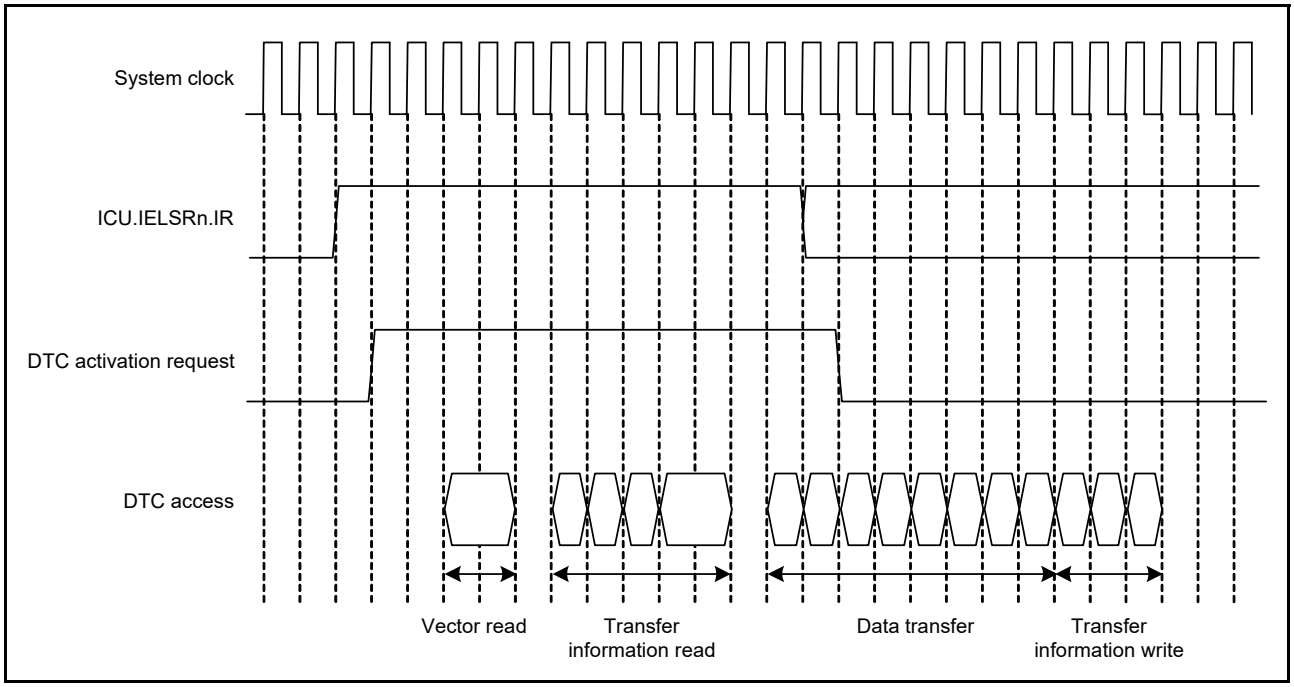


Figure 18.10 Example (2) of DTC operation timing (block transfer mode, block size = 4)

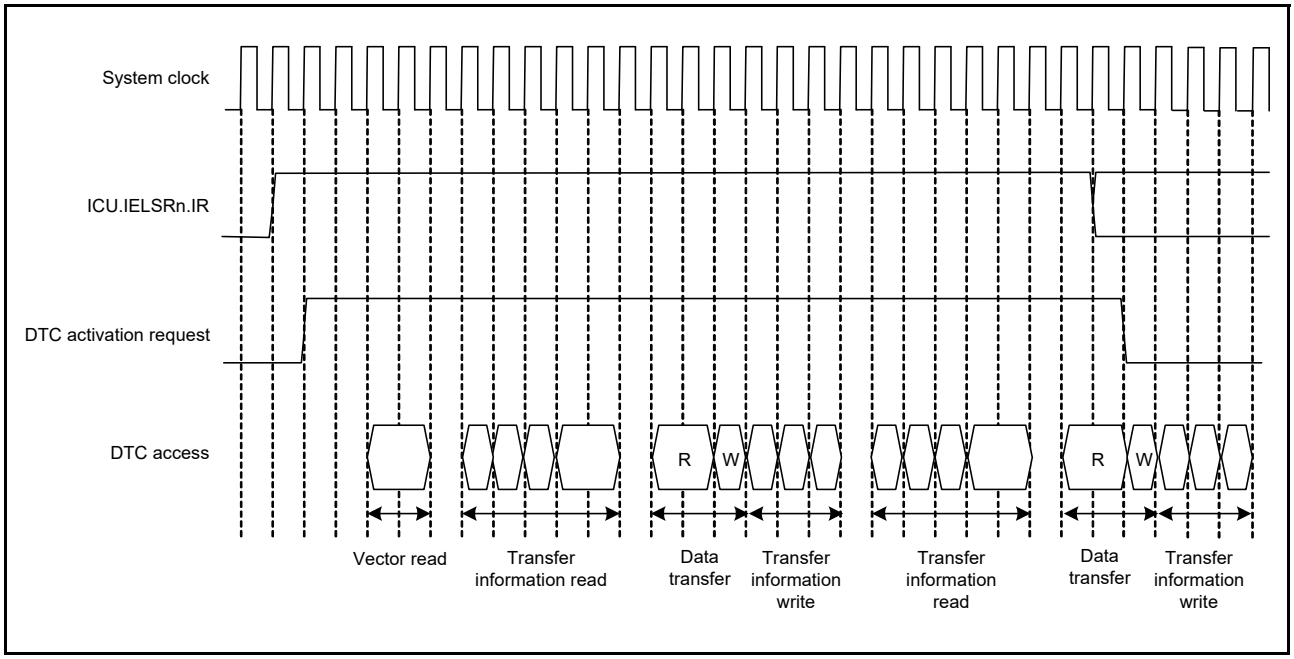
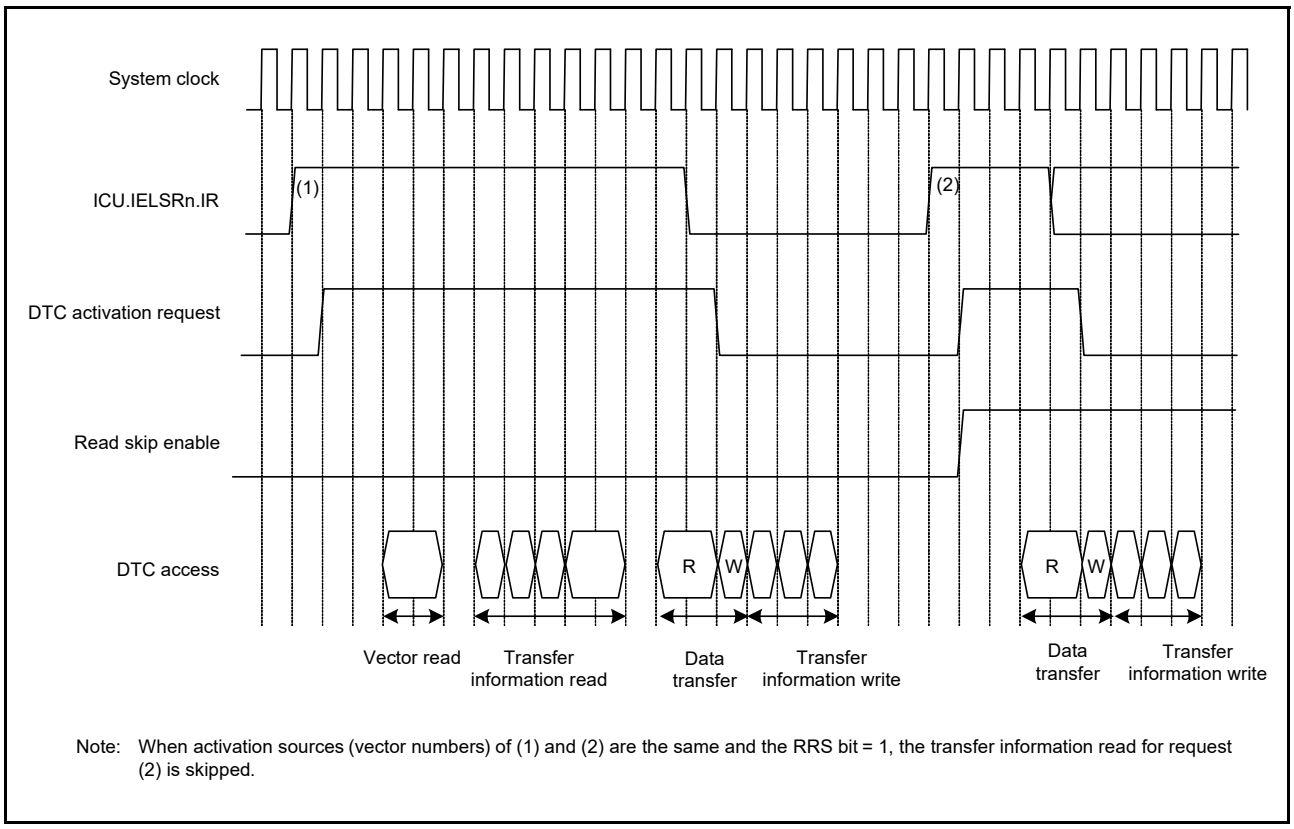


Figure 18.11 Example (3) of DTC operation timing (chain transfer)



Note: When activation sources (vector numbers) of (1) and (2) are the same and the RRS bit = 1, the transfer information read for request (2) is skipped.

Figure 18.12 Example of operation when transfer information read skip is executed (vector, transfer information, transfer destination data on the SRAM, and transfer source data on the peripheral module)

18.4.8 Execution Cycles of DTC

Table 18.8 lists the execution cycles of single data transfer of the DTC.

For the order of the execution states, see section 18.4.7, Operation Timing.

Table 18.8 Execution cycles of DTC

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 is applied.

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 system clocks (ICLK) for "+ 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 47, SRAM, section 48, Flash Memory, and section 15, 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.
This table does not include the time until DTC data transfer starts after the DTC activation source becomes active.

18.4.9 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 15, Buses.

18.5 DTC Setting Procedure

Before using the DTC, set the DTC vector base register (DTCVBR). Figure 18.13 shows the procedure to set the DTC.

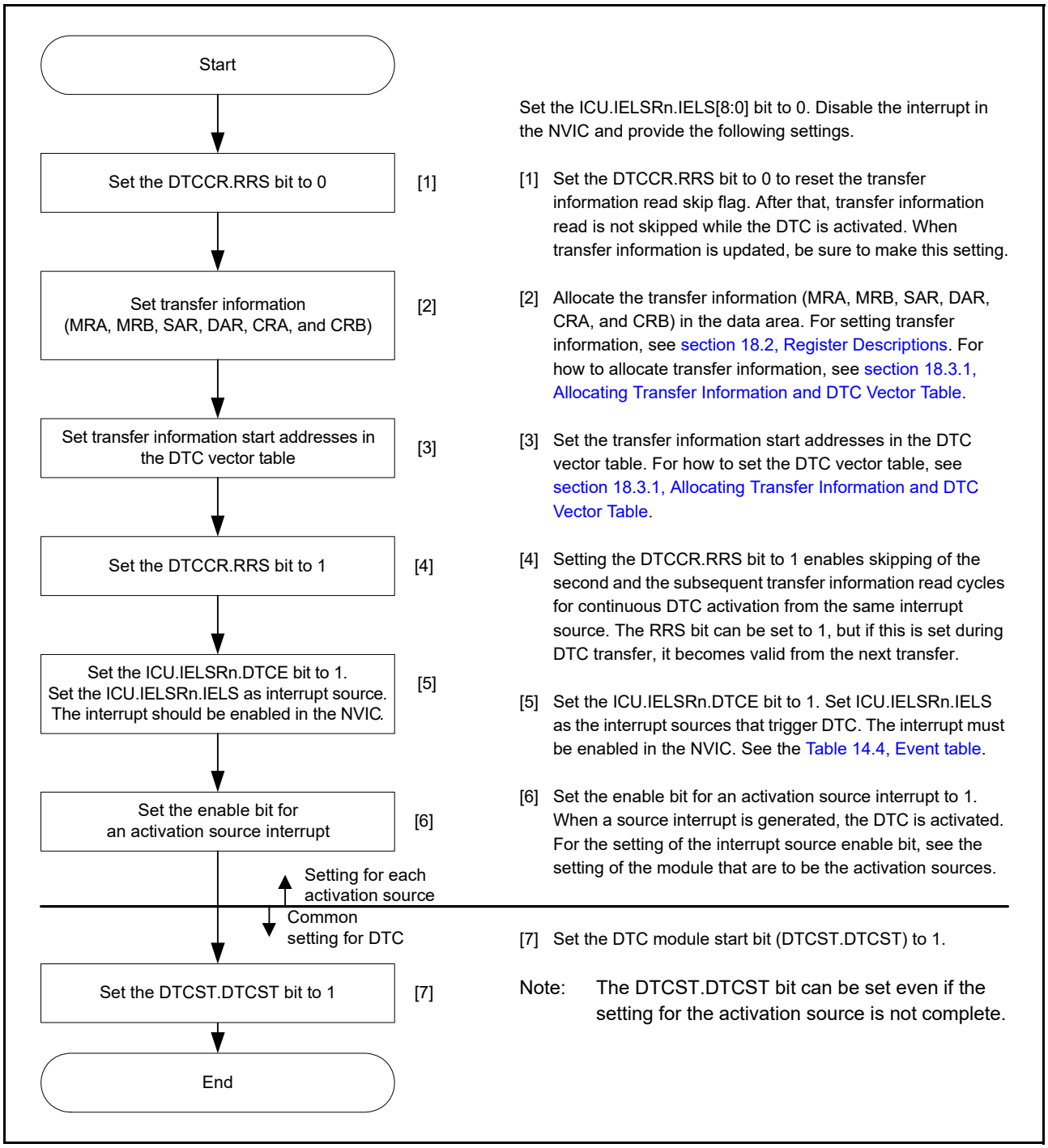


Figure 18.13 Procedure for setting the DTC

18.6 Examples of DTC Usage

18.6.1 Normal Transfer

This section provides an example of DTC usage and its application in the reception of 128 bytes of data from an SCI.

(1) Transfer Information Setting

In the MRA register, select a fixed source address (MRA.SM[1:0] bits = 00b), normal transfer mode (MRA.MD[1:0] bits = 00b), and byte-sized transfer (MRA.SZ[1:0] bits = 00b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] bits = 10b) and single data transfer by a single interrupt (MRB.CHNE bit = 0 and MRB.DISEL bit = 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 (0080h) in the CRA register. The CRB register can be set to any value.

(2) DTC Vector Table Setting

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

(3) ICU Setting and DTC Module Activation

Set the ICU.IELSRn.DTCE bit to 1 and set ICU.IELSRn.IELS as the SCI interrupt. The interrupt must be enabled in the NVIC. Set the DTCST.DTCST bit to 1.

(4) SCI Setting

Enable the RXI 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 allows the CPU to accept receive error interrupts.

(5) DTC Transfer

Every time a reception of 1 byte by the SCI completes, an 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 RXI interrupt request is generated for the CPU. Complete the process in the handling routine for this interrupt.

18.6.2 Chain Transfer

This section provides an example of chain transfer by the DTC and describes its employment in the output of pulses by a 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 transfer, normal transfer mode is specified for transfer to the GPTm.GTCCRC register. For the second transfer of the chained transfer, normal transfer mode is specified for transfer to the GPTm.GTCCRE register. For the third transfer, normal transfer mode is specified for transfer to the GPTm.GTPBR register. 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 bit = 0.

The following example shows how to use the counter overflow interrupt with a GPT320.GTPR register as an activating source for the DTC.

(1) First Transfer Information Setting

Set up transfer to the GPT320.GTCCRC register:

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] bits = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] bits = 00b) and word-sized transfer (MRA.SZ[1:0] bits = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] bits = 00b) and set up chain transfer (MRB.CHNE bit = 1 and MRB.CHNS bit = 0).

4. Set the SAR to the first address of the data table.
5. Set the DAR register to the address of the GPT320.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 transfer to the GPT320.GTCCRE register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] bits = 10b).
2. Set the transfer in normal transfer mode (MRA.MD[1:0] bits = 00b) and word-sized transfer (MRA.SZ[1:0] bits = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] bits = 00b) and set up chain transfer (MRB.CHNE bit = 1, MRB.CHNS bit = 0).
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT320.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 GPT320.GTPBR register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] bits = 10b).
2. Set the transfer in normal transfer mode (MRA.MD[1:0] bits = 00b) and word-sized transfer (MRA.SZ[1:0] bits = 10b)
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] bits = 00b) and for the single data transfer per interrupt (MRB.CHNE bit = 0, MRB.DISEL bit = 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 GPT320.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 GPT320.GTPBR immediately after the transfer control information for use in the GPT320.GTCCRC and GPT320.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 GPT320.GTCCRC and GPT320.GTCCRE registers starts.

(6) ICU Setting and DTC Module Activation

1. Set the ICU.IELSRn.DTCE bit associated with the GPT320 counter overflow interrupt
2. Set the ICU.IELSRn.IELS[8:0] to 182 (B6h) for the GPT320 counter overflow.
3. Set the DTCST.DTCST bit to 1.

(7) GPT Setting

1. Set the GPT320.GTIOR register so that the GTCCRA and GTCCRB registers operate as output compare registers.
2. Set the default PWM timer compare values in the GPT320.GTCCRA and GPT320.GTCCRB registers and the next PWM timer compare values in the GPT320.GTCCRC and GPT320.GTCCRE registers.
3. Set the default PWM timer period values in the GPT320.GTPR register and the next PWM timer period values in the GPT320.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 GPT320.GTSTR.CSTRT bits to 1 to start the GPT320.GTCNT counter.

(9) DTC Transfer

Every time a GPT320 counter overflow is generated with the GPT320.GTPR register, the next PWM timer compare values are transferred to the GPT320.GTCCRC and GPT320.GTCCRE registers. The setting for the next PWM timer period is transferred to the GPT320.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 GPT320 counter overflow interrupt request is issued for the CPU. Complete the process for this interrupt in the handling routine.

18.6.3 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 transfers enables transfers to be repeated 256 times or more.

The following procedure shows an example of configuring a 128 KB input buffer, where the input buffer is set so that its lower address starts with 0000h. Figure 18.14 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 = 0000h (65,536) 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 65,536 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 20 0000h to 21 FFFFh, prepare 21h and 20h.
3. For the second data transfer:
   f. Set the repeat transfer mode (with the source as the repeat area) to reset the transfer destination address of the first data transfer.
   g. Specify the upper 8 bits of the DAR register in the first transfer information area for the transfer destination.
   h. Set the MRB.CHNE bit = 0 (chain transfer is disabled).
   i. Set the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
   j. When setting the input buffer to 20 0000h to 21 FFFFh, also set the transfer counter to 2.
4. The first data transfer is performed by an interrupt 65,536 times. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the upper 8 bits of the transfer source address of the first data transfer to 21h. The lower 16 bits of the transfer destination address and the transfer counter of the first data transfer have become 0000h.
5. In succession, the first data transfer is performed by an interrupt 65,536 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 upper 8 bits of the transfer source address of the first data transfer to 20h. The lower 16 bits of the transfer destination address and the transfer counter of the first data transfer have become 0000h.
6. Steps 4 and 5 are repeated indefinitely. Because the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

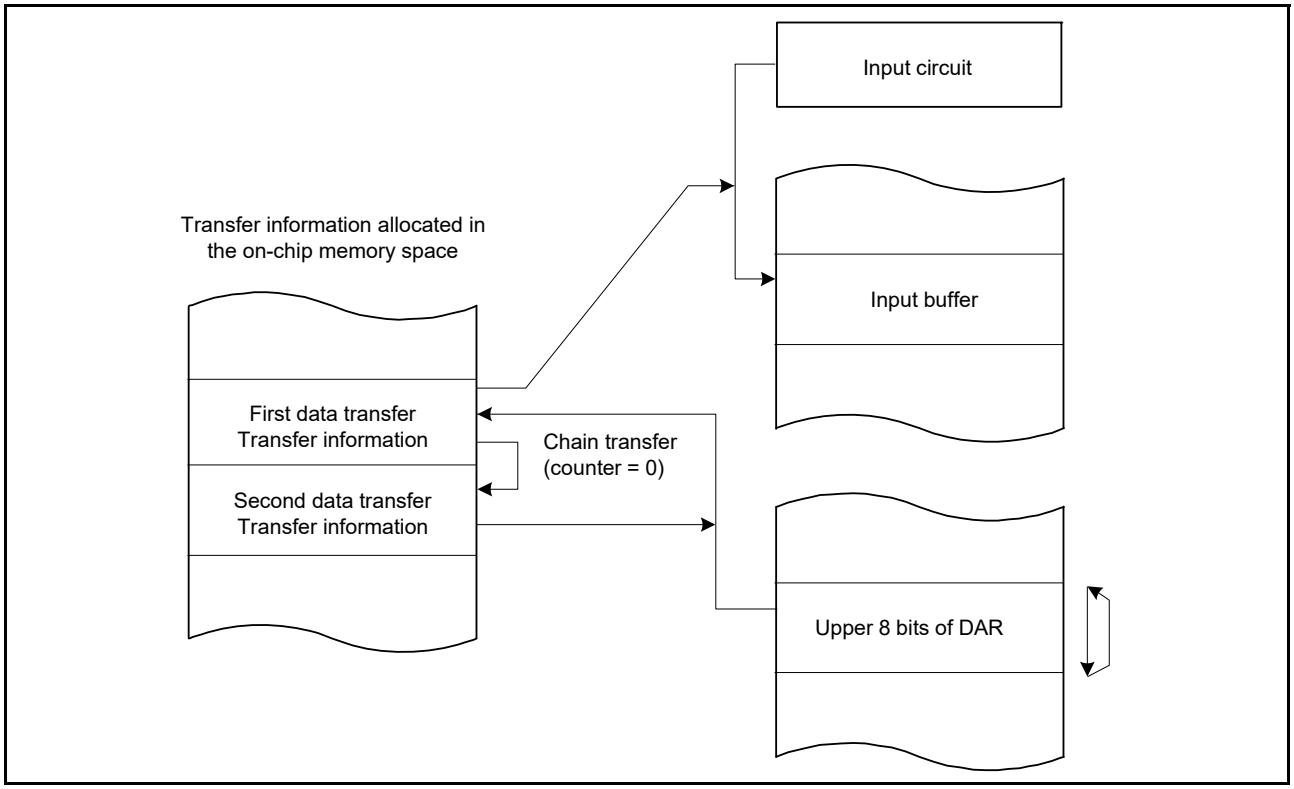


Figure 18.14 Chain transfer when counter = 0

18.7 Interrupt Source

When the DTC finishes data transfer of the specified count or when data transfer with the MRB.DISEL set to 1 is complete, a DTC activation source generates an interrupt to the CPU. Interrupts to the CPU are controlled according to the settings in the NVIC and ICU.IELSRn.IELS[8:0]. See section 14, 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.

18.8 Event Link

The DTC is capable of producing an event link request on completion of one transfer request. When the destination for the transfer is an external bus, the event link request is issued after completion of writing to the write buffer rather than after completion of writing to the actual transfer destination.

18.9 Snooze Control Interface

If a return to Software Standby mode from Snooze mode through the DTC, set the SYSTEM.SNZEDCR.DTCZRED or SYSTEM.SNZEDCR.DTCNZRED to 1. See section 11.8.3, Return to Software Standby Mode.

SYSTEM.SNZEDCR.DTCZRED enables or disables a snooze end request on completion of the last DTC transmission completion, detected on DTC transmission completion of CRA and CRB are 0.

SYSTEM.SNZEDCR.DTCNZRED enables or disables a snooze end request on a not last DTC transmission completion (CRA and CRB is not 0), detected on DTC transmission completion of CRA and CRB are not 0.

18.10 Module-Stop Function

Before transitioning to the module-stop function, Software Standby mode without Snooze mode transition, set the DTCST.DTCST bit to 0, and then perform the following. The DTC is available in Snooze mode by setting SYSTEM.SNZCR.SNZDTCEN to 1. See section 11, Low Power Modes.

(1) Module-Stop Function

Writing 1 to the MSTPCRA.MSTPA22 bit enables the module-stop function of the DTC. If the DTC transfer is in progress at the time, 1 is written to the MSTPCRA.MSTPA22 bit. The transition to the module-stop state proceeds after 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 11.7.1, Transition to Software Standby Mode.

If DTC transfer operations are in progress at the time the WFI instruction is executed, the transition to Software Standby mode follows the completion of the DTC transfer.

When the snooze control circuit receives a snooze request in Software Standby mode, the MCU transfers to Snooze mode. See section 11.8.1, 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 making a transition to Software Standby mode, set the DTCST.DTCST bit to 1. To return to Software Standby mode through the DTC, set the SYSTEM.SNZEDCR.DTCZRED or SYSTEM.SNZEDCR.DTCNZRED to 1. See section 11.8.3, Return to Software Standby Mode. The DTC activation request from the ICU is stopped during Software Standby mode but not during Snooze mode.

(3) Notes on Module Stop Function

For the WFI instruction and the register setting procedure, see section 11, Low Power Modes.

To perform a DTC transfer after returning from a low power consumption mode without 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 14.4.2, Selecting Interrupt Request Destinations, then execute a WFI instruction. If DTC operation is enabled in Snooze mode, do not use the module-stop function of the DTC.

18.11 Usage Notes

18.11.1 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.

19. Event Link Controller (ELC)

19.1 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 19.1 lists the ELC specifications and Figure 19.1 shows the block diagram.

Table 19.1 ELC specifications

Parameter	Description
Event link function	• 181 types of event signals can be directly connected to modules.The ELC can generate ELC event signal, and events that activate the DTC.
Module stop function	Module stop state can be set

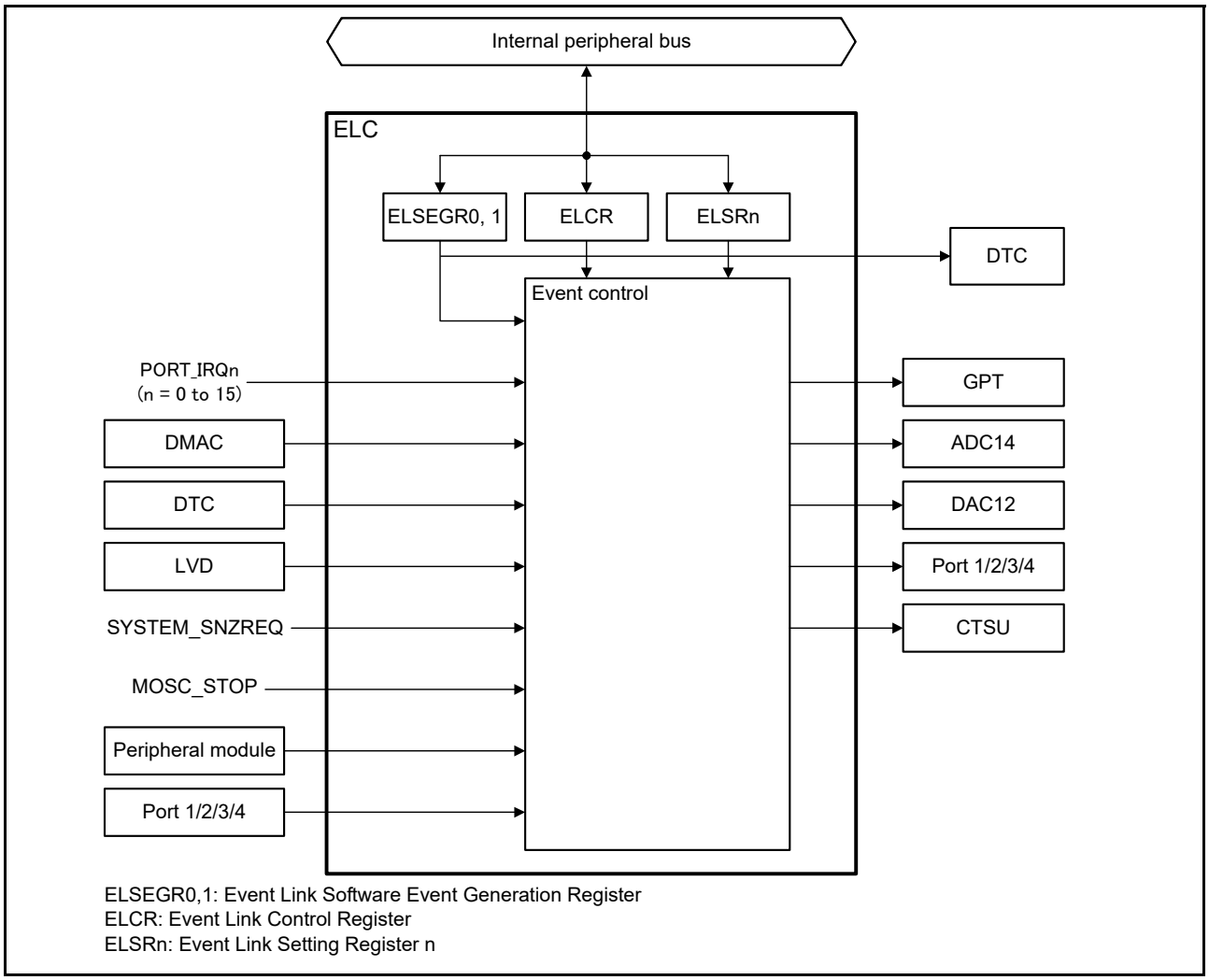


Figure 19.1 ELC block diagram (n = 0 to 9, 12 to 18)

19.2 Register Descriptions

19.2.1 Event Link Controller Register (ELCR)

Address(es): ELC.ELCR 4004 1000h

	b7	b6	b5	b4	b3	b2	b1	b0
	ELCON	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ELCON	All Event Link Enable	0: ELC function disabled 1: ELC function enabled.	R/W

The ELCR register controls the ELC operation.

19.2.2 Event Link Software Event Generation Register n (ELSEGRn) (n = 0, 1)

Address(es): ELC.ELSEGR0 4004 1002h, ELC.ELSEGR1 4004 1004h

	b7	b6	b5	b4	b3	b2	b1	b0
	WI	WE	—	—	—	—	—	SEG
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SEG	Software Event Generation	0: Normal operation 1: Software event is generated.	W
b5 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	WE	SEG Bit Write Enable	0: Write to SEG bit disabled 1: Write to SEG bit enabled.	R/W
b7	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 this 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.

19.2.3 Event Link Setting Register n (ELSRn) (n = 0 to 9, 12 to 18)

Address(es): ELC.ELSR0 4004 1010h, ELC.ELSR1 4004 1014h, ELC.ELSR2 4004 1018h, ELC.ELSR3 4004 101Ch, ELC.ELSR4 4004 1020h, ELC.ELSR5 4004 1024h, ELC.ELSR6 4004 1028h, ELC.ELSR7 4004 102Ch, ELC.ELSR8 4004 1030h, ELC.ELSR9 4004 1034h, ELC.ELSR12 4004 1040h, ELC.ELSR13 4004 1044h, ELC.ELSR14 4004 1048h, ELC.ELSR15 4004 104Ch, ELC.ELSR16 4004 1050h, ELC.ELSR17 4004 1054h, ELC.ELSR18 4004 1058h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ELS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	ELS[8:0]	Event Link Select	b8 b0 000000000:Event output disabled for the associated peripheral module 000000001 to 111000101: Number setting for the event signal to be linked. Other settings are prohibited.	R/W
b15 to b9	—	Reserved	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 for each peripheral module. Table 19.2 shows the associations between the ELSRn register and the peripheral modules. Table 19.3 shows the associations between the event signal names set in the ELSRn register and the signal numbers.

Table 19.2 Associations between the ELSRn register 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	ADC14A	ELC_AD00
ELSR9	ADC14B	ELC_AD01
ELSR12	DAC12 channel 0	ELC_DA0
ELSR13	DAC12 channel 1	ELC_DA1
ELSR14	PORT 1	ELC_PORT1
ELSR15	PORT 2	ELC_PORT2
ELSR16	PORT 3	ELC_PORT3
ELSR17	PORT 4	ELC_PORT4
ELSR18	CTSU	ELC_CTSU

Table 19.3 Association between event signal names set in ELSRn.ELS bits and signal numbers (1 of 5)

Event number	Source of interrupt request generation	Name	Description
001h	Port	PORT_IRQ0*1	External pin interrupt 0
002h		PORT_IRQ1*1	External pin interrupt 1
003h		PORT_IRQ2*1	External pin interrupt 2
004h		PORT_IRQ3*1	External pin interrupt 3
005h		PORT_IRQ4*1	External pin interrupt 4
006h		PORT_IRQ5*1	External pin interrupt 5
007h		PORT_IRQ6*1	External pin interrupt 6
008h		PORT_IRQ7*1	External pin interrupt 7
009h		PORT_IRQ8*1	External pin interrupt 8
00Ah		PORT_IRQ9*1	External pin interrupt 9
00Bh		PORT_IRQ10*1	External pin interrupt 10
00Ch		PORT_IRQ11*1	External pin interrupt 11
00Dh		PORT_IRQ12*1	External pin interrupt 12
00Eh		PORT_IRQ13*1	External pin interrupt 13
00Fh		PORT_IRQ14*1	External pin interrupt 14
010h		PORT_IRQ15*1	External pin interrupt 15
020h	DMAC0	DMAC0_INT	DMAC transfer end 0
021h	DMAC1	DMAC1_INT	DMAC transfer end 1
022h	DMAC2	DMAC2_INT	DMAC transfer end 2
023h	DMAC3	DMAC3_INT	DMAC transfer end 3
02Ah	DTC	DTC_DTCEND*3	DTC transfer end
038h	LVD	LVD_LVD1	Voltage monitor 1 interrupt
039h		LVD_LVD2	Voltage monitor 2 interrupt
03Bh	MOSC	MOSC_STOP	Main clock oscillation stop
03Ch	Low power mode	SYSTEM_SNZREQ*2, *3	Snooze entry
040h	AGT0	AGT0_AGTI	AGT interrupt
041h		AGT0_AGTCMAI	Compare match A
042h		AGT0_AGTCMBI	Compare match B
043h	AGT1	AGT1_AGTI	AGT interrupt
044h		AGT1_AGTCMAI	Compare match A
045h		AGT1_AGTCMBI	Compare match B
046h	IWDT	IWDT_NMIUNDF	IWDT underflow
047h	WDT	WDT_NMIUNDF	WDT underflow
049h	RTC	RTC_PRD	Periodic interrupt
04Bh	ADC140	ADC140_ADI	A/D scan end interrupt
04Fh		ADC140_WCMPM*3	Compare match
050h		ADC140_WCMPUM*3	Compare mismatch
057h	ACMPHS	ACMP_HS0	High-speed analog comparator interrupt 0
058h		ACMP_HS1	High-speed analog comparator interrupt 1
05Dh	ACMPLP	ACMP_LP0	Low-power analog comparator interrupt 0
05Eh		ACMP_LP1	Low-power analog comparator interrupt 1

Table 19.3 Association between event signal names set in ELSRn.ELS bits and signal numbers (2 of 5)

Event number	Source of interrupt request generation	Name	Description
063h	IIC0	IIC0_RXI	Receive data full
064h		IIC0_TXI	Transmit data empty
065h		IIC0_TEI	Transmit end
066h		IIC0_EEI	Transfer error
068h	IIC1	IIC1_RXI	Receive data full
069h		IIC1_TXI	Transmit data empty
06Ah		IIC1_TEI	Transmit end
06Bh		IIC1_EEI	Transfer error
06Dh	IIC2	IIC2_RXI	Receive data full
06Eh		IIC2_TXI	Transmit data empty
06Fh		IIC2_TEI	Transmit end
070h		IIC2_EEI	Transfer error
086h	DOC	DOC_DOPCI*3	Data operation circuit interrupt
094h	I/O Port	IOPORT_GROUP1	Port 1 event
095h		IOPORT_GROUP2	Port 2 event
096h		IOPORT_GROUP3	Port 3 event
097h		IOPORT_GROUP4	Port 4 event
098h	ELC	ELC_SWEVT0	Software event 0
099h		ELC_SWEVT1	Software event 1
0B0h	GPT320	GPT0_CCMPA	Compare match A
0B1h		GPT0_CCMPB	Compare match B
0B2h		GPT0_CMPC	Compare match C
0B3h		GPT0_CMPD	Compare match D
0B4h		GPT0_CMPE	Compare match E
0B5h		GPT0_CMPF	Compare match F
0B6h		GPT0_OVF	Overflow
0B7h		GPT0_UDF	Underflow
0BAh	GPT321	GPT1_CCMPA	Compare match A
0BBh		GPT1_CCMPB	Compare match B
0BCh		GPT1_CMPC	Compare match C
0BDh		GPT1_CMPD	Compare match D
0BEh		GPT1_CMPE	Compare match E
0BFh		GPT1_CMPF	Compare match F
0C0h		GPT1_OVF	Overflow
0C1h		GPT1_UDF	Underflow
0C4h	GPT322	GPT2_CCMPA	Compare match A
0C5h		GPT2_CCMPB	Compare match B
0C6h		GPT2_CMPC	Compare match C
0C7h		GPT2_CMPD	Compare match D
0C8h		GPT2_CMPE	Compare match E
0C9h		GPT2_CMPF	Compare match F
0CAh		GPT2_OVF	Overflow
0CBh		GPT2_UDF	Underflow

Table 19.3 Association between event signal names set in ELSRn.ELS bits and signal numbers (3 of 5)

Event number	Source of interrupt request generation	Name	Description
0CEh	GPT323	GPT3_CCMPA	Compare match A
0CFh		GPT3_CCMPB	Compare match B
0D0h		GPT3_CMPC	Compare match C
0D1h		GPT3_CMPD	Compare match D
0D2h		GPT3_CMPE	Compare match E
0D3h		GPT3_CMPF	Compare match F
0D4h		GPT3_OVF	Overflow
0D5h		GPT3_UDF	Underflow
0D8h	GPT324	GPT4_CCMPA	Compare match A
0D9h		GPT4_CCMPB	Compare match B
0DAh		GPT4_CMPC	Compare match C
0DBh		GPT4_CMPD	Compare match D
0DCh		GPT4_CMPE	Compare match E
0DDh		GPT4_CMPF	Compare match F
0DEh		GPT4_OVF	Overflow
0DFh		GPT4_UDF	Underflow
0E2h	GPT325	GPT5_CCMPA	Compare match A
0E3h		GPT5_CCMPB	Compare match B
0E4h		GPT5_CMPC	Compare match C
0E5h		GPT5_CMPD	Compare match D
0E6h		GPT5_CMPE	Compare match E
0E7h		GPT5_CMPF	Compare match F
0E8h		GPT5_OVF	Overflow
0E9h		GPT5_UDF	Underflow
0ECh	GPT326	GPT6_CCMPA	Compare match A
0EDh		GPT6_CCMPB	Compare match B
0EEh		GPT6_CMPC	Compare match C
0EFh		GPT6_CMPD	Compare match D
0F0h		GPT6_CMPE	Compare match E
0F1h		GPT6_CMPF	Compare match F
0F2h		GPT6_OVF	Overflow
0F3h		GPT6_UDF	Underflow
0F6h	GPT327	GPT7_CCMPA	Compare match A
0F7h		GPT7_CCMPB	Compare match B
0F8h		GPT7_CMPC	Compare match C
0F9h		GPT7_CMPD	Compare match D
0FAh		GPT7_CMPE	Compare match E
0FBh		GPT7_CMPF	Compare match F
0FCh		GPT7_OVF	Overflow
0FDh		GPT7_UDF	Underflow

Table 19.3 Association between event signal names set in ELSRn.ELS bits and signal numbers (4 of 5)

Event number	Source of interrupt request generation	Name	Description
100h	GPT328	GPT8_CCMPA	Compare match A
101h		GPT8_CCMPB	Compare match B
102h		GPT8_CMPC	Compare match C
103h		GPT8_CMPD	Compare match D
104h		GPT8_CMPE	Compare match E
105h		GPT8_CMPF	Compare match F
106h		GPT8_OVF	Overflow
107h		GPT8_UDF	Underflow
10Ah	GPT329	GPT9_CCMPA	Compare match A
10Bh		GPT9_CCMPB	Compare match B
10Ch		GPT9_CMPC	Compare match C
10Dh		GPT9_CMPD	Compare match D
10Eh		GPT9_CMPE	Compare match E
10Fh		GPT9_CMPF	Compare match F
110h		GPT9_OVF	Overflow
111h		GPT9_UDF	Underflow
150h	GPT	GPT_UVWEDGE	UVW edge event
174h	SCI0	SCI0_RXI*4	Receive data full
175h		SCI0_TXI*4	Transmit data empty
176h		SCI0_TEI	Transmit end
177h		SCI0_ERI*4	Receive error
178h		SCI0_AM	Address match event
17Ah	SCI1	SCI1_RXI*4	Receive data full
17Bh		SCI1_TXI*4	Transmit data empty
17Ch		SCI1_TEI	Transmit end
17Dh		SCI1_ERI*4	Receive error
17Eh		SCI1_AM	Address match event
180h	SCI2	SCI2_RXI*4	Receive data full
181h		SCI2_TXI*4	Transmit data empty
182h		SCI2_TEI	Transmit end
183h		SCI2_ERI*4	Receive error
184h		SCI2_AM	Address match event
186h	SCI3	SCI3_RXI*4	Receive data full
187h		SCI3_TXI*4	Transmit data empty
188h		SCI3_TEI	Transmit end
189h		SCI3_ERI*4	Receive error
18Ah		SCI3_AM	Address match event
18Ch	SCI4	SCI4_RXI*4	Receive data full
18Dh		SCI4_TXI*4	Transmit data empty
18Eh		SCI4_TEI	Transmit end
18Fh		SCI4_ERI*4	Receive error
190h		SCI4_AM	Address match event

Table 19.3 Association between event signal names set in ELSRn.ELS bits and signal numbers (5 of 5)

Event number	Source of interrupt request generation	Name	Description
1AAh	SCI9	SCI9_RXI*4	Receive data full
1ABh		SCI9_TXI*4	Transmit data empty
1ACh		SCI9_TEI	Transmit end
1ADh		SCI9_ERI*4	Receive error
1AEh		SCI9_AM	Address match event
1BCh	SPI0	SPI0_SPRI	Receive buffer full
1BDh		SPI0_SPTI	Transmit buffer empty
1BEh		SPI0_SPII	Idle
1BFh		SPI0_SPEI	Error
1C0h		SPI0_SPTEND	Transmission completed event
1C1h	SPI1	SPI1_SPRI	Receive buffer full
1C2h		SPI1_SPTI	Transmit buffer empty
1C3h		SPI1_SPII	Idle
1C4h		SPI1_SPEI	Error
1C5h		SPI1_SPTEND	Transmission completed event

Note 1. Only pulse (edge detection) is supported.
Note 2. ELSR8, 9, and ELSR14 to ELSR18 can select this event.
Note 3. This event can occur in Snooze Mode.
Note 4. This event is not supported in FIFO mode.

19.3 Operation

19.3.1 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.

19.3.2 Event Linkage

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. When the ELC activates the module, the operation of the module must be set up in advance. Table 19.4 lists the operations of modules when an event occurs.

Table 19.4 Module operations when event occurs

Module	Operations when event occurs
GPT	• Start counting • Stop counting • Clear counting • Up counting • Down counting • Input capture.
ADC14	Starts A/D conversion
DAC12	Starts D/A 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 PORT 3 PORT 4
CTSU	Starts measurement operation
DTC	Starts DTC data transfer

19.3.3 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 register 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 000000000b in the ELSRn.ELS[8:0] bits associated with the modules. To stop linkage of all events, set the ELCR.ELCON bit to 0.

If the event link output from the RTC is to be used, set the ELC after the RTC, for example, initialization and time setting. Unintended events can be generated if the RTC settings are made after the ELC settings.

19.4 Usage Notes

19.4.1 Linking DMAC/DTC Transfer End Signals as Events

When linking the DMAC or DTC transfer end signals as events, do not set the same peripheral module as the DMAC or DTC transfer destination and event link destination. If set, the peripheral module might be started before DMAC or DTC transfer to the peripheral module is complete.

19.4.2 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 modes in which the module is stopped (Software Standby Mode). Some modules can perform in Snooze mode. For more information, see Table 19.3 and section 11, Low Power Modes.

19.4.3 Module Stop Function Setting

The Module Stop Control Register C (MSTPCRC) can enable or disable ELC operation. After a reset, the ELC is disabled. Releasing the module-stop state enables access to the registers. For more information, see Table 19.3 and section 11, Low Power Modes.

The ELCON bit must be set to 0 before disabling ELC operation using the Module Stop Control Register.

19.4.4 ELC Delay Time

As shown in Figure 19.2, module A accesses the module B through ELC. There is a delay time in the ELC module between module A and module B, called the ELC delay time. The ELC delay time is shown in Table 19.5.

If the clock domains on both module A and B are the same, the delay time is 0. But, if the clock domains on modules A and B are different, ELC module has some delay. The time delay is defined by the slower clock frequency among module A and module B clocks.

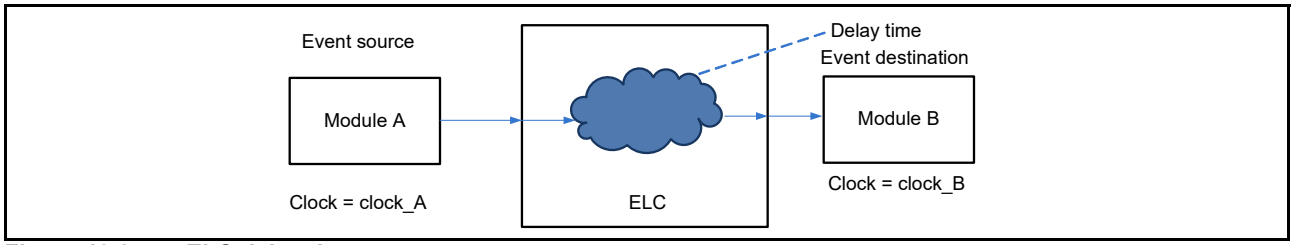


Figure 19.2 ELC delay time

Table 19.5 ELC delay time

Clock domain	Clock frequency	ELC delay time
Clock_A = Clock_B	Clock_A = Clock_B	0 cycles
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

20. I/O Ports

20.1 Overview

The pins of the I/O Ports operate as general I/O port pins, I/O pins for peripheral modules, interrupt input pins, analog I/O, port group function for ELC, or bus control pins. All pins operate as input pins immediately after a reset, and pin functions are switched by register settings. You can specify the associated I/O ports and peripheral modules for each pin in the registers. Figure 20.1 shows a connection diagram for the I/O Ports registers.

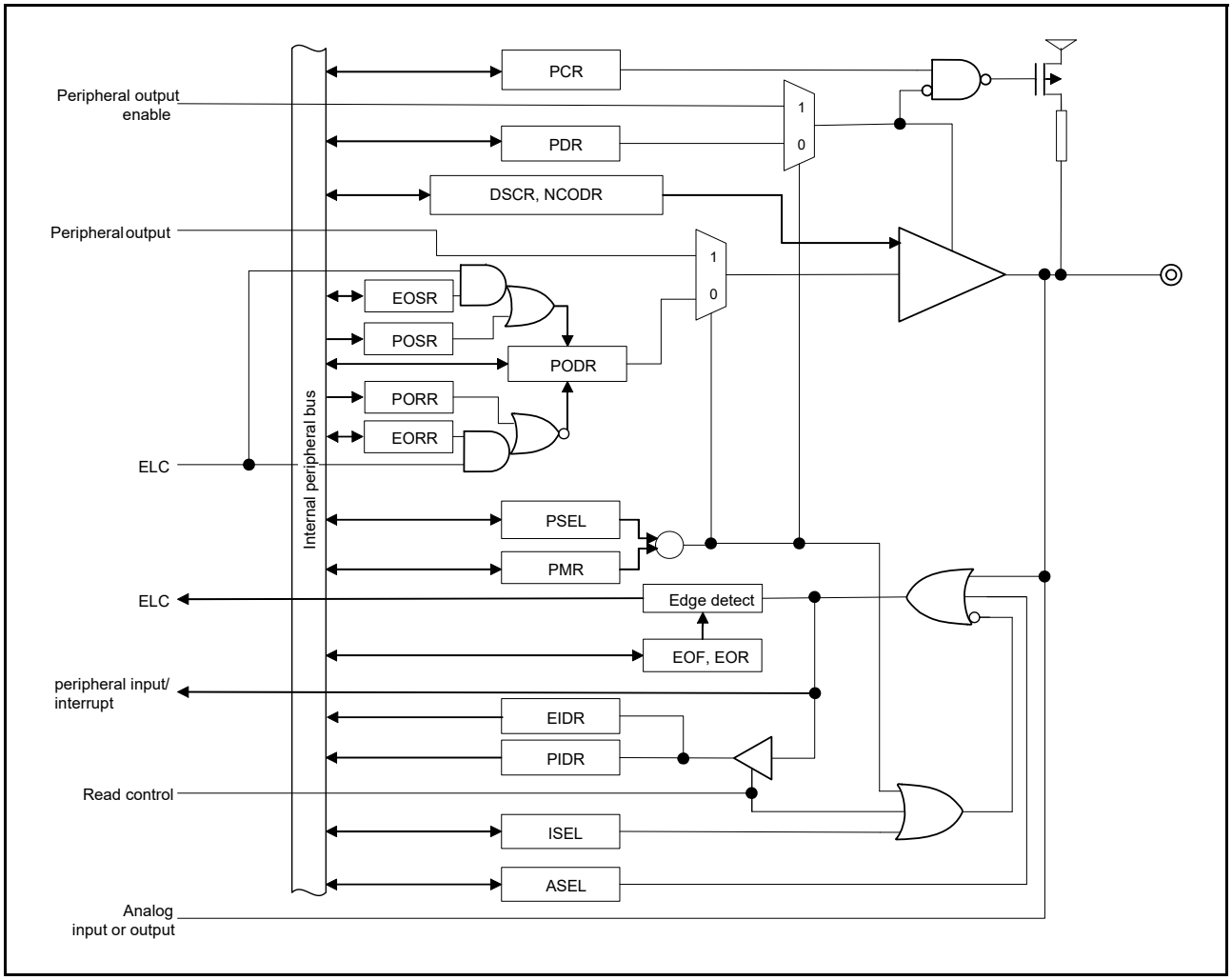


Figure 20.1 I/O Ports registers connection diagram

Note: This figure shows a basic port configuration. The configuration differs depending on the ports.

The configuration of the I/O ports differs according to the package. Table 20.1 shows the I/O Ports specifications, and Table 20.2 lists the port functions.

Table 20.1 I/O Ports specifications

Port	Package		Package		Package		Package	
	144 pins, 145 pins	No. of pins	121 pins	No. of pins	100 pins	No. of pins	64 pins	No. of pins
PORT0	P000 to P015	16	P000 to P008, P010 to P015	15	P000 to P008, P010 to P015	15	P000 to P004, P010 to P015	11
PORT1	P100 to P115	16	P100 to P115	16	P100 to P115	16	P100 to P113	14
PORT2	P200 to P206, P212 to P215	11	P200 to P206, P212 to P215	11	P200 to P206, P212 to P215	11	P200, P201, P204 to P206, P212 to P215	9
PORT3	P300 to P315	16	P300 to P309, P313 to P315	13	P300 to P307	8	P300 to P304	5
PORT4	P400 to P415	16	P400 to P415	16	P400 to P415	16	P400 to P402, P407 to P411	8
PORT5	P500 to P507, P511, P512	10	P500 to P506, P511, P512	9	P500 to P505	6	P500 to P502	3
PORT6	P600 to P606, P608 to P614	14	P600 to P605, P608 to P613	12	P600 to P603, P608 to P610	7	N/A	0
PORT7	P700 to P705, P708 to P713	12	P700 to P702, P708 to P710	6	P708	1	N/A	0
PORT8	P800 to P809	10	P800, P801, P808, P809	4	P808, P809	2	N/A	0
PORT9	P900 to P902	3	N/A	0	N/A	0	N/A	0
	Total of pins	124	Total of pins	102	Total of pins	82	Total of pins	50

Table 20.2 I/O Port functions

Port	Port name	Input pull-up	Open-drain output	Drive capacity switching	5-V tolerant
PORT0	P000 to P015	✓	-	Low/Middle	-
PORT1	P100 to P115	✓	✓	Low/Middle	-
PORT2	P200, P214, P215	-	-	-	-
	P201 to P204	✓	✓	Low/Middle	-
	P205, P206	✓	✓	Low/Middle	✓
	P212, P213	✓	✓	-	-
PORT3	P300 to P315	✓	✓	Low/Middle	-
PORT4	P400 to P404, P407	✓	✓	Low/Middle	✓
	P405, P406, P408 to P415	✓	✓	Low/Middle	-
PORT5	P500 to P507	✓	✓	Low/Middle	-
	P511, P512	✓	✓	Low/Middle	✓
PORT6	P600 to P606, P608 to P614	✓	✓	Low/Middle	-
PORT7	P700 to P705, P708 to P713	✓	✓	Low/Middle	-
PORT8	P800 to P809	✓	✓	Low/Middle	-
PORT9	P900 to P902	✓	✓	Low/Middle	-

✓: Available

20.2 Register Descriptions

20.2.1 Port Control Register 1 (PCNTR1/PODR/PDR)

Address(es): PORT0.PCNTR1 4004 0000h, PORT1.PCNTR1 4004 0020h, PORT2.PCNTR1 4004 0040h, PORT3.PCNTR1 4004 0060h, PORT4.PCNTR1 4004 0080h, PORT5.PCNTR1 4004 00A0h, PORT6.PCNTR1 4004 00C0h, PORT7.PCNTR1 4004 00E0h, PORT8.PCNTR1 4004 0100h, PORT9.PCNTR1 4004 0120h

PORT0.PODR 4004 0000h, PORT1.PODR 4004 0020h, PORT2.PODR 4004 0040h, PORT3.PODR 4004 0060h, PORT4.PODR 4004 0080h, PORT5.PODR 4004 00A0h, PORT6.PODR 4004 00C0h, PORT7.PODR 4004 00E0h, PORT8.PODR 4004 0100h, PORT9.PODR 4004 0120h

PORT0.PDR 4004 0002h, PORT1.PDR 4004 0022h, PORT2.PDR 4004 0042h, PORT3.PDR 4004 0062h, PORT4.PDR 4004 0082h, PORT5.PDR 4004 00A2h, PORT6.PDR 4004 00C2h, PORT7.PDR 4004 00E2h, PORT8.PDR 4004 0102h, PORT9.PDR 4004 0122h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PODR 15	PODR 14	PODR 13	PODR 12	PODR 11	PODR 10	PODR 09	PODR 08	PODR 07	PODR 06	PODR 05	PODR 04	PODR 03	PODR 02	PODR 01	PODR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit name	Description	R/W
b15 to b0	PDRn	Pmn Direction	0: Input (functions as an input pin) 1: Output (functions as an output pin).	R/W
b31 to b16	PODRn	Pmn Output Data	0: Low output 1: High output.	R/W

m = 0 to 9
n = 00 to 15

Port Control Register 1 is 32- or 16-bit readable/writable register that controls the port direction and port output data. PCNTR1 specifies the port direction and the output data, which is accessed in 32-bit units. PODR (bits [31:16]) and PDR (bits [15:0]) are accessed in 16-bit units.

PDRn selects 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. The bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0. P200, P214, and P215 are input only, so PORT2.PCNTR1.PDR00, PORT2.PCNTR1.PDR14, and PORT2.PCNTR1.PDR15 are reserved. The PDRn bit in the PORTm.PCNTR1 register serves the same function as the PDR bit in the PFS.PmnPFS register.

PODRn is a bit that holds data to be output from the pins used for general I/O. The bits of non-existent port m are reserved. Write 0 to these bits. The bit of a non-existent pin is reserved. P200, P214, P215 is input only, so PORT2.PCNTR1.PODR00, PORT2.PCNTR1.PODR14, and PORT2.PCNTR1.PODR15 bits are reserved. A reserved bit is read as 0. The write value should be 0. The PODRn bit in the PORTm.PCNTR1 register serves the same function as the PODR bit in the PFS.PmnPFS register.

20.2.2 Port Control Register 2 (PCNTR2/EIDR/PIDR)

Address(es): PORT0.PCNTR2 4004 0004h, PORT1.PCNTR2 4004 0024h, PORT2.PCNTR2 4004 0044h, PORT3.PCNTR2 4004 0064h, PORT4.PCNTR2 4004 0084h, PORT5.PCNTR2 4004 00A4h, PORT6.PCNTR2 4004 00C4h, PORT7.PCNTR2 4004 00E4h, PORT8.PCNTR2 4004 0104h, PORT9.PCNTR2 4004 0124h

PORT1.EIDR 4004 0024h, PORT2.EIDR 4004 0044h, PORT3.EIDR 4004 0064h, PORT4.EIDR 4004 0084h

PORT0.PIDR 4004 0006h, PORT1.PIDR 4004 0026h, PORT2.PIDR 4004 0046h, PORT3.PIDR 4004 0066h, PORT4.PIDR 4004 0086h, PORT5.PIDR 4004 00A6h, PORT6.PIDR 4004 00C6h, PORT7.PIDR 4004 00E6h, PORT8.PIDR 4004 0106h, PORT9.PIDR 4004 0126h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b15 to b0	PIDRn	Pmn State	0: Low level 1: High level.	R
b31 to b16	EIDRn	Port Event Input Data*1	When the ELC_PORTx occurs: 0: Low input 1: High input.	R

m = 0 to 9
n = 00 to 15
x = 1 to 4
Note 1. Supported for PORT1 to PORT4.

Port Control Register 2 allows read access to the Pmn state and the port event input data by 32- and 16-bit accesses. The PCNTR2 register specifies the Pmn state and the port event input data, which is accessed in 32-bit units. EIDRn (bits 31 to 16 in PCNTR2) and PIDRn (bits 15 to 0 in PCNTR2) respectively are accessed in 16-bit units. Bits associated with non-existent pins are reserved. Reserved bits are read as undefined.

PIDRn reflects 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)
- Sub-clock oscillator (SOSC)
- CS area controller (CSC)
- Analog function (ASEL = 1)
- Capacitive Touch Sensing Unit (CTSU)
- Segment LCD Controller (SLCDC)
- USC 2.0 Full-Speed Module (USBFS).

EIDRn latches a pin state when an ELC_PORTx signal occurs. Pin states can only be input to EIDRn when PmnPFS.PMR = 0 and PORTm.PCNTR1.PDRn = 0.

When PmnPFS.ASEL is set to 1, the associated pin state is not reflected in EIDRn.

20.2.3 Port Control Register 3 (PCNTR3/PORR/POSR)

Address(es): PORT0.PCNTR3 4004 0008h, PORT1.PCNTR3 4004 0028h, PORT2.PCNTR3 4004 0048h, PORT3.PCNTR3 4004 0068h, PORT4.PCNTR3 4004 0088h, PORT5.PCNTR3 4004 00A8h, PORT6.PCNTR3 4004 00C8h, PORT7.PCNTR3 4004 00E8h, PORT8.PCNTR3 4004 0108h, PORT9.PCNTR3 4004 0128h

PORT0.PORR 4004 0008h, PORT1.PORR 4004 0028h, PORT2.PORR 4004 0048h, PORT3.PORR 4004 0068h, PORT4.PORR 4004 0088h, PORT5.PORR 4004 00A8h, PORT6.PORR 4004 00C8h, PORT7.PORR 4004 00E8h, PORT8.PORR 4004 0108h, PORT9.PORR 4004 0128h

PORT0.POSR 4004 000Ah, PORT1.POSR 4004 002Ah, PORT2.POSR 4004 004Ah, PORT3.POSR 4004 006Ah, PORT4.POSR 4004 008Ah, PORT5.POSR 4004 00AAh, PORT6.POSR 4004 00CAh, PORT7.POSR 4004 00EAh, PORT8.POSR 4004 010Ah, PORT9.POSR 4004 012Ah

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit name	Description	R/W
b15 to b0	POSRn	Pmn Output Set	0: No affect to output 1: High output.	W
b31 to b16	PORRn	Pmn Output Reset	0: No affect to output 1: Low output.	W

m = 0 to 9
n = 00 to 15

Note: When EORRn or EOSRn is set, writing to PODRn, PORRn, and POSRn is prohibited.
Note: PORRn and POSRn should not be set at the same time.

Port Control Register 3 is a 32- and 16-bit writable register that controls the setting or resetting of the port output data. PCNTR3 controls the setting and resetting of the output data, which is set by 32-bit units. PORR (bits 31 to 16 in PCNTR3) and POSR (bits 15 to 0 in PCNTR3) respectively, are accessed in 16-bit units.

POSR changes PODR when set by a software write. For example, for P100, when PORT1.POSR00 = 1, PORT1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value must always be 0. P200, P214, and P215 are input only, so PORT2.PCNTR3.POSR00, PORT2.PCNTR3.POSR14, and PORT2.PCNTR3.POSR15 are reserved.

PORR changes PODR when reset by a software write. For example, for P100, when PORT1.PORR00 = 1, PORT1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. P200, P214, and P215 are input only, so PORT2.PCNTR3.PORR00, PORT2.PCNTR3.PORR14, and PORT2.PCNTR3.PORR15 are reserved.

20.2.4 Port Control Register 4 (PCNTR4/EORR/EOSR)

Address(es): PORT1.PCNTR4 4004 002Ch, PORT2.PCNTR4 4004 004Ch, PORT3.PCNTR4 4004 006Ch, PORT4.PCNTR4 4004 008Ch, PORT1.EORR 4004 002Ch, PORT2.EORR 4004 004Ch, PORT3.EORR 4004 006Ch, PORT4.EORR 4004 008Ch, PORT1.EOSR 4004 002Eh, PORT2.EOSR 4004 004Eh, PORT3.EOSR 4004 006Eh, PORT4.EOSR 4004 008Eh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit name	Description	R/W
b15 to b0	EOSRn	Pmn Event Output Set	When the ELC_PORTx occurs: 0: No affect to output 1: High output.	R/W
b31 to b16	EORRn	Pmn Event Output Reset	When the ELC_PORTx occurs: 0: No affect to output 1: Low output.	R/W

m = 1 to 4
n = 00 to 15
x = 1 to 4

Note: When EORRn or EOSRn is set, writing to PODRn, PORRn, and POSRn is prohibited.
Note: EORRn and EOSRn should not be set at the same time.

Port Control Register 4 is a 32- or 16-bit readable/ writable register that controls the setting or resetting of the port output data by event input from the ELC. PCNTR4 is accessed in 32-units, while EORR (bits [31:16] in PCNTR4) and EOSR (bits [15:0] in PCNTR4) are accessed in 16-bit units.

EOSR changes PODR when set because an ELC_PORTx signal occurs. For example, for P100, if PORT1.EOSR00 is set to 1 when ELC_PORTx occurs, PORT1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value must always be 0. P200, P214, and P215 are input only, so PORT2.PCNTR4.EOSR00, PORT2.PCNTR4.EOSR14, and PORT2.PCNTR4.EOSR15 are reserved.

EORR changes PODR when reset because an ELC_PORTx signal occurs. For example, for P100 if PORT1.EORR00 is set to 1 when ELC_PORTx occurs, PORT1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value must always be 0. P200, P214, and P215 are input only, so PORT2.PCNTR4.EORR00, PORT2.PCNTR4.EORR14, and PORT2.PCNTR4.EORR15 bits are reserved.

20.2.5 Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) (m = 0 to 9; n = 00 to 15)

Address(es): PFS.P000PFS 4004 0800h to PFS.P015PFS 4004 083Ch, PFS.P100PFS 4004 0840h to PFS.P115PFS 4004 087Ch, PFS.P200PFS 4004 0880h to PFS.P206PFS 4004 0898h, PFS.P212PFS 4004 08B0h to PFS.P215PFS 4004 08BCh, PFS.P300PFS 4004 08C0h to PFS.P315PFS 4004 08FCh, PFS.P400PFS 4004 0900h to PFS.P415PFS 4004 093Ch, PFS.P500PFS 4004 0940h to PFS.P507PFS 4004 095Ch, PFS.P511PFS 4004 096Ch, PFS.P512PFS 4004 0970h, PFS.P600PFS 4004 0980h to PFS.P606PFS 4004 0998h, PFS.P608PFS 4004 09A0h to PFS.P614PFS 4004 09B8h, PFS.P700PFS 4004 09C0h to PFS.P705PFS 4004 09D4h, PFS.P708PFS 4004 09E0h to PFS.P713PFS 4004 09F4h, PFS.P800PFS 4004 0A00h to PFS.P809PFS 4004 0A24h, PFS.P900PFS 4004 0A40h to PFS.P902PFS 4004 0A48h
PFS.P000PFS_HA 4004 0802h to PFS.P015PFS_HA 4004 083Eh, PFS.P100PFS_HA 4004 0842h to PFS.P115PFS_HA 4004 087Eh, PFS.P200PFS_HA 4004 0882h to PFS.P215PFS_HA 4004 08BEh, PFS.P300PFS_HA 4004 08C2h to PFS.P315PFS_HA 4004 08FEh, PFS.P400PFS_HA 4004 0902h to PFS.P415PFS_HA 4004 093Eh, PFS.P500PFS_HA 4004 0942h to PFS.P515PFS_HA 4004 097Eh, PFS.P600PFS_HA 4004 0982h to PFS.P615PFS_HA 4004 09BEh, PFS.P700PFS_HA 4004 09C2h to PFS.P715PFS_HA 4004 09FEh, PFS.P800PFS_HA 4004 0A02h to PFS.P815PFS_HA 4004 0A3Eh, PFS.P900PFS_HA 4004 0A42h to PFS.P915PFS_HA 4004 0A7Eh, PFS.P000PFS_BY 4004 0803h to PFS.P015PFS_BY 4004 083Fh, PFS.P100PFS_BY 4004 0843h to PFS.P115PFS_BY 4004 087Fh, PFS.P200PFS_BY 4004 0883h to PFS.P215PFS_BY 4004 08BFh, PFS.P300PFS_BY 4004 08C3h to PFS.P315PFS_BY 4004 08FFh, PFS.P400PFS_BY 4004 0903h to PFS.P415PFS_BY 4004 093Fh, PFS.P500PFS_BY 4004 0943h to PFS.P515PFS_BY 4004 097Fh, PFS.P600PFS_BY 4004 0983h to PFS.P615PFS_BY 4004 09BFh, PFS.P700PFS_BY 4004 09C3h to PFS.P715PFS_BY 4004 09FFh, PFS.P800PFS_BY 4004 0A03h to PFS.P815PFS_BY 4004 0A3Fh, PFS.P900PFS_BY 4004 0A43h to PFS.P915PFS_BY 4004 0A7Fh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	PSEL[4:0]					—	—	—	—	—	—	—	PMR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*2

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	EOFR[1:0]		—	DSCR	—	—	—	NCODR	—	PCR	—	PDR	PIDR	PODR
Value after reset:	0	0	0	0	0	0*2	0	0	0	0	0	0*2	0	0	x	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	PODR	Port Output Data	0: Low output 1: High output.	R/W
b1	PIDR	Pmn State	0: Low level 1: High level.	R
b2	PDR	Port Direction	0: Input (functions as an input pin) 1: Output (functions as an output pin).	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	PCR	Pull-up Control	0: Disables an input pull-up 1: Enables an input pull-up	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	NCODR	N-Channel Open Drain Control	0: CMOS output 1: NMOS open-drain output.	R/W
b9 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10	DSCR	Port Drive Capability	0: Low drive 1: Middle drive.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13, b12	EOFR[1:0]	Event on Falling/Event on Rising*1	b13 b12 0 0: Don't care 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edge.	R/W
b14	ISEL	IRQ Input Enable	0: Not used as an IRQn input pin 1: Used as an IRQn input pin.	R/W
b15	ASEL	Analog Input Enable	0: Not used as an analog pin 1: Used as an analog pin.	R/W
b16	PMR	Port Mode Control	0: Used as a general I/O pin. 1: Used as an I/O port for peripheral functions.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28 to b24	PSEL[4:0]	Peripheral Select	These bits select the peripheral function. For individual pin functions, see the associated tables in this chapter.	R/W

Bit	Symbol	Bit name	Description	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Supported for PORT1 to PORT4.

Note 2. The initial value of P108, P109, P110, P201 and P300 is not 0000 0000h.
P108 is 0001 0010h, P109 is 0001 0000h, P110 is 0001 0010h, P201 is 0000 0010h, and P300 is 0001 0010h. The Port mn Pin Function Select Register (PmnPFS) selects the pin function.

Port mn Pin Function Select Register is a 32-, 16-, or 8-bit readable/writable control register that controls the selection of the port mn function. PmnPFS is set in 32-bit units, PmnPFS_HA (bits [15:0 in PmnPFS) is accessed in 16-bit units, and PmnPFS_BY (bits [7:0]) is accessed in 8-bit units.

The PDR/PIDR/PODR bits serve the same function as the PCNTR. When these bits are read, the PCNTR value is read.

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 an external bus pin, 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.

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.

The DSCR bit switches the drive capacity of the port. If the drive capacity of a pin is fixed, the associated bit is readable and writable, but the drive capacity cannot be changed. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

The EOR and EOF bits select the edge detection method for the port group input signal. These bits support rising, falling, or both edge detections. When the EOR/EOF 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 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.

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.

The ASEL bit specifies analog pins. When a pin is set to 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 with the Pull-up Control bit (PmnPFS.PCR).
3. Specify 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 input/output is reserved.

The PMR bit specifies the port pin function. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

The PSEL[4:0] bits assign the peripheral function.

For details of the peripheral settings for each product, see section 20.6, Peripheral Select Settings for each product.

20.2.6 Write-Protect Register (PWPR)

Address(es): PMISC.PWPR 4004 0D03h

	b7	b6	b5	b4	b3	b2	b1	b0
	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PFSWE	PmnPFS Register Write Enable	0: Writing to the PmnPFS register is disabled 1: Writing to the PmnPFS register is enabled.	R/W
b7	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.

20.3 Operation

20.3.1 General I/O Ports

All pins except P108, P109, P110, and P300 operate as general I/O input 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 pin with the Pin Function Select Registers. For details on these registers, see section 20.2, Register Descriptions.

Each port has the following bits:

- Port Direction bit (PDR), which selects input or output direction
- Port Output Data bit (PODR), which holds data for output
- Port Input Data bit (PIDR), which indicates the pin states
- Event Input Data bit (EIDR), which indicates the pin state when an ELC_PORTx (x = 1, 2, 3, or 4) signal occurs
- Port Output Set bit (POSR), which indicates the output value when a software write occurs
- Port Output Reset bit (PORR), which indicates the output value when a software write occurs
- Event Output Set bit (EOSR), which indicates the output value when an ELC_PORTx (x = 1, 2, 3, or 4) signal occurs
- Event Output Reset bit (EORR), which indicates the output value when the ELC_PORTx (x = 1, 2, 3, or 4) signal occurs.

20.3.2 Port Function Select

The following port functions are available for configuring each pin:

- I/O configuration — Complementary or 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 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
- DSCR: Drive capacity control bits that select the drive capacity
- EOR: Event on rising bit used to detect rising edges on the port input
- EOF: Event on falling bit used to detect falling edges on the port input
- 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: Port function select bits to select the associated peripheral function.

These configurations can be made by a single-register access to the Pin Function Select Register. For details, see section 20, Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) (m = 0 to 9; n = 00 to 15).

20.3.3 Port Group Function for ELC

In the MCU, PORT1 to PORT4 are assigned for the port group function.

20.3.3.1 Behavior When ELC_PORTx (x = 1, 2, 3, or 4) is Input from ELC

The MCU supports two functions when an ELC_PORTx (x = 1, 2, 3, or 4) 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_PORTx (x = 1, 2, 3, or 4) signal comes from the ELC, the input enable of the I/O cell is asserted, and then data from the external pins are read into the EIDR bit.

For the GPO function (PDR = 1) or the peripheral mode (PMR = 1), 0 is input into the EIDR bit from the external pins.

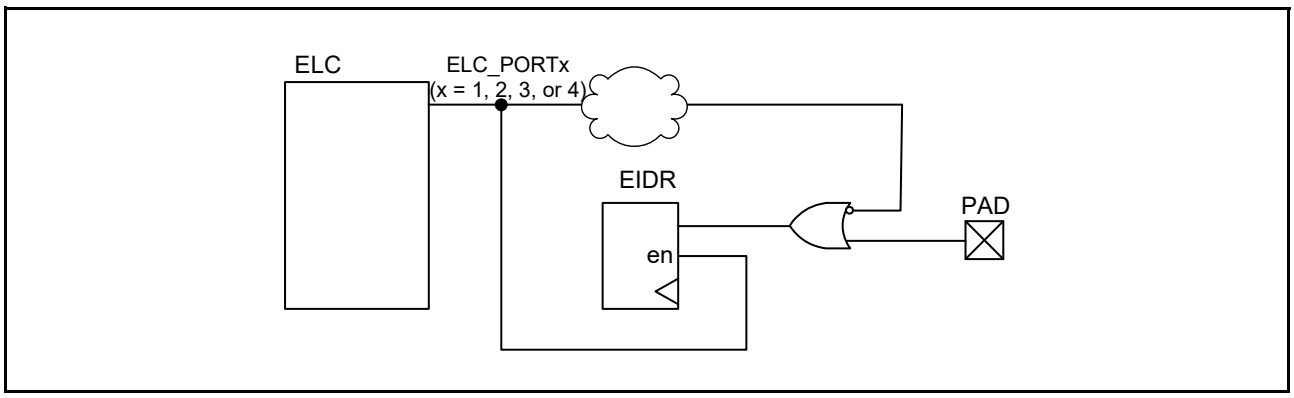


Figure 20.2 Event ports input data

(2) Output from PODR by EOSR/EORR

When an ELC_PORTx (x = 1, 2, 3, or 4) signal occurs, the data is output from the PODR to the external pin based on the setting in the EOSR/EORR bits.

- If EOSR is set to 1, when an ELC_PORTx (x = 1, 2, 3, or 4) signal occurs, the PODR register outputs 1 to the external pin.
  Otherwise, when EOSR = 0, the PODR value is kept.
- If 1 is set for the EORR register, when the ELC_PORTx (x = 1, 2, 3, or 4) occurs, the PODR register outputs 0 to the external pin.
  Otherwise, when EORR = 0, the PODR value is kept.

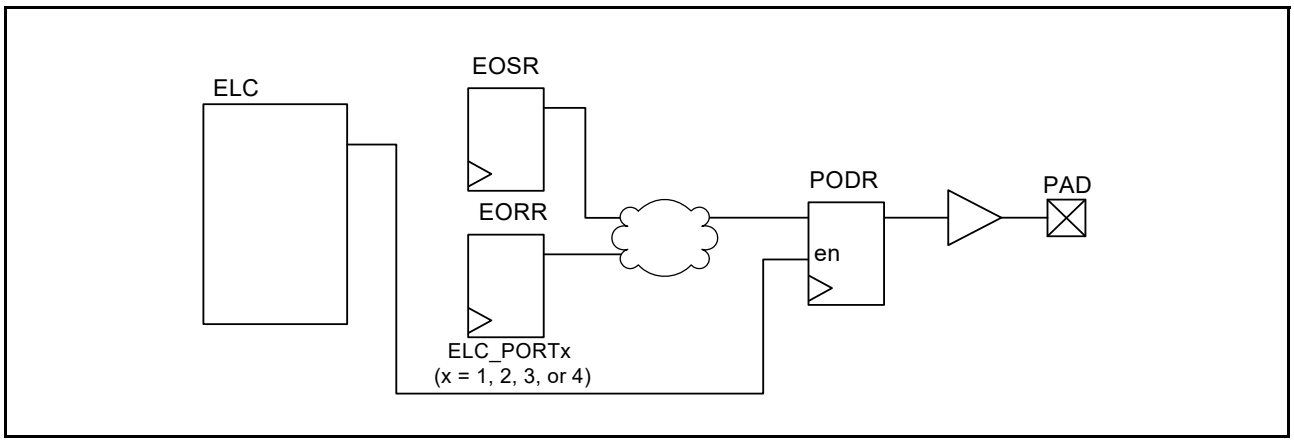


Figure 20.3 Event ports output data

20.3.3.2 Behavior when Event Pulse is Output to ELC

To output the event pulse from the external pins to the ELC, set the EOR/EOF bit in the PmnPFS register. For details, see section 20.2.5, Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) (m = 0 to 9; n = 00 to 15). When the EOR/EOF bits are set, the input enable of the I/O cell is asserted.

Data of the external pin is the input. For example, for PORT1 when the data is input from P100 to P115, the data of those 16 pins is organized by OR logic. This data is formed into a one-shot pulse that goes to the ELC. The operation of PORT2 to PORT4 is the same.

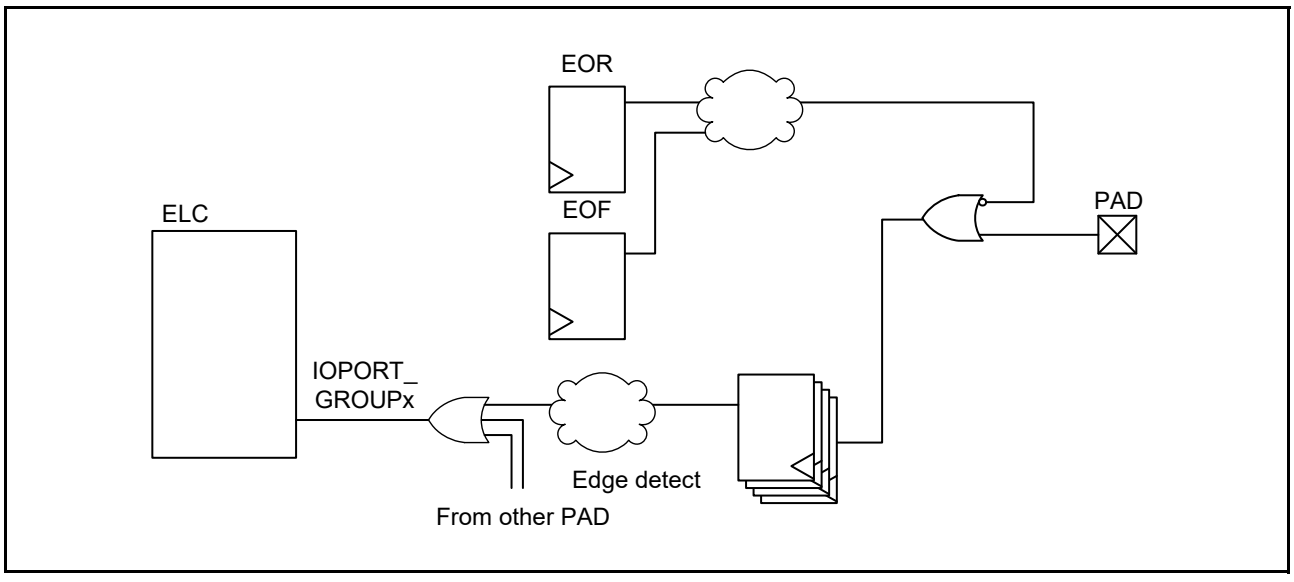


Figure 20.4 Generation of event pulse

20.4 Handling of Unused Pins

Table 20.3, Handling of unused pins shows how to handle unused pins.

Table 20.3 Handling of unused pins

Pin Name	Description
P201/MD	Use this as a mode pin
RES	Connect this pin to VCC through a resistor (pulling up)
USB_DP	Keep these pins open
USB_DM	Keep these pins open
P200/NMI	Connect this pin 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, it is configured in the same way as port 1 to 9.
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 port 1 to 9. When the external clock is input to the EXTAL pin, leave this pin open.
P215/XCIN	When the Sub-clock oscillator is not used, set the SOSCCR.SOSTP bit to 1 (general port P215). When this pin is not used as port P215, it is configured in the same way as port 1 to 9.
P214/XCOUT	When the Sub-clock oscillator is not used, set the SOSCCR.SOSTP bit to 1 (general port P214). When this pin is not used as port P214, it is configured in the same way as port 1 to 9.
P000 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.
P1x to P9x other than P200, P201 and P212 to P215	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), the pin is released.*1 *4

Note 1. Clear the PmnPFS.PMR bit, the PmnPFS.ISEL bit, PmnPFS.PCR, and the PmnPFS.ASEL bit to 0.

Note 2. P108, P110, P300 are recommended for pull up VCC (pulled up) through a resistor, because these pins are input pull-up enabled from initial value (PmnPFS.PCR = 1).

Note 3. For details, see section 51, Internal Voltage Regulator.

Note 4. P109 is recommended to set the direction to output (PCNTR1.PDRn = 1), because this pin is output from initial value.

20.5 Usage Notes

20.5.1 Procedure for Specifying the Pin Functions

To specify the input/output 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 in the PMR for the target pin to select the general I/O port.
4. Specify the input/output function for the pin through the PSEL[4:0] bit settings in the PmnPFS register.
5. Set the PMR to 1 as necessary to switch to the selected input/output 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.

20.5.2 Procedure for Using Port Group Input

To use the port group input (PORT1 to PORT4):

1. Set the ELSRx.ELS[8:0] bits to 000000000b to ignore the unexpected pulse. For more information, see section 19, Event Link Controller (ELC).
2. Set the EOF/EOR bit 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 the unexpected pulses depends on the initial value of the external pin.
4. Set the ELSRx.ELS[8:0] bits to enable the event signals.

20.5.3 Port Output Data Register (PODR) Summary

This register outputs data as follows:

1. Output 0 if PCNTR4.EORR is set to 1 when an ELC_PORTx (x = 1, 2, 3, or 4) signal occurs.
2. Output 1 if the PCNTR4.EOSR is set to 1 when the ELC_PORTx (x = 1, 2, 3, or 4) occurs from the ELC.
3. Output 0 if PCNTR3.PORR is set to 1.
4. Output 1 if PCNTR3.POSR is set to 1.
5. Output 0 or 1 because PCNTR1.PODR is set.
6. Output 0 or 1 because PmnPFS.PODR is set.

Numbers in this list correspond to the priority for writing to the PODR. For example, if 1. and number 3. from the list occur at same time, the higher priority number 1. is executed.

20.5.4 Notes on Using of Analog Functions

To use an analog function, set the associated bits in both the Port Mode Control bit (PMR) and Port Direction bit (PDR) to 0 so that the pin acts as a general input port. Next, set the Analog Input Enable bit in the Port mn Pin Function Select Register (PmnPFS.ASEL) to 1.

20.5.5 I/O Buffer Specification

In case the P402, P403, and P404 pins are configured as outputs or inputs with the internal pull-up resistor, set the VBTCR1.BPWSWSTP bit to 1 before setting the I/O registers regardless of whether or not the battery backup function is used. This setting is needed only one time after a power-on reset. And clear the VBTCR1.BPWSWSTP bit to 0 again after setting registers associated with the battery backup function when using the battery backup function.

The setting flow of the VBTCR1.BPWSWSTP bit is shown in Figure 12.2.

The P402, P403, and P404 pins can be used as the RTC input pins RTCICn, where n = 0 to 2. When these input pins are enabled by the VBTICTLR register, the output function of these pins is forced to disable. Therefore, the VBTICTLR register must be set to 0 to use the port function.

Note: The VBTICTLR register is not initialized on reset. For more information, see section 12, Battery Backup Function.

20.6 Peripheral Select Settings for each product

This section provides the detail of the Pin Function Select configuration by the PmnPFS register. Several pin names have added _A, _B, and _C suffixes. When assigning IIC, SPI, and SSI functionality, select the functional pins of the same suffix. The other pins can be selected regardless of the suffix. Assigning the same function to two or more pins simultaneously is prohibited.

Table 20.4 Register settings for input/output pin function (PORT0)

PSEL[4:0] bits settings	Function	Pin P000	P001	P002	P003	P004	P005	P006	P007
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
01100b	CTSU	TS21	TS22	-	-	-	TS26	TS27	-
ASEL bit		AN000/ AMP0+/ IVREF0/ IVCMP0	AN001/ AMP0-/ IVREF1/ IVCMP1	AN002/ AMP0O/ IVREF2/ IVCMP2	AN003/ AMP1O/ IVREF3/ IVCMP3	AN004/ AMP2O/ IVCMP0	AN005/ AMP3+/ IVREF0	AN006/ AMP3-/ IVREF4/ IVREF1	AN007/ AMP3O/ IVCMP4/ IVCMP1
ISEL bit		IRQ6	IRQ7	IRQ8	—	IRQ9	IRQ10	IRQ11	—
PCR bit		✓	✓	✓	✓	✓	✓	✓	✓
DSCR bit		L/M	L/M	L/M	L/M	L/M	L/M	L/M	L/M
145-pin product, 144-pin product		✓	✓	✓	✓	✓	✓	✓	✓
121-pin product		✓	✓	✓	✓	✓	✓	✓	✓
100-pin product		✓	✓	✓	✓	✓	✓	✓	✓
64-pin product		✓	✓	✓	✓	✓			

PSEL[4:0] bits settings	Function	Pin P008	P009	P010	P011	P012	P013	P014	P015
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
01100b	CTSU	TS29	—	TS30	TS31	—	—	—	—
ASEL bit		AN008	AN009	AN010/ AMP2-/ VREFH0	AN011/ AMP2+/ VREFL0	AN012/ AMP1-/ VREFH	AN013/ AMP1+/ VREFL	AN014/ IVREF5/ IVREF2/ DA0	AN015/ IVCMP5/ IVCMP2/ DA1
ISEL bit		IRQ12	IRQ13	IRQ14	IRQ15	—	—	—	IRQ13
PCR bit		✓	✓	✓	✓	✓	✓	✓	✓
DSCR bit		L/M	L/M	L/M	L/M	L/M	L/M	L/M	L/M
145-pin product, 144-pin product		✓	✓	✓	✓	✓	✓	✓	✓
121-pin product		✓		✓	✓	✓	✓	✓	✓
100-pin product		✓		✓	✓	✓	✓	✓	✓
64-pin product				✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Table 20.5 Register settings for input/output pin function (PORT1) (1)

PSEL[4:0] bits settings	Function	Pin P100	P101	P102	P103	P104	P105	P106	P107
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
00001b	AGT	AGTIO0_A	AGTEE0	AGTO0	—	—	—	—	—
00010b	GPT	GTETRGA_A	GTETRGB_A	GTOWLO_A	GTOWUP_A	GTETRGB_B	GTETRGA_C	—	—
00011b	GPT	—	—	GTIOC2B_A	GTIOC2A_A	—	—	GTIOC8B_A	GTIOC8A_A
00100b	SCI	RXD0_A/ MISO0_A/ SCL0_A	TXD0_A/ MOSI0_A/ SDA0_A	SCK0_A	CTS0_RTS0_A /SS0_A	—	—	—	—
00101b	SCI	SCK1_A	CTS1_RTS1_A /SS1_A	—	—	—	—	—	—
00110b	SPI	MISOA_A	MOSIA_A	RSPCKA_A	SSLA0_A	SSLA1_A	SSLA2_A	SSLA3_A	—
00111b	IIC	SCL1_B	SDA1_B	—	—	—	—	—	—
01000b	KINT	KR00	KR01	KR02	KR03	KR04	KR05	KR06	KR07
01001b	CLKOUT/ ACMPHS/ ACMPLP/RTC	—	—	—	—	—	—	—	—
01010b	CAC/ADC14	—	—	ADTRG0_A	—	—	—	—	—
01011b	BUS	D00	D01	D02	D03	D04	D05	D06	D07
01101b	SLCDC	VL1	VL2	VL3	VL4	COM0	COM1	COM2	COM3
10010b	SSI	—	—	—	—	—	—	—	—
ASEL bit		AN027/ CMPIN0	AN026/ CMPREF0	AN025/ CMPIN1	AN024/ CMPREF1	—	—	—	—
ISEL bit		IRQ2	IRQ1	—	—	IRQ1	IRQ0	—	—
NCODR bit		✓	✓	✓	✓	✓	✓	✓	✓
PCR bit		✓	✓	✓	✓	✓	✓	✓	✓
DSCR bit		L/M	L/M	L/M	L/M	L/M	L/M	L/M	L/M
145-pin product, 144-pin product		✓	✓	✓	✓	✓	✓	✓	✓
121-pin product		✓	✓	✓	✓	✓	✓	✓	✓
100-pin product		✓	✓	✓	✓	✓	✓	✓	✓
64-pin product		✓	✓	✓	✓	✓	✓	✓	✓

✓: Available
—: Setting prohibited

Table 20.6 Register settings for input/output pin function (PORT1) (2)

PSEL[4:0] bits settings	Function	Pin P108	P109	P110	P111	P112	P113	P114	P115
00000b (Value after reset)	Hi-Z/JTAG/SWD	TMS/ SWDIO	TDO/ SWO	TDI	Hi-Z				
00001b	AGT	—	—	—	—	—	—	—	—
00010b	GPT	—	GTOVUP_A	GTOVLO_A	—	—	—	—	—
00011b	GPT	GTIOC0B_A	GTIOC1A_A	GTIOC1B_A	GTIOC3A_A	GTIOC3B_A	—	—	—
00100b	SCI	—	—	CTS2_RTS2_B /SS2_B	SCK2_B	TXD2_B/ MOSI2_B/ SDA2_B	RXD2_B/ MISO2_B/ SCL2_B	—	—
00101b	SCI	CTS9_RTS9_B /SS9_B	TXD9_B/ MOSI9_B/ SDA9_B	RXD9_B/ MISO9_B/ SCL9_B	SCK9_B	—	—	—	—
00110b	SPI	SSLB0_B	MOSIB_B	MISOB_B	RSPCKB_B	—	—	—	—
00111b	IIC	—	—	—	—	—	—	—	—
01000b	KINT	—	—	—	—	—	—	—	—
01001b	CLKOUT/ ACMPHS/ ACMPLP/RTC	—	CLKOUT_B	VCOUT	—	—	—	—	—
01010b	CAC/ADC14	—	—	—	—	—	—	—	—
01011b	BUS	—	—	—	A05	A04	A03	A02	A01
01101b	SLCDC	—	—	—	CAPH	CAPL	SEG0/COM4	SEG24	SEG25
10010b	SSI	—	—	—	—	SSISCK0_B	SSIWS0_B	SSIRXD0_B	SSITXD0_B
ASEL bit		—	—	—	—	—	—	—	—
ISEL bit		—	—	IRQ3	IRQ4	—	—	—	—
NCODR bit		✓	✓	✓	✓	✓	✓	✓	✓
PCR bit		✓	✓	✓	✓	✓	✓	✓	✓
DSCR bit		L/M	L/M	L/M	L/M	L/M	L/M	L/M	L/M
145-pin product, 144-pin product		✓	✓	✓	✓	✓	✓	✓	✓
121-pin product		✓	✓	✓	✓	✓	✓	✓	✓
100-pin product		✓	✓	✓	✓	✓	✓	✓	✓
64-pin product		✓	✓	✓	✓	✓	✓		

✓: Available
—: Setting prohibited

Table 20.7 Register settings for input/output pin function (PORT2) (1)

PSEL[4:0] bits settings	Function	Pin P200	P201	P202	P203	P204	P205	P206
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z						
00001b	AGT	—	—	—	—	AGTIO1_A	AGTO1	—
00010b	GPT	—	—	—	—	GTIW_A	GTIV_A	GTIU_A
00011b	GPT	—	—	GTIOC5B_A	GTIOC5A_A	GTIOC4B_B	GTIOC4A_B	—
00100b	SCI	—	—	SCK2_A	CTS2_RTS2_A/ SS2_A	SCK4_A	TXD4_A/ MOSI4_A/ SDA4_A	RXD4_A/ MISO4_A/ SCL4_A
00101b	SCI	—	—	RXD9_A/ MISO9_A/ SCL9_A	TXD9_A/ MOSI9_A/ SDA9_A	SCK9_A	CTS9_RTS9_A/ SS9_A	—
00110b	SPI	—	—	MISOB_A	MOSIB_A	RSPCKB_A	SSLB0_A	SSLB1_A
00111b	IIC	—	—	—	—	SCL0_B	SCL1_A	SDA1_A
01000b	KINT	—	—	—	—	—	—	—
01001b	CLKOUT/ ACMPHS/ ACMPLP/RTC	—	—	—	—	—	CLKOUT_A	—
01010b	CAC/ADC14	—	—	—	—	CACREF_A	—	—
01011b	BUS	—	—	WR1/ BC1	—	—	A16	WAIT
01100b	CTSU	—	—	—	TSCAP_B	TS00	TSCAP_A	TS01
01101b	SLCDC	—	—	SEG21	SEG22	SEG23	—	—
10000b	CAN	—	—	CRX0_A	CTX0_A	—	—	—
10001b	QSPI	—	—	—	—	—	—	—
10010b	SSI	—	—	—	—	SSISCK1_A	SSIWS1_A	SSIDATA1_A
10011b	USBFS	—	—	—	—	USB_OVRCURB _A	USB_OVRCURA _A	USB_VBUSEN_A
10101b	SDHI	—	—	SD0DAT6	SD0DAT5	SD0DAT4	SD0DAT3	SD0DAT2
ISEL bit		NMI	—	IRQ3	IRQ2	—	IRQ1	IRQ0
NCODR bit		—	✓	✓	✓	✓	✓	✓
PCR bit		—	✓	✓	✓	✓	✓	✓
DSCR bit		—	L/M	L/M	L/M	L/M	L/M	L/M
145-pin product, 144-pin product		✓	✓	✓	✓	✓	✓	✓
121-pin product		✓	✓	✓	✓	✓	✓	✓
100-pin product		✓	✓	✓	✓	✓	✓	✓
64-pin product		✓	✓			✓	✓	✓

✓: Available
—: Setting prohibited

Table 20.8 Register settings for input/output pin function (PORT2) (2)

PSEL[4:0] bits settings	Function	Pin P212	P213	P214	P215
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z			
00001b	AGT	AGTEE1	—	—	—
00010b	GPT	GTETRGD_A	GTETRGC_A	—	—
00011b	GPT	—	—	—	—
00100b	SCI	—	—	—	—
00101b	SCI	RXD1_A/ MISO1_A/ SCL1_A	TXD1_A/ MOSI1_A/ SDA1_A	—	—
00110b	SPI	—	—	—	—
00111b	IIC	—	—	—	—
01000b	KINT	—	—	—	—
01001b	CLKOUT/ ACMPHS/ ACMPLP/RTC	—	—	—	—
01010b	CAC/ADC14	—	—	—	—
01011b	BUS	—	—	—	—
01100b	CTSU	—	—	—	—
01101b	SLCDC	—	—	—	—
10000b	CAN	—	—	—	—
10001b	QSPI	—	—	—	—
10010b	SSI	—	—	—	—
10011b	USBFS	—	—	—	—
10101b	SDHI	—	—	—	—
ISEL bit		IRQ3	IRQ2	—	—
NCODR bit		✓	✓	—	—
PCR bit		✓	✓	—	—
DSCR bit		—	—	—	—
145-pin product, 144-pin product		✓	✓	✓	✓
121-pin product		✓	✓	✓	✓
100-pin product		✓	✓	✓	✓
64-pin product		✓	✓	✓	✓

✓: Available
—: Setting prohibited

Table 20.9 Register settings for input/output pin function (PORT3)

PSEL[4:0] bits settings	Function	Pin P300	P301	P302	P303	P304	P305	P306	P307
00000b (Value after reset)	Hi-Z/JTAG/SWD	TCK/ SWCLK	Hi-Z						
00010b	GPT	—	GTOULO_A	GTOUUP_A	—	—	—	—	—
00011b	GPT	GTIOC0A_A	GTIOC4B_A	GTIOC4A_A	GTIOC7B_A	GTIOC7A_A	—	—	—
00100b	SCI	—	RXD2_A/ MISO2_A/ SCL2_A	TXD2_A/ MOSI2_A/ SDA2_A	—	—	—	—	—
00110b	SPI	SSLB1_B	SSLB2_B	SSLB3_B	—	—	—	—	—
01011b	BUS	—	A06	A07	A08	A09	A10	A11	A12
01101b	SLCDC	—	SEG1/COM5	SEG2/COM6	SEG3/COM7	SEG17	SEG16	SEG15	SEG14
10101b	SDHI/MMC	—	—	—	—	—	—	—	—
ISEL bit		—	IRQ6	IRQ5	—	IRQ9	IRQ8	—	—
NCODR bit		✓	✓	✓	✓	✓	✓	✓	✓
PCR bit		✓	✓	✓	✓	✓	✓	✓	✓
DSCR bit		L/M	L/M	L/M	L/M	L/M	L/M	L/M	L/M
145-pin product, 144-pin product		✓	✓	✓	✓	✓	✓	✓	✓
121-pin product		✓	✓	✓	✓	✓	✓	✓	✓
100-pin product		✓	✓	✓	✓	✓	✓	✓	✓
64-pin product		✓	✓	✓	✓	✓			

PSEL[4:0] bits settings	Function	Pin P308	P309	P310	P311	P312	P313	P314	P315
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
00010b	GPT	—	—	—	—	—	—	—	—
00011b	GPT	—	—	—	—	—	—	—	—
00100b	SCI	—	—	—	—	—	—	—	—
00110b	SPI	—	—	—	—	—	—	—	—
01011b	BUS	A13	A14	A15	CS2	CS3	—	—	—
01101b	SLCDC	SEG13	SEG12	SEG11	SEG10	SEG9	SEG20	SEG4	SEG5
10101b	SDHI	—	—	—	—	—	SD0DAT7	—	—
ISEL bit		—	—	—	—	—	—	—	—
NCODR bit		✓	✓	✓	✓	✓	✓	✓	✓
PCR bit		✓	✓	✓	✓	✓	✓	✓	✓
DSCR bit		L/M	L/M	L/M	L/M	L/M	L/M	L/M	L/M
145-pin product, 144-pin product		✓	✓	✓	✓	✓	✓	✓	✓
121-pin product		✓	✓				✓	✓	✓
100-pin product									
64-pin product									

✓: Available
—: Setting prohibited

Table 20.10 Register settings for input/output pin function (PORT4) (1)

PSEL[4:0] bits settings	Function	Pin P400	P401	P402	P403	P404	P405	P406	P407
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
00001b	AGT	—	—	—	—	—	—	—	—
00010b	GPT	—	GTETRGA_B	—	—	—	—	—	—
00011b	GPT	GTIOC6A_A	GTIOC6B_A	—	GTIOC3A_B	GTIOC3B_B	GTIOC1A_B	GTIOC1B_B	—
00100b	SCI	SCK4_B	CTS4_RTS4_B /SS4_B	—	—	—	—	—	CTS4_RTS4_A /SS4_A
00101b	SCI	—	—	—	—	—	—	—	—
00110b	SPI	—	—	—	—	—	—	—	SSLB3_A
00111b	IIC	SCL0_A	SDA0_A	—	—	—	—	—	SDA0_B
01001b	CLKOUT/ ACMPHS/ ACMPLP/RTC	—	—	—	—	—	—	—	RTCOUT
01010b	CAC/ADC14	—	—	—	—	—	—	—	ADTRG0_B
01100b	CTSU	TS20	TS19	TS18	TS17	TS16	TS15	TS14	TS03
10000b	CAN	—	CTX0_B	CRX0_B	—	—	—	—	—
10010b	SSI	AUDIO_CLK	—	—	SSISCK0_A	SSIWS0_A	SSITXD0_A	SSIRXD0_A	—
10011b	USBFS	—	—	—	—	—	—	—	USB_VBUS
10101b	SDHI	—	—	—	—	—	—	—	—
Don't care	RTC/AGT/VBATT	—	—	RTCIC0*1/ AGTIO0_B*1/ AGTIO1_B*1	RTCIC1*1/ AGTIO0_C*1/ AGTIO1_C*1	RTCIC2*1	—	—	—
ASEL bit		—	—	—	—	—	—	—	—
ISEL bit		IRQ0	IRQ5	IRQ4	—	—	—	—	—
NCODR bit		✓	✓	✓	✓	✓	✓	✓	✓
PCR bit		✓	✓	✓	✓	✓	✓	✓	✓
DSCR bit		L/M	L/M	L/M	L/M	L/M	L/M	L/M	L/M
145-pin product, 144-pin product		✓	✓	✓	✓	✓	✓	✓	✓
121-pin product		✓	✓	✓	✓	✓	✓	✓	✓
100-pin product		✓	✓	✓	✓	✓	✓	✓	✓
64-pin product		✓	✓	✓					✓

✓: Available
—: Setting prohibited

Note 1. To use this pin function, set the corresponding pin as general input (set the PmnPFS.PDR and PmnPFS.PMR bits to 0).

Table 20.11 Register settings for input/output pin function (PORT4) (2)

PSEL[4:0] bits settings	Function	Pin P408	P409	P410	P411	P412	P413	P414	P415
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
00001b	AGT	—	—	AGTOB1	AGTOA1	—	—	—	—
00010b	GPT	GTOWLO_B	GTOWUP_B	GTOVLO_B	GTOVUP_B	GTOULO_B	GTOUUP_B	—	—
00011b	GPT	—	—	GTIOC9B_A	GTIOC9A_A	—	—	—	—
00100b	SCI	—	—	RXD0_B/ MISO0_B/ SCL0_B	TXD0_B/ MOSI0_B/ SDA0_B	SCK0_B	CTS0_RTS0_B /SS0_B	—	—
00101b	SCI	RXD3_A/ MISO3_A/ SCL3_A	TXD3_A/ MOSI3_A/ SDA3_A	SCK3_A	CTS3_RTS3_A /SS3_A	—	—	—	—
00110b	SPI	—	—	MISOA_B	MOSIA_B	RSPCKA_B	SSLA0_B	SSLA1_B	SSLA2_B
00111b	IIC	—	—	—	—	—	—	—	—
01001b	CLKOUT/ ACMPHS/ ACMPLP/RTC	—	—	—	—	—	—	—	—
01010b	CAC/ADC14	—	—	—	—	—	—	—	—
01100b	CTSU	TS04	TS05	TS06	TS07	TS08	TS09	TS10	TS11
10000b	CAN	—	—	—	—	—	—	—	—
10010b	SSI	—	—	—	—	—	—	—	—
10011b	USBFS	USB_ID_A	USB_EXICEN_ A	—	—	—	—	—	—
10101b	SDHI	—	—	SD0DAT1	SD0DAT0	SD0CMD	SD0CLK	SD0WP	—
Don't care	RTC/AGT/VBATT	—	—	—	—	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—
ISEL bit		IRQ7	IRQ6	IRQ5	IRQ4	—	—	—	—
NCODR bit		✓	✓	✓	✓	✓	✓	✓	✓
PCR bit		✓	✓	✓	✓	✓	✓	✓	✓
DSCR bit		L/M	L/M	L/M	L/M	L/M	L/M	L/M	L/M
145-pin product, 144-pin product		✓	✓	✓	✓	✓	✓	✓	✓
121-pin product		✓	✓	✓	✓	✓	✓	✓	✓
100-pin product		✓	✓	✓	✓	✓	✓	✓	✓
64-pin product		✓	✓	✓	✓				

Table 20.12 Register settings for input/output pin function (PORT5)

PSEL[4:0] bits settings	Function	Pin P500	P501	P502	P503	P504	P505	P506	P507
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
00001b	AGT	AGTOA0	AGTOB0	—	—	—	—	—	—
00010b	GPT	GTIU_B	GTIV_B	GTIW_B	GTETRGC_B	GTETRGD_B	—	—	—
00011b	GPT	—	—	—	—	—	—	—	—
00100b	SCI	—	—	—	—	—	—	—	—
00111b	IIC	—	—	—	—	—	—	—	—
01101b	SLCDC	SEG48	SEG49	SEG50	SEG51	—	—	—	—
10001b	QSPI	QSPCLK	QSSL	QIO0	QIO1	QIO2	QIO3	—	—
10011b	USBFS	USB_VBUSEN_B	USB_OVRCURA_B	USB_OVRCURB_B	USB_EXICEN_B	USB_ID_B	—	—	—
ASEL bit		AN016	AN017	AN018	AN019	AN020	AN021	AN022	AN023
ISEL bit		—	IRQ11	IRQ12	—	—	IRQ14	IRQ15	—
NCODR bit		✓	✓	✓	✓	✓	✓	✓	✓
PCR bit		✓	✓	✓	✓	✓	✓	✓	✓
DSCR bit		L/M	L/M	L/M	L/M	L/M	L/M	L/M	L/M
145-pin product, 144-pin product		✓	✓	✓	✓	✓	✓	✓	✓
121-pin product		✓	✓	✓	✓	✓	✓	✓	
100-pin product		✓	✓	✓	✓	✓	✓		
64-pin product		✓	✓	✓					

PSEL[4:0] bits settings	Function	Pin P511	P512
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z	
00001b	AGT	—	—
00010b	GPT	—	—
00011b	GPT	GTIOC0B_B	GTIOC0A_B
00100b	SCI	RXD4_B/MISO4_B/SCL4_B	TXD4_B/MOSI4_B/SDA4_B
00111b	IIC	SDA2	SCL2
01101b	SLCDC	—	—
10001b	QSPI	—	—
10011b	USBFS	—	—
ASEL bit		—	—
ISEL bit		IRQ15	IRQ14
NCODR bit		✓	✓
PCR bit		✓	✓
DSCR bit		L/M	L/M
145-pin product, 144-pin product		✓	✓
121-pin product		✓	✓
100-pin product			
64-pin product			

✓: Available
—: Setting prohibited

Table 20.13 Register settings for input/output pin function (PORT6)

PSEL[4:0] bits settings	Function	Pin P600	P601	P602	P603	P604	P605	P606
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z						
01011b	BUS	RD	WR/ WR0	EBCLK	D13	D12	D11	—
01101b	SLCDC	SEG41	SEG40	SEG39	SEG38	SEG37	SEG36	SEG35
NCODR bit		✓	✓	✓	✓	✓	✓	✓
PCR bit		✓	✓	✓	✓	✓	✓	✓
DSCR bit		L/M	L/M	L/M	L/M	L/M	L/M	L/M
145-pin product, 144-pin product		✓	✓	✓	✓	✓	✓	✓
121-pin product		✓	✓	✓	✓	✓	✓	
100-pin product		✓	✓	✓	✓			
64-pin product								

PSEL[4:0] bits settings	Function	Pin P608	P609	P610	P611	P612	P613	P614
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z						
01011b	BUS	A00/ BC0	CS1	CS0	—	D8	D9	D10
01101b	SLCDC	SEG28	SEG29	SEG30	SEG31	SEG32	SEG33	SEG34
NCODR bit		✓	✓	✓	✓	✓	✓	✓
PCR bit		✓	✓	✓	✓	✓	✓	✓
DSCR bit		L/M	L/M	L/M	L/M	L/M	L/M	L/M
145-pin product, 144-pin product		✓	✓	✓	✓	✓	✓	✓
121-pin product		✓	✓	✓	✓	✓	✓	
100-pin product		✓	✓	✓				
64-pin product								

✓: Available
—: Setting prohibited

Table 20.14 Register settings for input/output pin function (PORT7)

PSEL[4:0] bits settings	Function	Pin P700	P701	P702	P703	P704	P705
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z					
00011b	GPT	GTIOC5A_B	GTIOC5B_B	GTIOC6A_B	GTIOC6B_B	—	—
00101b	SCI	—	—	—	—	—	—
00110b	SPI	—	—	—	—	—	—
01010b	CAC/ADC14	—	—	—	—	—	—
01100b	CTSU	TS32	TS33	TS34	—	—	—
ISEL bit		—	—	—	—	—	—
NCODR bit		✓	✓	✓	✓	✓	✓
PCR bit		✓	✓	✓	✓	✓	✓
DSCR bit		L/M	L/M	L/M	L/M	L/M	L/M
145-pin product, 144-pin product		✓	✓	✓	✓	✓	✓
121-pin product		✓	✓	✓			
100-pin product							
64-pin product							

PSEL[4:0] bits settings	Function	Pin P708	P709	P710	P711	P712	P713
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z					
00011b	GPT	—	—	—	—	GTIOC2B_B	GTIOC2A_B
00101b	SCI	RXD1_B/ MISO1_B/ SCL1_B	TXD1_B/ MOSI1_B/ SDA1_B	SCK1_B	CTS1_RTS1_B /SS1_B	—	—
00110b	SPI	SSLA3_B	—	—	—	—	—
01010b	CAC/ADC14	CACREF_B	—	—	—	—	—
01100b	CTSU	TS12	TS13	TS35	—	—	—
ISEL bit		IRQ11	IRQ10	—	—	—	—
NCODR bit		✓	✓	✓	✓	✓	✓
PCR bit		✓	✓	✓	✓	✓	✓
DSCR bit		L/M	L/M	L/M	L/M	L/M	L/M
145-pin product, 144-pin product		✓	✓	✓	✓	✓	✓
121-pin product		✓	✓	✓			
100-pin product		✓					
64-pin product							

✓: Available
—: Setting prohibited

Table 20.15 Register settings for input/output pin function (PORT8)

PSEL[4:0] bits settings	Function	Pin P800	P801	P802	P803	P804	P805	P806	P807
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z							
01011b	BUS	D14	D15	—	—	—	—	—	—
01101b	SLCDC	SEG44	SEG45	SEG46	SEG47	SEG43	SEG42	SEG26	SEG27
NCODR bit		✓	✓	✓	✓	✓	✓	✓	✓
PCR bit		✓	✓	✓	✓	✓	✓	✓	✓
DSCR bit		L/M	L/M	L/M	L/M	L/M	L/M	L/M	L/M
145-pin product, 144-pin product		✓	✓	✓	✓	✓	✓	✓	✓
121-pin product		✓	✓						
100-pin product									
64-pin product									

PSEL[4:0] bits settings	Function	Pin P808	P809
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z	
01011b	BUS	—	—
01101b	SLCDC	SEG18	SEG19
NCODR bit		✓	✓
PCR bit		✓	✓
DSCR bit		L/M	L/M
145-pin product, 144-pin product		✓	✓
121-pin product		✓	✓
100-pin product		✓	✓
64-pin product			

✓: Available
—: Setting prohibited

Table 20.16 Register settings for input/output pin function (PORT9)

PSEL[4:0] bits settings	Function	Pin P900	P901	P902
00000b (Value after reset)	Hi-Z/JTAG/SWD	Hi-Z		
01101b	SLCDC	SEG6	SEG7	SEG8
NCODR bit		✓	✓	✓
PCR bit		✓	✓	✓
DSCR bit		L/M	L/M	L/M
145-pin product, 144-pin product		✓	✓	✓
121-pin product				
100-pin product				
64-pin product				

✓: Available
—: Setting prohibited

21. Key Interrupt Function (KINT)

21.1 Overview

A key interrupt (KEY_INTKR) can be generated by setting the Key Return Mode Register (KRM) and inputting a rising or falling edge to the key interrupt input pins, KR00 to KR07. Table 21.1 shows the assignment for key interrupt detection, Table 21.2 shows the function configuration, and Figure 21.1 shows the block diagram.

Table 21.1 Assignment of key interrupt detection pins

Key Interrupt Mode Control n (n = 0-7)	Description
KRM0	Controls KR00 signal in 1-bit units
KRM1	Controls KR01 signal in 1-bit units
KRM2	Controls KR02 signal in 1-bit units
KRM3	Controls KR03 signal in 1-bit units
KRM4	Controls KR04 signal in 1-bit units
KRM5	Controls KR05 signal in 1-bit units
KRM6	Controls KR06 signal in 1-bit units
KRM7	Controls KR07 signal in 1-bit units

Table 21.2 Configuration of Key Interrupt Function (KINT)

Parameter	Configuration
Input	KR00 to KR07
Control registers	Key Return Control Register (KRCTL) Key Return Mode Register (KRM) Key Return Flag Register (KRF)

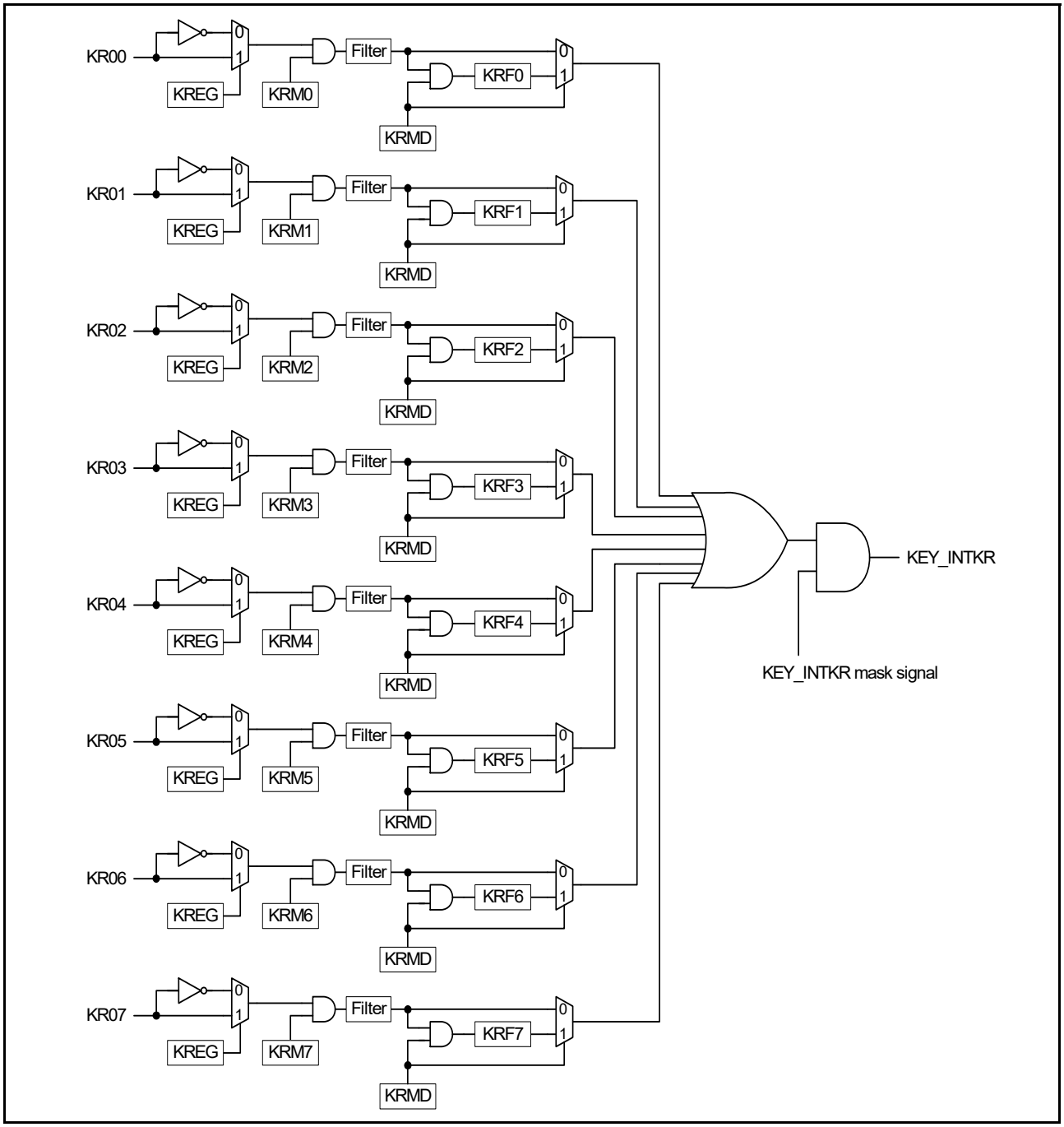


Figure 21.1 Key interrupt function block diagram

In Figure 21.1, all key return factors are merged by an OR gate, and the key interrupt (KEY_INTKR) is the output of AND gate to mask the merged key return factor by the KEY_INTKR mask signal. When using KRFn (KRMD = 1), KEY_INTKR mask signal is used as the output mask that is asserted by clearing KRFn.

21.2 Register Descriptions

21.2.1 Key Return Control Register (KRCTL)

Address(es): KINT.KRCTL 4008 0000h

	b7	b6	b5	b4	b3	b2	b1	b0
	KRMD	—	—	—	—	—	—	KREG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	KREG	Selection of Detection Edge (KR00 to KR07)	0: Falling edge 1: Rising edge.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	KRMD	Usage of Key Interrupt Flags (KRF0 to KRF7)	0: Do not use key interrupt flags 1: Uses key interrupt flags.	R/W

The KRCTL controls the usage of the key interrupt flags, KRF0 to KRF7, and sets the detection edge.

21.2.2 Key Return Flag Register (KRF)

Address(es): KINT.KRF 4008 0004h

	b7	b6	b5	b4	b3	b2	b1	b0
	KRF7	KRF6	KRF5	KRF4	KRF3	KRF2	KRF1	KRF0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	KRFn	Key Interrupt Flag n	0: No key interrupt detected 1: Key interrupt detected.	R/W

n = 0 to 7

Note: When KRMD = 0, setting the KRFn bit to 1 is prohibited.
When setting the KRFn bit to 1, the KRFn value does not change. To clear the KRFn bit, confirm that the target bit is 1 before writing to 0 to the bit, then write 1 to the other bits.

The KRF controls the key interrupt flags, KRF0 to KRF7.

21.2.3 Key Return Mode Register (KRM)

Address(es): KINT.KRM 4008 0008h

	b7	b6	b5	b4	b3	b2	b1	b0
	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	KRMn	Key Interrupt Mode Control n	0: Does not detect key interrupt signal 1: Detects key interrupt signal	R/W

n = 0 to 7

Note: The on-chip pull-up resistors can be applied by setting the associated key interrupt input pin in the pull-up resistor. For details, see section 20, I/O Ports.
Key interrupts can be assigned by the PmnPFS.PSEL bits. For more information, see section 20, I/O Ports.
An interrupt is generated when the target bit in the KRM is set while a low level (KREG is set to 0) or a high level (KREG is set to 1) is being input to the key interrupt input pin. To ignore this interrupt, set the KRM after disabling the interrupt handling.

The KRM sets the key interrupt mode.

21.3 Operation

21.3.1 When Not Using Key Interrupt Flag (KRMD = 0)

A key interrupt (KEY_INTKR) is generated when the valid edge specified in the KREG bit is input to a key interrupt pin, KR00 to KR07. To identify the channel to which the valid edge is input, read the port register and check the port level after the key interrupt (KEY_INTKR) is generated.

The KEY_INTKR signal changes according to the input level of the key interrupt input pin, KR00 to KR07.

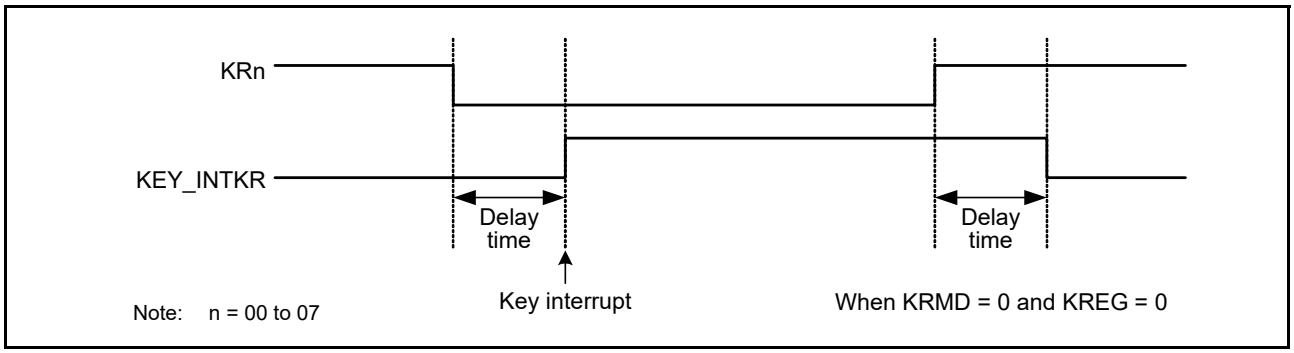


Figure 21.2 Operation of KEY_INTKR signal when key interrupt is input to a single channel

Figure 21.3 shows the operation when a valid edge is input to multiple key interrupt input pins. The KEY_INTKR signal is set while a low level is being input to one pin, that is, when KREG is set to 0. Therefore, even if a falling edge is input to another pin in this period, a key interrupt (KEY_INTKR) is not generated again. See [1] in Figure 21.3.

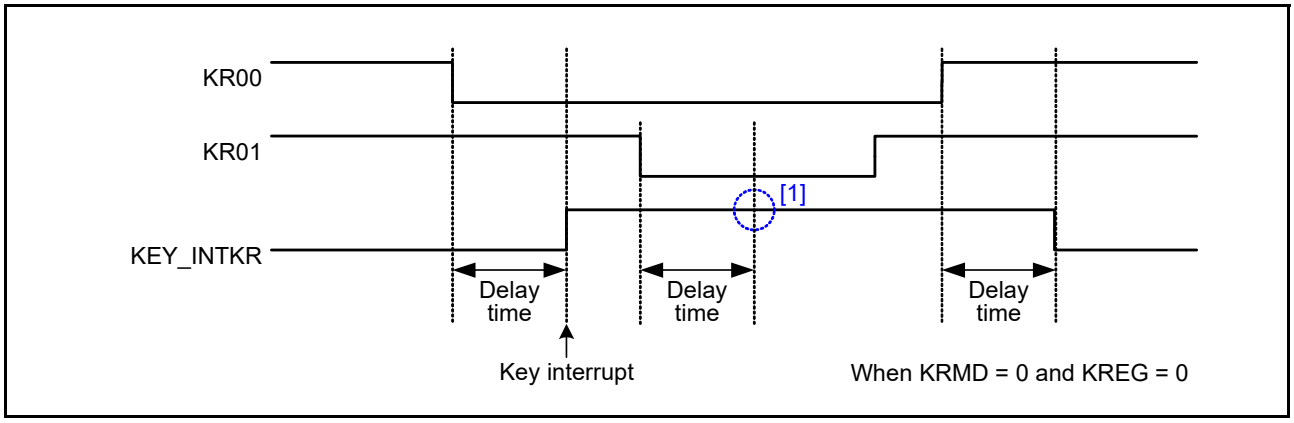


Figure 21.3 Operation of KEY_INTKR signal when key interrupts are input to multiple channels

21.3.2 When Using Key Interrupt Flag (KRMD = 1)

A key interrupt (KEY_INTKR) is generated when the valid edge specified in the KREG bit is input to a key interrupt pin, KR00 to KR07. To identify the channels to which the valid edge is input, read the Key Return Flag Register (KRF) after the key interrupt (KEY_INTKR) is generated. If the KRMD bit is set to 1, clear the KEY_INTKR signal by clearing the associated bit in the KRF.

As Figure 21.4 shows, only one interrupt is generated each time a falling edge is input to one channel, that is, when KREG = 0, regardless of whether the KRFn bit is cleared before or after a rising edge is input.

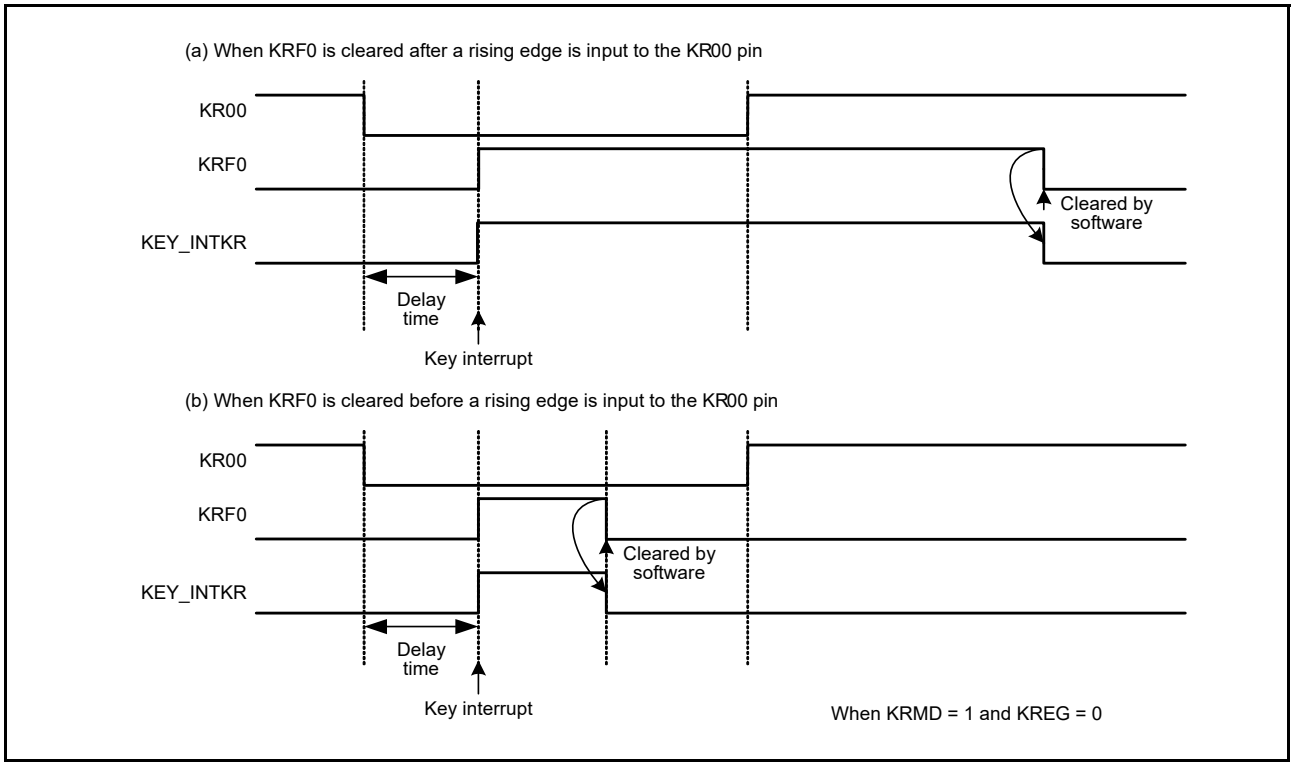


Figure 21.4 Basic operation of KEY_INTKR signal when key interrupt flag is used

The operation when a valid edge is input to multiple key interrupt input pins is shown in Figure 21.5. A falling edge is also input to the KR01 and KR05 pins after a falling edge is input to the KR00 pin, that is, when KREG = 0. The KRF1 bit is set when the KRF0 bit is cleared. A key interrupt generates one clock, PCLKB, after the KRF0 bit is cleared. See [1] in Figure 21.5. Also, after a falling edge is input to the KR05 pin, the KRF5 bit is set. See [2] in the figure when the KRF1 bit is cleared. A key interrupt generates one clock, PCLKB, after the KRF1 bit is cleared. See [3] in the figure. It is therefore possible to generate a key interrupt when a valid edge is input to multiple channels.

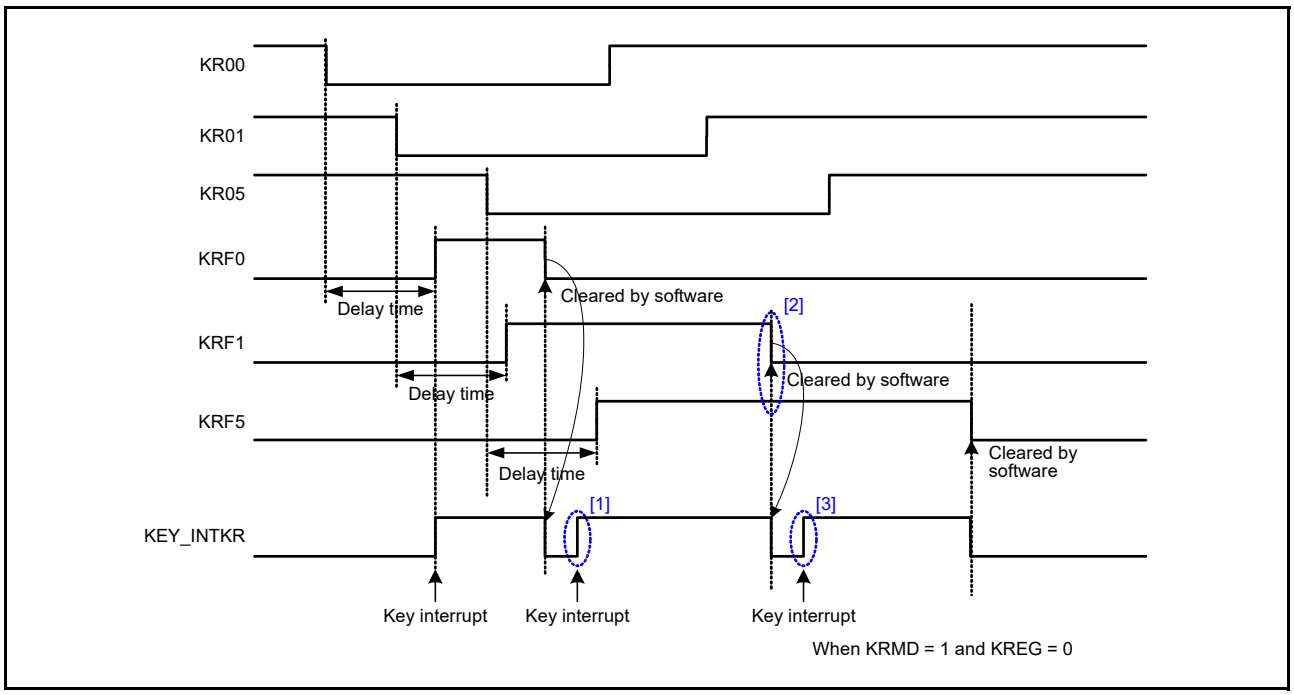


Figure 21.5 Operation of KEY_INTKR signal when key interrupts are input to multiple channels

21.4 Usage Note

- If the KEY_INTKR is used as the snooze request, the KRMD must be set to 0
- If the KEY_INTKR is used as the interrupt source for returning to Normal mode from Snooze mode and Software Standby mode, the KRMD must be set to 1
- When the Key Interrupt function (KINT) is assigned to a pin by the MPC, this pin input is always enabled in the Software Standby mode, and if this pin level changes, the associated KRFn can be set. Therefore, a key interrupt might occur on canceling the Software Standby mode.

To ignore changes to the key interrupt pin during a software standby, clear the associated KRM bit before entering software standby. After canceling software standby, you must clear KRFn before the associated KRM bit is set.

22. Port Output Enable for GPT (POEG)

Use the Port Output Enable (POEG) function to 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 (n = A,B,C, D) pins
- Output-disable request from the GPT
- Comparator interrupt request detection
- Oscillation stop detection of the Clock Generation Circuit
- Register settings.

The GTETRGn (n = A,B,C, D) pins can also be used as GPT external trigger input pins.

22.1 Overview

Table 22.1 lists the specifications of the POEG. Figure 22.1 shows a block diagram of the POEG.

Table 22.1 POEG specifications

Parameter	Description
Output-disable control by the input level detection	• The GPT output pins can disabled when a GTETRGn rising edge or high is sampled after polarity and filter selection.
Output-disable request from GPT	• When the GTIOCA pin and the GTIOCB pin are driven to an active level simultaneously, the GPT generates an output-disable request to the POEG. Through reception of these requests, the POEG can control whether the GTIOCA and GTIOCB pins are placed in output disable.
Output-disable control by the comparator (ACMPHS) interrupt request detection	• The GPT output pins can be disabled when an interrupt request is generated by a change in the output results of any of the comparators.
Output-disable control by the oscillation stop detection	• The GPT output pins can be disabled when oscillation by the Clock Generation Circuit stops.
Output-disable control by software (registers)	• The GPT output pins can be disabled by modifying the register settings.
Interrupts	• Allows output-disable control by the input level detection • Allows output-disable request from GPT or ACMPHS.
External trigger output function to GPT (count start/count stop/count clear/up-count/ down-count/input capture function)	• The GTETRGn signals can be output to the GPT after polarity and filter selection.
Noise filtering	• Three times sampling for every PCLKB/1, PCLKB/8, PCLKB/32, or PCLKB/128 can be set for any of the input pins GTETRGn • Positive or negative polarity can be selected for any of the input pins GTETRGn • The signal state after polarity and filter selection can be monitored.

Figure 22.1 shows a block diagram of the POEG.

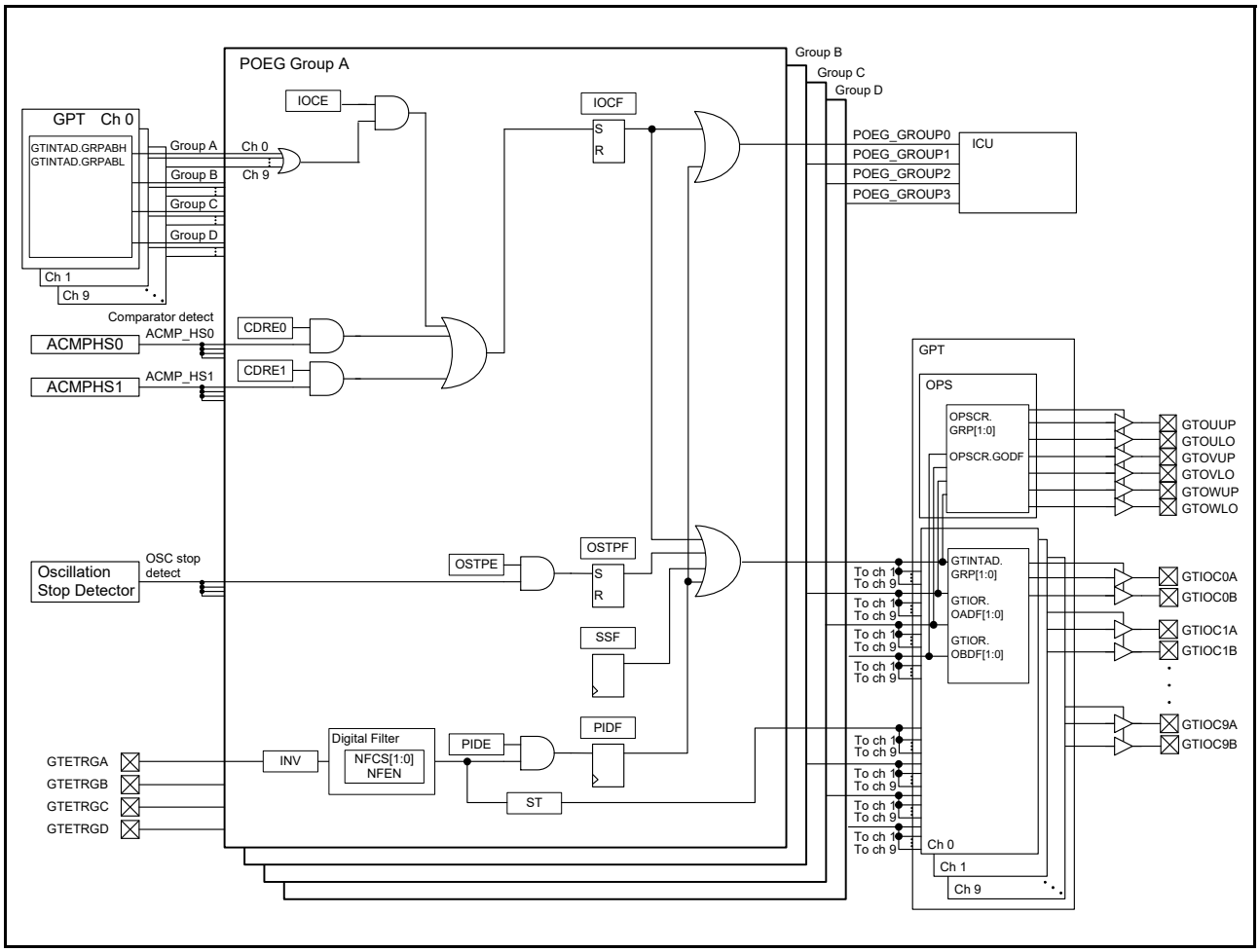


Figure 22.1 POEG block diagram

Table 22.2 shows input pins to be used by the POEG.

Table 22.2 POEG input pins

Pin Name	I/O	Description
GTETRGA	Input	GPT output pin output disable request signal and GPT external trigger input pin A
GTETRGB	Input	GPT output pin output disable request signal and GPT external trigger input pin B
GTETRGC	Input	GPT output pin output disable request signal and GPT external trigger input pin C
GTETRGD	Input	GPT output pin output disable request signal and GPT external trigger input pin D

22.2 Register Descriptions

22.2.1 POEG Group n Setting Register (POEGGn) (n = A to D)

Address(es): POEG.POEGGA 4004 2000h, POEG.POEGGB 4004 2100h, POEG.POEGGC 4004 2200h, POEG.POEGGD 4004 2300h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	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	Bit name	Description	R/W
b0	PIDF	Port Input Detection Flag	0: No output-disable request from the GTETRGn pin has occurred 1: Output-disable request from the GTETRGn pin occurred.	R(/W)*1
b1	IOCF	Output-disable Request Detection Flag from GPT or ACMPHS	0: No output-disable request from the GPT disable request or comparator interrupt has occurred 1: Output-disable request from the GPT disable request or comparator interrupt occurred.	R(/W)*1
b2	OSTPF	Oscillation Stop Detection Flag	0: No output-disable request from oscillation stop detection has occurred 1: Output-disable request from oscillation stop detection occurred.	R(/W)*1
b3	SSF	Software Stop Flag	0: No output-disable request from software has occurred 1: Output-disable request from software occurred.	R/W
b4	PIDE	Port Input Detection Enable	0: Output-disable request from the GTETRGn pins disabled 1: Output-disable request from the GTETRGn pins enabled.	R/W*2
b5	IOCE	Enable for GPT Output-disable Request	0: Disable output-disable requests from GPT disable request 1: Enable output-disable requests from GPT disable request.	R/W*2
b6	OSTPE	Oscillation Stop Detection Enable	0: Output-disable request from the oscillation stop detection disabled 1: Output-disable request from the oscillation stop detection enabled.	R/W*2
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	CDRE0	ACMP_HS0 Enable	0: Disable requests from comparator 0 disabled 1: Disable requests from comparator 0 enabled.	R/W*2
b9	CDRE1	ACMP_HS1 Enable	0: Disable requests from comparator 1 disabled 1: Disable requests from comparator 1 enabled.	R/W*2
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	ST	GTETRGn Input Status Flag	0: GTETRGn input after filtering is 0 1: GTETRGn input after filtering is 1.	R
b27 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28	INV	GTETRGn Input Reverse	0: GTETRGn input 1: GTETRGn input reversed.	R/W
b29	NFEN	Noise Filter Enable	0: Filtering noise disabled 1: Filtering noise enabled.	R/W
b31, b30	NFCS[1:0]	Noise Filter Clock Select	b1 b0 0 0: GTETRGn pin input level sampled three times every PCLKB 0 1: GTETRGn pin input level sampled three times every PCLKB/8 1 0: GTETRGn pin input level sampled three times every PCLKB/32 1 1: GTETRGn pin input level sampled three times every PCLKB/128.	R/W

Note 1. Only 0 can be written to clear the flag.
Note 2. Can be modified only once after a reset.

The POEGGA to POEGGD registers control the output disable state of the GPT pins output, interrupts, and the external trigger input to GPT. In the descriptions, POEGGn represents all the POEGGA to POEGGD registers.

22.3 Output-Disable Control Operation

If any of the following conditions is satisfied, GTIOCxA, GTIOCxB, and the three-phase PWM output for BLDC motor control pins can be set to output-disable.

- Input level or edge detection of the GTETRGn pins
  When POEGGn.PIDE is 1, the POEGGn.PIDF flag is set to 1.
- Output-disable request from the GPT
  When POEGGn.IOCE is 1, the POEGGn.IOCF flag is set to 1 if the disable request enabled in the GTINTAD.GRPABH, or GTINTAD.GRPABL bits applies to the group selected in the GPT registers GTINTAD.GRP[1:0] and OPSCR.GRP[1:0].
- Comparator (ACMPHS) interrupt request detection
  Comparator interrupt detection is activated when any of the POEGGn.CDRE[5:0] registers is 1. When the associated comparator interrupt is generated, the GPT output pins are disabled. POEGGn.IOCF indicates the detection status.
- Oscillation stop detection for the Clock Generation Circuit
  When POEGGn.OSTPE is 1, the POEGGn.OSTPF flag is set to 1.
- SSF bit setting
  When POEGGn.SSF is set to 1.

The state of output-disable is controlled in the GPT. The output-disable of the GTIOCxA and GTIOCxB pins are set in the GTINTAD.GRP[1:0], GTIOR.OADF[1:0], and GTIOR.OBDF[1:0] bits in the GPT. The output-disable of the three-phase PWM output for BLDC motor control pins are set to the OPSCR.GRP[1:0] and OPSCR.GODF bits in GPT_OPS.

22.3.1 Pin Input Level Detection Operation

If the input conditions set by POEGGn.PIDE, POEGGn.NFCS[1:0], POEGGn.NFEN and POEGGn.INV occur on the GTETRGn pins, the GPT output pins are placed in output disable.

22.3.1.1 Digital Filter

Figure 22.2 shows high level detection by the digital filter. When a high level associated with the POEGGn.INV polarity setting is detected three times consecutively with the sampling clock selected in POEGGn.NFCS[1:0] and POEGGn.NFEN, the detected level is recognized as high, and the GPT output pins are placed in output disable. If even one low level is detected during this interval, the detected level is not recognized as high. In addition, in an interval where the sampling clock is not being output, changes of the levels on the GTETRGn pins are ignored.

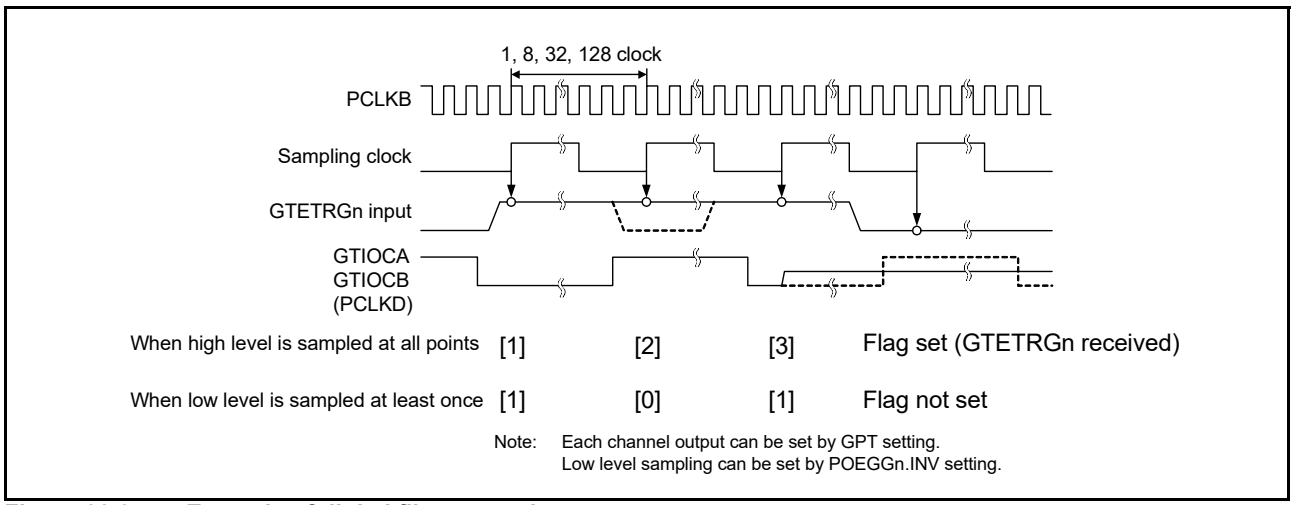


Figure 22.2 Example of digital filter operation

22.3.2 Output-Disable Request from GPT

For details on the operation, see the GTIOC Pin Output Negate Control in section 23, General PWM Timer (GPT).

22.3.3 Comparator Interrupt Detection

If POEGGn.CDRE[1:0] is 1 when a corresponding comparator interrupt request is generated, the GPT output pins are placed in output disable for each group. The status flag is POEGGn.IOCF, which is shared with GPT output disable detection.

22.3.4 Output Disable Control on Detection of Stopped Oscillation

When the oscillation stop detection function in the Clock Generation Circuit detects stopped oscillation while POEGGn.OSTPE is 1, the GPT output pins are placed in output disable for each group.

22.3.5 Output Disable Control Using Registers

The GPT output pins can be directly controlled by writing to the software stop flag, POEGGn.SSF.

22.3.6 Release from Output Disable

To release the GPT output pins placed in the output-disable state, either return them to their initial state with a reset or clear all of the following:

- POEGGn.PIDF flag
- POEGGn.IOCF flag
- POEGGn.OSTPF flag
- POEGGn.SSF flag.

Writing 0 to the POEGGn.PIDF flag is ignored (the flag is not cleared) if the external input pins GTETRGn are not disabled and the POEGGn.ST bit is not set to 0.

Writing 0 to the POEGGn.IOCF flag is valid (the flag is cleared) only if all of the GTST.OABHF and GTST.OABLF flags in GPT are set to 0.

Writing 0 to the POEGGn.OSTPF flag is ignored (the flag is not cleared) if the OSTDSR.OSTDF flag in the Clock Generation Circuit is not set to 0. In addition, when the flag set and release occur at the same time, the flag set takes precedence.

Figure 22.3 shows the released timing for output disable. The output disable is released at the beginning of the next count cycle of the GPT after the flag is cleared.

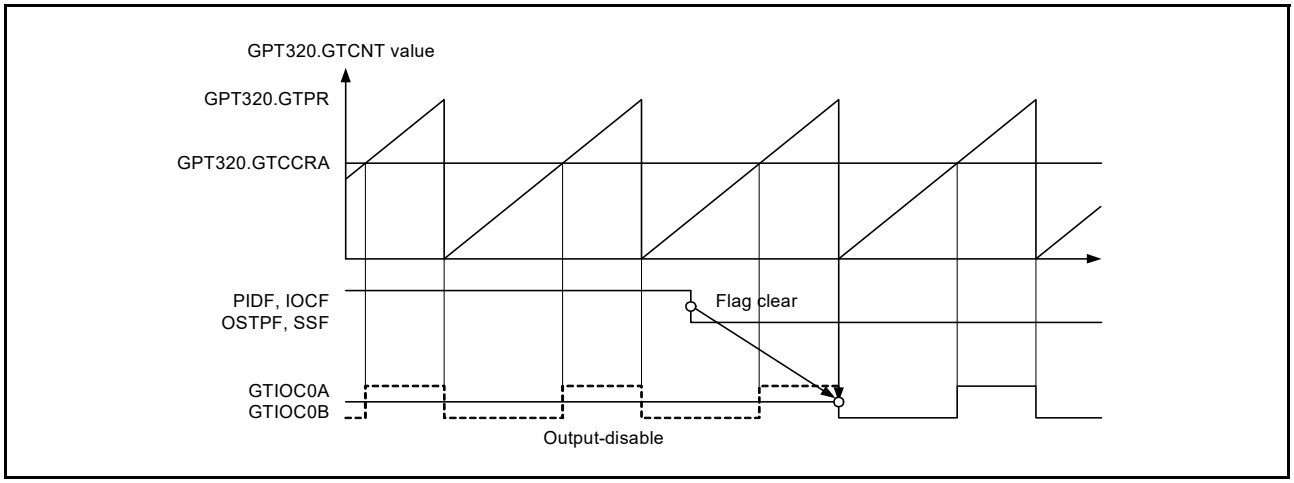


Figure 22.3 Output-disable release timing for GPT pin outputs

22.4 Interrupt Sources

The POEG generates an interrupt request when triggered by these sources:

- Output-disable control by the input level detection
- Output-disable request from GPT
- Comparator interrupt request detection.

Table 22.3 lists the conditions for interrupt requests.

Table 22.3 Interrupt sources and conditions

Interrupt source	Symbol	Associated flag	Trigger conditions
POEG Group A interrupt	POEG_GROUP0	POEGGA.IOCF	An output-disable request from a GPT disable request occurred
			An output-disable request from a comparator interrupt occurred
		POEGGA.PIDF	An output-disable request from the GTETRGA pin occurred
POEG Group B interrupt	POEG_GROUP1	POEGGB.IOCF	An output-disable request from a GPT disable request occurred
			An output-disable request from a comparator interrupt occurred
		POEGGB.PIDF	An output-disable request from the GTETRGB pin occurred
POEG Group C interrupt	POEG_GROUP2	POEGGC.IOCF	An output-disable request from a GPT disable request occurred
			An output-disable request from a comparator interrupt occurred
		POEGGC.PIDF	An output-disable request from the GTETRGC pin occurred
POEG Group D interrupt	POEG_GROUP3	POEGGD.IOCF	An output-disable request from GPT disable request has been generated.
			An output-disable request from comparator interrupt has been generated.
		POEGGD.PIDF	An output-disable request from the GTETRGD pin has been generated.

22.5 External Trigger Output to GPT

The POEG outputs the GTETRGn signals as the GPT operation trigger signal for the following:

- Count start
- Count stop
- Count clear
- Up-count
- Down-count
- Input capture.

For the POEGG.INV polarity setting signal, when the same level is input three times continuously with the sampling clock selected in the POEGGn.NFCS[1:0] and POEGGn.NFEN, that value is output. Set the control registers the same as for the input level detection operation described in section 22.3.1, Pin Input Level Detection Operation. The state after filtering can be monitored in POEGGn.ST.

Figure 22.4 shows the output timing of an external trigger to the GPT.

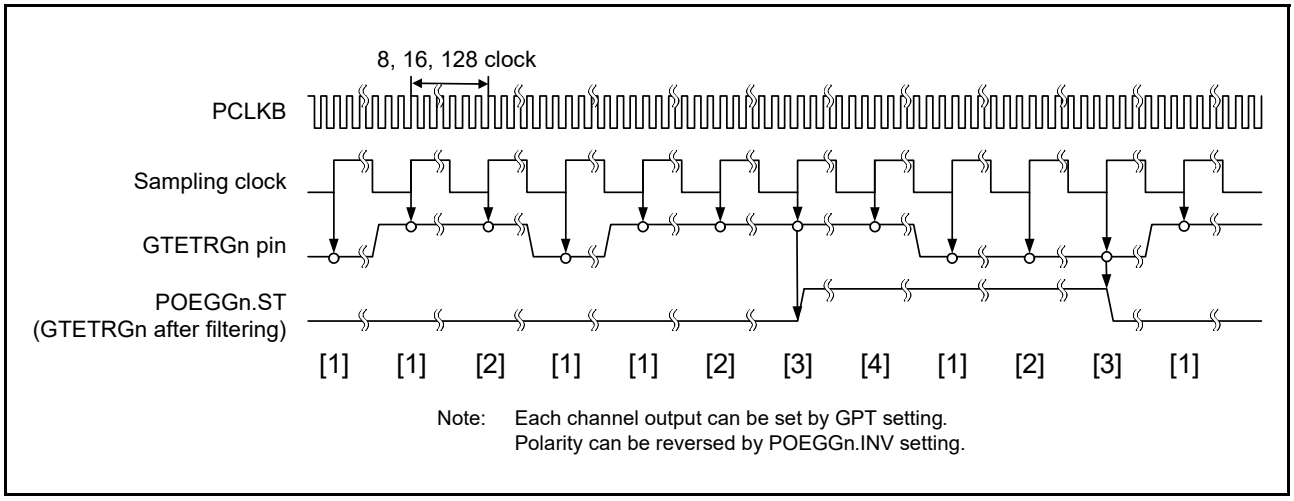


Figure 22.4 Output timing of external trigger to GPT

22.6 Usage Notes

22.6.1 Transition to Software Standby Mode

When using the POEG, do not invoke the Software Standby mode. In this mode, the POEG stops and therefore output disable of the pins cannot be controlled.

22.6.2 Specifying Pins Corresponding to GPT

The POEG controls output disable only when a pin is associated with the GPT in the PmnPFS.PMR and PmnPFS.PSEL settings. When the pin is specified as a general I/O pin, the POEG does not perform output-disable control.

23. General PWM Timer (GPT)

23.1 Overview

The GPT is a 32-bit timer with 10 channels. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up-counter 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.

Table 23.1 lists the GPT specifications, Table 23.2 shows the GPT functions, and Figure 23.1 shows the block diagram.

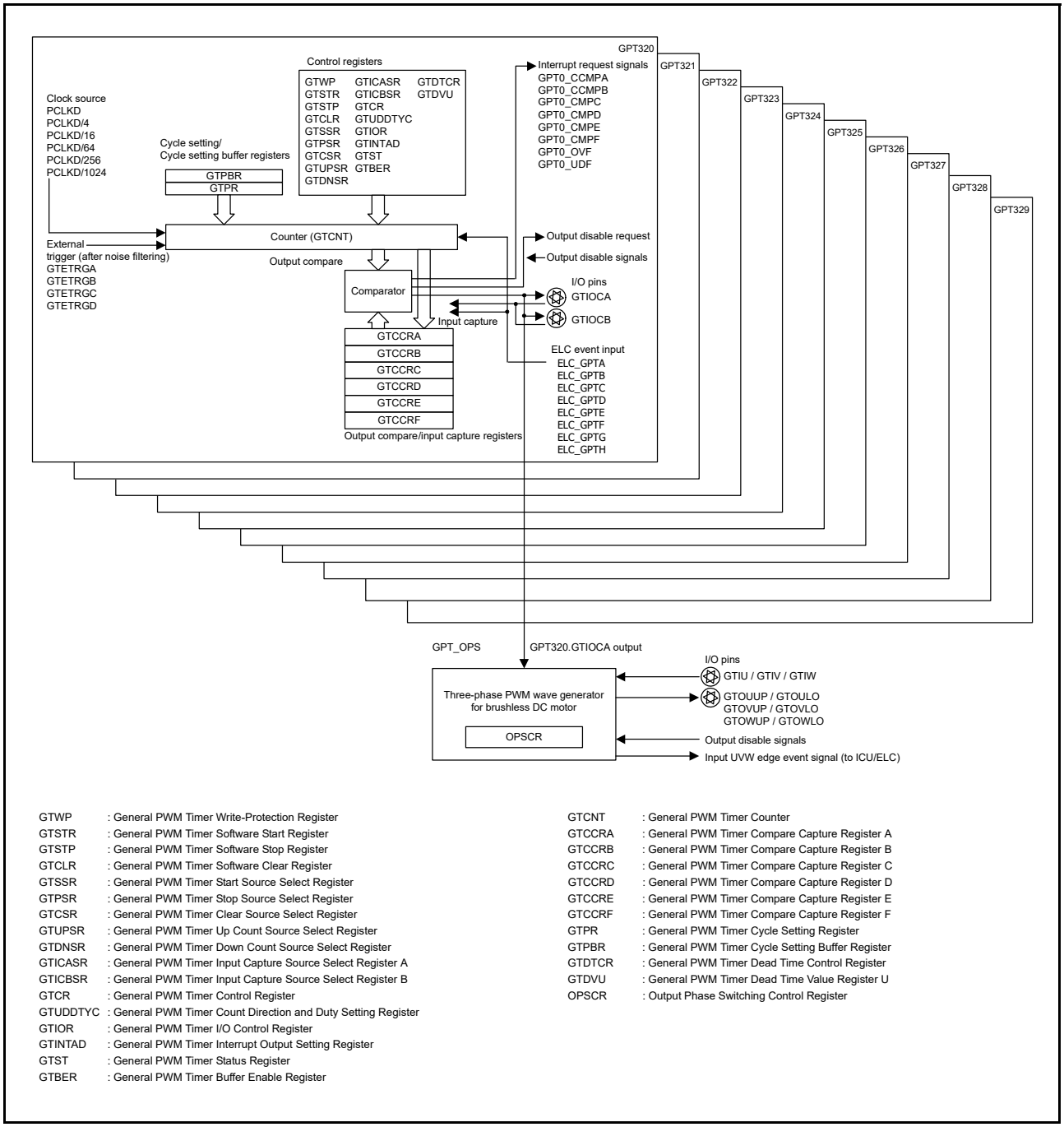
Table 23.1 GPT specifications

Parameter	Description
Functions	• 32 bits × 10 channels • 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 • Synchronous starting, stopping and clearing counters for arbitrary channels • Starting, stopping, clearing and up/down counters in response to a maximum of eight ELC events • Starting, stopping, clearing and up/down counters in response to input level comparison • Starting, clearing, stopping and up/down counters in response to a maximum of four external triggers • Output pin disable function by detected short-circuits between output pins • 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 • Bus clock: PLCKA • Core clock: PCLKD • Frequency ratio: PCLKA:PCLKD = 1:N (N = 1/2/4/8/16/32/64).

Table 23.2 GPT Functions

Parameter		GPT32
Count clock		PCLKD PCLKD/4 PCLKD/16 PCLKD/64 PCLKD/256 PCLKD/1024
Output compare/input capture registers (GTCCR)		GTCCRA GTCCRB
Compare/buffer registers		GTCCRC GTCCRD GTCCRE GTCCRF
Cycle setting register		GTPR
Cycle setting buffer registers		GTPBR
I/O pins		GTIOCA GTIOCB
External trigger input pin*1		GTETRGA GTETRGB GTETRGC GTETRGD
Counter clear sources		GTPR register compare match, input capture, input pin status, ELC event input, and GTETRGn (n = A, B, C, D) pin input
Compare match output	Low output	Available
	High output	Available
	Toggle output	Available
Input capture function		Available
Automatic addition of dead time		Available (no dead time buffer)
PWM mode		Available
Phase count function		Available
Buffer operation		Double buffer
One-shot operation		Available
DTC activation		All the interrupt sources
Brushless DC motor control function		Available
Interrupt sources		8 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) • GTCNT overflow (GTPR compare match) (GPTn_OVF) • GTCNT underflow (GPTn_UDF). Note: n = 0 to 9
Event linking (ELC) function		Available
Noise filtering function		Available

Note 1. GTRETRGn connects to GPT through the POEG module. So, in order to use the GPT function, the POEG clock needs to be supplied by clearing the MSTPD14 bit.



GTWP : General PWM Timer Write-Protection Register
GTSTR : General PWM Timer Software Start Register
GTSTP : General PWM Timer Software Stop Register
GTCLR : General PWM Timer Software Clear Register
GTSSR : General PWM Timer Start Source Select Register
GTPSR : General PWM Timer Stop Source Select Register
GTCSR : General PWM Timer Clear Source Select Register
GTUPSR : General PWM Timer Up Count Source Select Register
GTDNSR : General PWM Timer Down Count Source Select Register
GTICASR : General PWM Timer Input Capture Source Select Register A
GTICBSR : General PWM Timer Input Capture Source Select Register B
GTCR : General PWM Timer Control Register
GTUDDTYC : General PWM Timer Count Direction and Duty Setting Register
GTIOR : General PWM Timer I/O Control Register
GTINTAD : General PWM Timer Interrupt Output Setting Register
GTST : General PWM Timer Status Register
GTBER : General PWM Timer Buffer Enable Register

GTCNT : General PWM Timer Counter
GTCCRA : General PWM Timer Compare Capture Register A
GTCCRB : General PWM Timer Compare Capture Register B
GTCCRC : General PWM Timer Compare Capture Register C
GTCCRD : General PWM Timer Compare Capture Register D
GTCCRE : General PWM Timer Compare Capture Register E
GTCCRF : General PWM Timer Compare Capture Register F
GTPR : General PWM Timer Cycle Setting Register
GTPBR : General PWM Timer Cycle Setting Buffer Register
GTDTCR : General PWM Timer Dead Time Control Register
GTDVU : General PWM Timer Dead Time Value Register U
OPSCR : Output Phase Switching Control Register

Figure 23.1 GPT block diagram

Figure 23.2 shows an example using multiple GPTs.

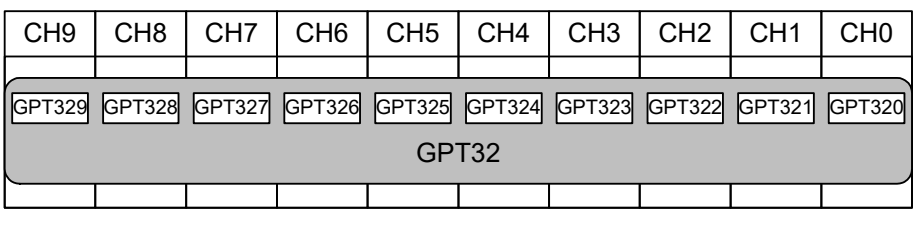


Figure 23.2 Correspondence between GPT channels and module names

Table 23.3 lists the I/O pins used in the GPT.

Table 23.3 I/O pins of GPT

Channel	Pin name	I/O	Function
Common	GTETRGA	Input	External trigger input pin A (after noise filtering)
	GTETRGB	Input	External trigger input pin B (after noise filtering)
	GTETRGC	Input	External trigger input pin C (after noise filtering)
	GTETRGD	Input	External trigger input pin D (after noise filtering)
GPT320	GTIOC0A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC0B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT321	GTIOC1A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC1B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT322	GTIOC2A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC2B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT323	GTIOC3A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC3B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT324	GTIOC4A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC4B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT325	GTIOC5A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC5B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT326	GTIOC6A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC6B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT327	GTIOC7A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC7B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT328	GTIOC8A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC8B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT329	GTIOC9A	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC9B	I/O	GTCCRB register input capture input/output compare output/PWM output pin
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	Three-phase PWM output for BLDC motor control (positive U-phase)
	GTOULO	Output	Three-phase PWM output for BLDC motor control (negative U-phase)
	GTOVUP	Output	Three-phase PWM output for BLDC motor control (positive V-phase)
	GTOVLO	Output	Three-phase PWM output for BLDC motor control (negative V-phase)
	GTOWUP	Output	Three-phase PWM output for BLDC motor control (positive W-phase)
	GTOWLO	Output	Three-phase PWM output for BLDC motor control (negative W-phase)

23.2 Register Descriptions

Table 23.4 lists the registers in the GPT.

Table 23.4 Registers of GPT

Module symbol	Register name	Register symbol	Reset value	Address	Access size
GPT32m	General PWM Timer Write Protection Register	GTWP	00000000h	4007 8000h + 0100h × m	32
GPT32m	General PWM Timer Software Start Register	GTSTR	00000000h	4007 8004h + 0100h × m	32
GPT32m	General PWM Timer Software Stop Register	GTSTP	FFFFFFFFh	4007 8008h + 0100h × m	32
GPT32m	General PWM Timer Software Clear Register	GTCLR	00000000h	4007 800Ch + 0100h × m	32
GPT32m	General PWM Timer Start Source Select Register	GTSSR	00000000h	4007 8010h + 0100h × m	32
GPT32m	General PWM Timer Stop Source Select Register	GTPSR	00000000h	4007 8014h + 0100h × m	32
GPT32m	General PWM Timer Clear Source Select Register	GTCSR	00000000h	4007 8018h + 0100h × m	32
GPT32m	General PWM Timer Up Count Source Select Register	GTUPSR	00000000h	4007 801Ch + 0100h × m	32
GPT32m	General PWM Timer Down Count Source Select Register	GTDNSR	00000000h	4007 8020h + 0100h × m	32
GPT32m	General PWM Timer Input Capture Source Select Register A	GTICASR	00000000h	4007 8024h + 0100h × m	32
GPT32m	General PWM Timer Input Capture Source Select Register B	GTICBSR	00000000h	4007 8028h + 0100h × m	32
GPT32m	General PWM Timer Control Register	GTCR	00000000h	4007 802Ch + 0100h × m	32
GPT32m	General PWM Timer Count Direction and Duty Setting Register	GTUDDTYC	00000001h	4007 8030h + 0100h × m	32
GPT32m	General PWM Timer I/O Control Register	GTIOR	00000000h	4007 8034h + 0100h × m	32
GPT32m	General PWM Timer Interrupt Output Setting Register	GTINTAD	00000000h	4007 8038h + 0100h × m	32
GPT32m	General PWM Timer Status Register	GTST	00008000h	4007 803Ch + 0100h × m	32
GPT32m	General PWM Timer Buffer Enable Register	GTBER	00000000h	4007 8040h + 0100h × m	32
GPT32m	General PWM Timer Counter	GTCNT	00000000h	4007 8048h + 0100h × m	32
GPT32m	General PWM Timer Compare Capture Register A	GTCCRA	FFFFFFFFh	4007 804Ch + 0100h × m	32
GPT32m	General PWM Timer Compare Capture Register B	GTCCRB	FFFFFFFFh	4007 8050h + 0100h × m	32
GPT32m	General PWM Timer Compare Capture Register C	GTCCRC	FFFFFFFFh	4007 8054h + 0100h × m	32
GPT32m	General PWM Timer Compare Capture Register E	GTCCRE	FFFFFFFFh	4007 8058h + 0100h × m	32
GPT32m	General PWM Timer Compare Capture Register D	GTCCRD	FFFFFFFFh	4007 805Ch + 0100h × m	32
GPT32m	General PWM Timer Compare Capture Register F	GTCCRF	FFFFFFFFh	4007 8060h + 0100h × m	32
GPT32m	General PWM Timer Cycle Setting Register	GTPR	FFFFFFFFh	4007 8064h + 0100h × m	32
GPT32m	General PWM Timer Cycle Setting Buffer Register	GTPBR	FFFFFFFFh	4007 8068h + 0100h × m	32
GPT32m	General PWM Timer Dead Time Control Register	GTDTCR	00000000h	4007 8088h + 0100h × m	32
GPT32m	General PWM Timer Dead Time Value Register U	GTDVU	FFFFFFFFh	4007 808Ch + 0100h × m	32
GPT_OPS	Output Phase Switching Control Register	OPSCR	00000000h	4007 8FF0h	32

Note: m = 0 to 9

23.2.1 General PWM Timer Write-Protection Register (GTWP)

Address(es): GPT32m.GTWP 4007 8000h + 0100h × m (m = 0 to 9)

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit name	Description	R/W
b0	WP	Register Write Disable	0: Write to the register enabled 1: Write to the register disabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	GTWP Key Code	When A5h is written to these bits, the WP bits write is permitted. These bits are read as 0.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The GTWP enables or disables writing to registers to prevent accidental modification. The following is a list of write enabled or disabled registers:

GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTIBCSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTDTCR, GTDVU.

23.2.2 General PWM Timer Software Start Register (GTSTR)

Address(es): GPT32m.GTSTR 4007 8004h + 0100h × m (m = 0 to 9)

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CSTRT9	CSTRT8	CSTRT7	CSTRT6	CSTRT5	CSTRT4	CSTRT3	CSTRT2	CSTRT1	CSTRT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The GTSTR starts the GTCNT counter operation for each channel n (n = 0 to 9).

The GTSTR bit number represents the channel number. The GTSTR register of each channel is common. 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. For the association between GTSTR bit number and channel number, see Figure 23.2.

CSTRTn bit (Channel n GTCNT Count Start) (n = 0 to 9)

This bit starts channel n of the GTCNT counter operation. Writing to GTSTR.CSTRTn bit (n = 0 to 9) has no effect unless GPT32m.GTSSR.CSTRT bit is set to 1 (for GPT32, m = 0 to 9).

Read data shows the counter status of each channel (GTCR.CST bit). Zero means the counter stops and 1 means the counter is running.

23.2.3 General PWM Timer Software Stop Register (GTSTP)

Address(es): GPT32m.GTSTP 4007 8008h + 0100h × m (m = 0 to 9)

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CSTOP 9	CSTOP 8	CSTOP 7	CSTOP 6	CSTOP 5	CSTOP 4	CSTOP 3	CSTOP 2	CSTOP 1	CSTOP 0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The GTSTP stops the GTCNT counter operation for each channel n (n = 0 to 9).

The GTSTP bit number represents the channel number. The GTSTP register of each channel is common. 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 GTCNT counter and the value of GTSTP register.

For the association between GTSTP bit number and a channel number, see Figure 23.2.

CSTOPn bit (Channel n GTCNT Count Stop) (n = 0 to 9)

This bit stops channel n of the GTCNT counter operation. Writing to GTSTP.CSTOPn bit (n = 0 to 9) has no effect unless GPT32m.GTPSR.CSTOP bit is set to 1 (for GPT32, m = 0 to 9). Read data shows the counter status of each channel (invert of GTCR.CST bit). Zero means the counter is running and 1 means the counter stops.

23.2.4 General PWM Timer Software Clear Register (GTCLR)

Address(es): GPT32m.GTCLR 4007 800Ch + 0100h × m (m = 0 to 9)

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CCLR 9	CCLR 8	CCLR 7	CCLR 6	CCLR 5	CCLR 4	CCLR 3	CCLR 2	CCLR 1	CCLR 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GTCLR is a write-only register that clears the GTCNT counter operation for each channel n (n = 0 to 9).

The GTCLR bit number represents the channel number. Each channel of the GTCLR register is common. 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 the GTCNT counter. For the association between the GTCLR bit number and a channel number, see Figure 23.2.

CCLRn bit (Channel n GTCNT Count Clear) (n = 0 to 9)

Channel n of the GTCNT counter value is cleared on writing 1 to this bit. This bit is read as 0.

23.2.5 General PWM Timer Start Source Select Register (GTSSR)

Address(es): GPT32m.GTSSR 4007 8010h + 0100h × m (m = 0 to 9)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SSCBFAH	SSCBFAL	SSCBRAH	SSCBRAL	SSCAFBH	SSCAFBL	SSCARBH	SSCARBL	SSGTRGDF	SSGTRGDR	SSGTRGCF	SSGTRGCR	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	Bit name	Description	R/W
b0	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
b1	SSGTRGAF	GTETRGA Pin Falling Input Source Counter Start Enable	0: Counter start disabled at the falling edge of GTETRGA input 1: Counter start enabled at the falling edge of GTETRGA input.	R/W
b2	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
b3	SSGTRGBF	GTETRGB Pin Falling Input Source Counter Start Enable	0: Counter start disabled at the falling edge of GTETRGB input 1: Counter start enabled at the falling edge of GTETRGB input.	R/W
b4	SSGTRGCR	GTETRGC Pin Rising Input Source Counter Start Enable	0: Counter start disabled on the rising edge of GTETRGC input 1: Counter start enabled on the rising edge of GTETRGC input.	R/W
b5	SSGTRGCF	GTETRGC Pin Falling Input Source Counter Start Enable	0: Counter start disabled at the falling edge of GTETRGC input 1: Counter start enabled at the falling edge of GTETRGC input.	R/W
b6	SSGTRGDR	GTETRGD Pin Rising Input Source Counter Start Enable	0: Counter start disabled on the rising edge of GTETRGD input 1: Counter start enabled on the rising edge of GTETRGD input.	R/W
b7	SSGTRGDF	GTETRGD Pin Falling Input Source Counter Start Enable	0: Counter start disabled at the falling edge of GTETRGD input 1: Counter start enabled at the falling edge of GTETRGD input.	R/W
b8	SSCARBL	GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Start Enable	0: Counter start disabled on the rising edge of GTIOCA input when GTIOCB input is 0 1: Counter start enabled on the rising edge of GTIOCA input when GTIOCB input is 0.	R/W
b9	SSCARBH	GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Start Enable	0: Counter start disabled on the rising edge of GTIOCA input when GTIOCB input is 1 1: Counter start enabled on the rising edge of GTIOCA input when GTIOCB input is 1.	R/W
b10	SSCAFBL	GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Start Enable	0: Counter start disabled at the falling edge of GTIOCA input when GTIOCB input is 0 1: Counter start enabled at the falling edge of GTIOCA input when GTIOCB input is 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b11	SSCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Start Enable	0: Counter start disabled at the falling edge of GTIOCA input when GTIOCB input is 1 1: Counter start enabled at the falling edge of GTIOCA input when GTIOCB input is 1.	R/W
b12	SSCBRAL	GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Start Enable	0: Counter start disabled on the rising edge of GTIOCB input when GTIOCA input is 0 1: Counter start enabled on the rising edge of GTIOCB input when GTIOCA input is 0.	R/W
b13	SSCBRAH	GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Start Enable	0: Counter start disabled on the rising edge of GTIOCB input when GTIOCA input is 1 1: Counter start enabled on the rising edge of GTIOCB input when GTIOCA input is 1.	R/W
b14	SSCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Start Enable	0: Counter start disabled at the falling edge of GTIOCB input when GTIOCA input is 0 1: Counter start enabled at the falling edge of GTIOCB input when GTIOCA input is 0.	R/W
b15	SSCBFAH	GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Start Enable	0: Counter start disabled at the falling edge of GTIOCB input when GTIOCA input is 1 1: Counter start enabled at the falling edge of GTIOCB input when GTIOCA input is 1.	R/W
b16	SSELCA	ELC_GPTA Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTA event input 1: Counter start enabled at the ELC_GPTA event input.	R/W
b17	SSELCB	ELC_GPTB Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTB event input 1: Counter start enabled at the ELC_GPTB event input.	R/W
b18	SSELCC	ELC_GPTC Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTC event input 1: Counter start enabled at the ELC_GPTC event input.	R/W
b19	SSELCD	ELC_GPTD Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTD event input 1: Counter start enabled at the ELC_GPTD event input.	R/W
b20	SSELCE	ELC_GPTE Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTE event input 1: Counter start enabled at the ELC_GPTE event input.	R/W
b21	SSELCF	ELC_GPTF Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTF event input 1: Counter start enabled at the ELC_GPTF event input.	R/W
b22	SSELCG	ELC_GPTG Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTG event input 1: Counter start enabled at the ELC_GPTG event input.	R/W
b23	SSELCH	ELC_GPTH Event Source Counter Start Enable	0: Counter start disabled at the ELC_GPTH event input 1: Counter start enabled at the ELC_GPTH event input.	R/W
b30 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	CSTRT	Software Source Counter Start Enable	0: Counter start disabled by the GTSTR register 1: Counter start enabled by the GTSTR register.	R/W

GTSSR sets the source to start the GTCNT counter.

SSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Start Enable)

This bit enables or disables GTCNT counter start on the rising edge of GTETRGA pin input.

SSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Start Enable)

This bit enables or disables GTCNT counter start at the falling edge of GTETRGA pin input.

SSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Start Enable)

This bit enables or disables GTCNT counter start on the rising edge of GTETRGB pin input.

SSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Start Enable)

This bit enables or disables GTCNT counter start at the falling edge of GTETRGB pin input.

SSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Start Enable)

This bit enables or disables GTCNT counter start on the rising edge of GTETRGC pin input.

SSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Start Enable)

This bit enables or disables GTCNT counter start at the falling edge of GTETRGC pin input.

SSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Start Enable)

This bit enables or disables GTCNT counter start on the rising edge of GTETRGD pin input.

SSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Start Enable)

This bit enables or disables GTCNT counter start at the falling edge of GTETRGD pin input.

SSCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Start Enable)

This bit enables or disables GTCNT counter start on the rising edge of GTIOCA pin input, when GTIOCB input is 0.

SSCARBH bit (GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Start Enable)

This bit enables or disables GTCNT counter start on the rising edge of GTIOCA pin input, when GTIOCB input is 1.

SSCAFBL bit (GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Start Enable)

This bit enables or disables GTCNT counter start at the falling edge of GTIOCA pin input, when GTIOCB input is 0.

SSCAFBH bit (GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Start Enable)

This bit enables or disables GTCNT counter start at the falling edge of GTIOCA pin input, when GTIOCB input is 1.

SSCBRAL bit (GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Start Enable)

This bit enables or disables GTCNT counter start on the rising edge of GTIOCB pin input, when GTIOCA input is 0.

SSCBRAH bit (GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Start Enable)

This bit enables or disables GTCNT counter start on the rising edge of GTIOCB pin input, when GTIOCA input is 1.

SSCBFAL bit (GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Start Enable)

This bit enables or disables GTCNT counter start at the falling edge of GTIOCB pin input, when GTIOCA input is 0.

SSCBFAH bit (GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Start Enable)

This bit enables or disables GTCNT counter start at the falling edge of GTIOCB pin input, when GTIOCA input is 1.

SSELCm bit (ELC_GPTm Event Source Counter Start Enable) (m = A to H)

This bit enables or disables GTCNT counter start at the ELC_GPTm event input.

CSTRT bit (Software Source Counter Start Enable)

This bit enables or disables GTCNT counter start by GTSTR register.

23.2.6 General PWM Timer Stop Source Select Register (GTPSR)

Address(es): GPT32m.GTPSR 4007 8014h + 0100h × m (m = 0 to 9)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PSCBFAH	PSCBFAL	PSCBRAH	PSCBRAL	PSCAFBH	PSCAFBL	PSCARBH	PSCARBL	PSGTRGDF	PSGTRGDR	PSGTRGCF	PSGTRGCR	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	Bit name	Description	R/W
b0	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
b1	PSGTRGAF	GTETRGA Pin Falling Input Source Counter Stop Enable	0: Counter stop disabled at the falling edge of GTETRGA input 1: Counter stop enabled at the falling edge of GTETRGA input.	R/W
b2	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
b3	PSGTRGBF	GTETRGB Pin Falling Input Source Counter Stop Enable	0: Counter stop disabled at the falling edge of GTETRGB input 1: Counter stop enabled at the falling edge of GTETRGB input.	R/W
b4	PSGTRGCR	GTETRGC Pin Rising Input Source Counter Stop Enable	0: Counter stop disabled on the rising edge of GTETRGC input 1: Counter stop enabled on the rising edge of GTETRGC input.	R/W
b5	PSGTRGCF	GTETRGC Pin Falling Input Source Counter Stop Enable	0: Counter stop disabled at the falling edge of GTETRGC input 1: Counter stop enabled at the falling edge of GTETRGC input.	R/W
b6	PSGTRGDR	GTETRGD Pin Rising Input Source Counter Stop Enable	0: Counter stop disabled on the rising edge of GTETRGD input 1: Counter stop enabled on the rising edge of GTETRGD input.	R/W
b7	PSGTRGDF	GTETRGD Pin Falling Input Source Counter Stop Enable	0: Counter stop disabled at the falling edge of GTETRGD input 1: Counter stop enabled at the falling edge of GTETRGD input.	R/W
b8	PSCARBL	GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Stop Enable	0: Counter stop disabled on the rising edge of GTIOCA input when GTIOCB input is 0 1: Counter stop enabled on the rising edge of GTIOCA input when GTIOCB input is 0.	R/W
b9	PSCARBH	GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Stop Enable	0: Counter stop disabled on the rising edge of GTIOCA input when GTIOCB input is 1 1: Counter stop enabled on the rising edge of GTIOCA input when GTIOCB input is 1.	R/W
b10	PSCAFBL	GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Stop Enable	0: Counter stop disabled at the falling edge of GTIOCA input when GTIOCB input is 0 1: Counter stop enabled at the falling edge of GTIOCA input when GTIOCB input is 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b11	PSCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Stop Enable	0: Counter stop disabled at the falling edge of GTIOCA input when GTIOCB input is 1 1: Counter stop enabled at the falling edge of GTIOCA input when GTIOCB input is 1.	R/W
b12	PSCBRAL	GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Stop Enable	0: Counter stop disabled on the rising edge of GTIOCB input when GTIOCA input is 0 1: Counter stop enabled on the rising edge of GTIOCB input when GTIOCA input is 0.	R/W
b13	PSCBRAH	GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Stop Enable	0: Counter stop disabled on the rising edge of GTIOCB input when GTIOCA input is 1 1: Counter stop enabled on the rising edge of GTIOCB input when GTIOCA input is 1.	R/W
b14	PSCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Stop Enable	0: Counter stop disabled at the falling edge of GTIOCB input when GTIOCA input is 0 1: Counter stop enabled at the falling edge of GTIOCB input when GTIOCA input is 0.	R/W
b15	PSCBFAH	GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Stop Enable	0: Counter stop disabled at the falling edge of GTIOCB input when GTIOCA input is 1 1: Counter stop enabled at the falling edge of GTIOCB input when GTIOCA input is 1.	R/W
b16	PSELCA	ELC_GPTA Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTA event input 1: Counter stop enabled at the ELC_GPTA event input.	R/W
b17	PSELCB	ELC_GPTB Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTB event input 1: Counter stop enabled at the ELC_GPTB event input.	R/W
b18	PSELCC	ELC_GPTC Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTC event input 1: Counter stop enabled at the ELC_GPTC event input.	R/W
b19	PSELCD	ELC_GPTD Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTD event input 1: Counter stop enabled at the ELC_GPTD event input.	R/W
b20	PSELCE	ELC_GPTE Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTE event input 1: Counter stop enabled at the ELC_GPTE event input.	R/W
b21	PSELCF	ELC_GPTF Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTF event input 1: Counter stop enabled at the ELC_GPTF event input.	R/W
b22	PSELCG	ELC_GPTG Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTG event input 1: Counter stop enabled at the ELC_GPTG event input.	R/W
b23	PSELCH	ELC_GPTH Event Source Counter Stop Enable	0: Counter stop disabled at the ELC_GPTH event input 1: Counter stop enabled at the ELC_GPTH event input.	R/W
b30 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	CSTOP	Software Source Counter Stop Enable	0: Counter stop disabled by the GTSTP register 1: Counter stop enabled by the GTSTP register.	R/W

GTPSR sets the source to stop the GTCNT counter.

PSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop on the rising edge of GTETRGA pin input.

PSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop at the falling edge of GTETRGA pin input.

PSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop on the rising edge of GTETRGB pin input.

PSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop at the falling edge of GTETRGB pin input.

PSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop on the rising edge of GTETRGC pin input.

PSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop at the falling edge of GTETRGC pin input.

PSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop on the rising edge of GTETRGD pin input.

PSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop at the falling edge of GTETRGD pin input.

PSCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop on the rising edge of GTIOCA pin input, when GTIOCB input is 0.

PSCARBH bit (GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop on the rising edge of GTIOCA pin input, when GTIOCB input is 1.

PSCAFBL bit (GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop at the falling edge of GTIOCA pin input, when GTIOCB input is 0.

PSCAFBH bit (GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop at the falling edge of GTIOCA pin input, when GTIOCB input is 1.

PSCBRAL bit (GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop on the rising edge of GTIOCB pin input, when GTIOCA input is 0.

PSCBRAH bit (GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop on the rising edge of GTIOCB pin input, when GTIOCA input is 1.

PSCBFAL bit (GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop at the falling edge of GTIOCB pin input, when GTIOCA input is 0.

PSCBFAH bit (GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop at the falling edge of GTIOCB pin input, when GTIOCA input is 1.

PSELCm bit (ELC_GPTm Event Source Counter Stop Enable) (m = A to H)

This bit enables or disables GTCNT counter stop at the ELC_GPTm event input.

CSTOP bit (Software Source Counter Stop Enable)

This bit enables or disables GTCNT counter stop by GTSTP register.

23.2.7 General PWM Timer Clear Source Select Register (GTCSR)

Address(es): GPT32m.GTCSR 4007 8018h + 0100h × m (m = 0 to 9)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CCLR	—	—	—	—	—	—	—	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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CSCBFAH	CSCBFAL	CSCBRAH	CSCBRAL	CSCAFBH	CSCAFBL	CSCARBH	CSCARBL	CSGTRGDF	CSGTRGDR	CSGTRGCF	CSGTRGCR	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	Bit name	Description	R/W
b0	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
b1	CSGTRGAF	GTETRGA Pin Falling Input Source Counter Clear Enable	0: Counter clear disabled at the falling edge of GTETRGA input 1: Counter clear enabled at the falling edge of GTETRGA input.	R/W
b2	CSGTRGBR	GTETRGB Pin Rising Input Source Counter Clear Enable	0: Counter clear disabled on the rising edge of GTETRGB input 1: Counter clear enabled on the rising edge of GTETRGB input.	R/W
b3	CSGTRGBF	GTETRGB Pin Falling Input Source Counter Clear Enable	0: Counter clear disabled at the falling edge of GTETRGB input 1: Counter clear enabled at the falling edge of GTETRGB input.	R/W
b4	CSGTRGCR	GTETRGC Pin Rising Input Source Counter Clear Enable	0: Counter clear disabled on the rising edge of GTETRGC input 1: Counter clear enabled on the rising edge of GTETRGC input.	R/W
b5	CSGTRGCF	GTETRGC Pin Falling Input Source Counter Clear Enable	0: Counter clear disabled at the falling edge of GTETRGC input 1: Counter clear enabled at the falling edge of GTETRGC input.	R/W
b6	CSGTRGDR	GTETRGD Pin Rising Input Source Counter Clear Enable	0: Counter clear disabled on the rising edge of GTETRGD input 1: Counter clear enabled on the rising edge of GTETRGD input.	R/W
b7	CSGTRGDF	GTETRGD Pin Falling Input Source Counter Clear Enable	0: Counter clear disabled at the falling edge of GTETRGD input 1: Counter clear enabled at the falling edge of GTETRGD input.	R/W
b8	CSCARBL	GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Clear Enable	0: Counter clear disabled on the rising edge of GTIOCA input when GTIOCB input is 0 1: Counter clear enabled on the rising edge of GTIOCA input when GTIOCB input is 0.	R/W
b9	CSCARBH	GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Clear Enable	0: Counter clear disabled on the rising edge of GTIOCA input when GTIOCB input is 1 1: Counter clear enabled on the rising edge of GTIOCA input when GTIOCB input is 1.	R/W
b10	CSCAFBL	GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Clear Enable	0: Counter clear disabled at the falling edge of GTIOCA input when GTIOCB input is 0 1: Counter clear enabled at the falling edge of GTIOCA input when GTIOCB input is 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b11	CSCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Clear Enable	0: Counter clear disabled at the falling edge of GTIOCA input when GTIOCB input is 1 1: Counter clear enabled at the falling edge of GTIOCA input when GTIOCB input is 1.	R/W
b12	CSCBRAL	GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Clear Enable	0: Counter clear disabled on the rising edge of GTIOCB input when GTIOCA input is 0 1: Counter clear enabled on the rising edge of GTIOCB input when GTIOCA input is 0.	R/W
b13	CSCBRAH	GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Clear Enable	0: Counter clear disabled on the rising edge of GTIOCB input when GTIOCA input is 1 1: Counter clear enabled on the rising edge of GTIOCB input when GTIOCA input is 1.	R/W
b14	CSCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Clear Enable	0: Counter clear disabled at the falling edge of GTIOCB input when GTIOCA input is 0 1: Counter clear enabled at the falling edge of GTIOCB input when GTIOCA input is 0.	R/W
b15	CSCBFAH	GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Clear Enable	0: Counter clear disabled at the falling edge of GTIOCB input when GTIOCA input is 1 1: Counter clear enabled at the falling edge of GTIOCB input when GTIOCA input is 1.	R/W
b16	CSELCA	ELC_GPTA Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTA event input 1: Counter clear enabled at the ELC_GPTA event input.	R/W
b17	CSELCB	ELC_GPTB Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTB event input 1: Counter clear enabled at the ELC_GPTB event input.	R/W
b18	CSELCC	ELC_GPTC Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTC event input 1: Counter clear enabled at the ELC_GPTC event input.	R/W
b19	CSELCD	ELC_GPTD Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTD event input 1: Counter clear enabled at the ELC_GPTD event input.	R/W
b20	CSELCE	ELC_GPTE Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTE event input 1: Counter clear enabled at the ELC_GPTE event input.	R/W
b21	CSELCF	ELC_GPTF Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTF event input 1: Counter clear enabled at the ELC_GPTF event input.	R/W
b22	CSELCG	ELC_GPTG Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTG event input 1: Counter clear enabled at the ELC_GPTG event input.	R/W
b23	CSELCH	ELC_GPTH Event Source Counter Clear Enable	0: Counter clear disabled at the ELC_GPTH event input 1: Counter clear enabled at the ELC_GPTH event input.	R/W
b30 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	CCLR	Software Source Counter Clear Enable	0: Counter clear disabled by the GTCLR register 1: Counter clear enabled by the GTCLR register.	R/W

GTCSR sets the source to clear the GTCNT counter.

CSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear on the rising edge of GTETRGA pin input.

CSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear at the falling edge of GTETRGA pin input.

CSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear on the rising edge of GTETRGB pin input.

CSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear at the falling edge of GTETRGB pin input.

CSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear on the rising edge of GTETRGC pin input.

CSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear at the falling edge of GTETRGC pin input.

CSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear on the rising edge of GTETRGD pin input.

CSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear at the falling edge of GTETRGD pin input.

CSCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear on the rising edge of GTIOCA pin input, when GTIOCB input is 0.

CSCARBH bit (GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear on the rising edge of GTIOCA pin input, when GTIOCB input is 1.

CSCAFBL bit (GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear at the falling edge of GTIOCA pin input, when GTIOCB input is 0.

CSCAFBH bit (GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear at the falling edge of GTIOCA pin input, when GTIOCB input is 1.

CSCBRAL bit (GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear on the rising edge of GTIOCB pin input, when GTIOCA input is 0.

CSCBRAH bit (GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear on the rising edge of GTIOCB pin input, when GTIOCA input is 1.

CSCBFAL bit (GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear at the falling edge of GTIOCB pin input, when GTIOCA input is 0.

CSCBFAH bit (GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear at the falling edge of GTIOCB pin input, when GTIOCA input is 1.

CSELCm bit (ELC_GPTm Event Source Counter Clear Enable) (m = A to H)

This bit enables or disables GTCNT counter clear at the ELC_GPTm event input.

CCLR bit (Software Source Counter Clear Enable)

This bit enables or disables GTCNT counter clear by GTCLR register.

23.2.8 General PWM Timer Up Count Source Select Register (GTUPSR)

Address(es): GPT32m.GTUPSR 4007 801Ch + 0100h × m (m = 0 to 9)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	USCBFAH	USCBFAL	USCBRAH	USCBRAL	USCAFBH	USCAFBL	USCARBH	USCARBL	USGTRGDF	USGTRGDR	USGTRGCF	USGTRGCR	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	Bit name	Description	R/W
b0	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
b1	USGTRGAF	GTETRGA Pin Falling Input Source Counter Count Up Enable	0: Counter count up disabled at the falling edge of GTETRGA input 1: Counter count up enabled at the falling edge of GTETRGA input.	R/W
b2	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
b3	USGTRGBF	GTETRGB Pin Falling Input Source Counter Count Up Enable	0: Counter count up disabled at the falling edge of GTETRGB input 1: Counter count up enabled at the falling edge of GTETRGB input.	R/W
b4	USGTRGCR	GTETRGC Pin Rising Input Source Counter Count Up Enable	0: Counter count up disabled on the rising edge of GTETRGC input 1: Counter count up enabled on the rising edge of GTETRGC input.	R/W
b5	USGTRGCF	GTETRGC Pin Falling Input Source Counter Count Up Enable	0: Counter count up disabled at the falling edge of GTETRGC input 1: Counter count up enabled at the falling edge of GTETRGC input.	R/W
b6	USGTRGDR	GTETRGD Pin Rising Input Source Counter Count Up Enable	0: Counter count up disabled on the rising edge of GTETRGD input 1: Counter count up enabled on the rising edge of GTETRGD input.	R/W
b7	USGTRGDF	GTETRGD Pin Falling Input Source Counter Count Up Enable	0: Counter count up disabled at the falling edge of GTETRGD input 1: Counter count up enabled at the falling edge of GTETRGD input.	R/W
b8	USCARBL	GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Count Up Enable	0: Counter count up disabled on the rising edge of GTIOCA input when GTIOCB input is 0 1: Counter count up enabled on the rising edge of GTIOCA input when GTIOCB input is 0.	R/W
b9	USCARBH	GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Count Up Enable	0: Counter count up disabled on the rising edge of GTIOCA input when GTIOCB input is 1 1: Counter count up enabled on the rising edge of GTIOCA input when GTIOCB input is 1.	R/W
b10	USCAFBL	GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Count Up Enable	0: Counter count up disabled at the falling edge of GTIOCA input when GTIOCB input is 0 1: Counter count up enabled at the falling edge of GTIOCA input when GTIOCB input is 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b11	USCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Count Up Enable	0: Counter count up disabled at the falling edge of GTIOCA input when GTIOCB input is 1 1: Counter count up enabled at the falling edge of GTIOCA input when GTIOCB input is 1.	R/W
b12	USCBRAL	GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Count Up Enable	0: Counter count up disabled on the rising edge of GTIOCB input when GTIOCA input is 0 1: Counter count up enabled on the rising edge of GTIOCB input when GTIOCA input is 0.	R/W
b13	USCBRAH	GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Count Up Enable	0: Counter count up disabled on the rising edge of GTIOCB input when GTIOCA input is 1 1: Counter count up enabled on the rising edge of GTIOCB input when GTIOCA input is 1.	R/W
b14	USCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Count Up Enable	0: Counter count up disabled at the falling edge of GTIOCB input when GTIOCA input is 0 1: Counter count up enabled at the falling edge of GTIOCB input when GTIOCA input is 0.	R/W
b15	USCBFAH	GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Count Up Enable	0: Counter count up disabled at the falling edge of GTIOCB input when GTIOCA input is 1 1: Counter count up enabled at the falling edge of GTIOCB input when GTIOCA input is 1.	R/W
b16	USELCA	ELC_GPTA Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTA event input 1: Counter count up enabled at the ELC_GPTA event input.	R/W
b17	USELCB	ELC_GPTB Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTB event input 1: Counter count up enabled at the ELC_GPTB event input.	R/W
b18	USELCC	ELC_GPTC Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTC event input 1: Counter count up enabled at the ELC_GPTC event input.	R/W
b19	USELCD	ELC_GPTD Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTD event input 1: Counter count up enabled at the ELC_GPTD event input.	R/W
b20	USELCE	ELC_GPTE Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTE event input 1: Counter count up enabled at the ELC_GPTE event input.	R/W
b21	USELCF	ELC_GPTF Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTF event input 1: Counter count up enabled at the ELC_GPTF event input	R/W
b22	USELCG	ELC_GPTG Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTG event input 1: Counter count up enabled at the ELC_GPTG event input	R/W
b23	USELCH	ELC_GPTH Event Source Counter Count Up Enable	0: Counter count up disabled at the ELC_GPTH event input 1: Counter count up enabled at the ELC_GPTH event input	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

GTUPSR sets the source to count up the GTCNT counter.

When at least 1 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, but the GTCNT counter set by GTCR.TPCS does not perform the count.

USGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Up Enable)

This bit enables or disables GTCNT counter count up on the rising edge of GTETRGA pin input.

USGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Up Enable)

This bit enables or disables GTCNT counter count up at the falling edge of GTETRGA pin input.

USGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Up Enable)

This bit enables or disables GTCNT counter count up on the rising edge of GTETRGB pin input.

USGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Up Enable)

This bit enables or disables GTCNT counter count up at the falling edge of GTETRGB pin input.

USGTRGCR bit (GTETRGC Pin Rising Input Source Counter Count Up Enable)

This bit enables or disables GTCNT counter count up on the rising edge of GTETRGC pin input.

USGTRGCF bit (GTETRGC Pin Falling Input Source Counter Count Up Enable)

This bit enables or disables GTCNT counter count up at the falling edge of GTETRGC pin input.

USGTRGDR bit (GTETRGD Pin Rising Input Source Counter Count Up Enable)

This bit enables or disables GTCNT counter count up on the rising edge of GTETRGD pin input.

USGTRGDF bit (GTETRGD Pin Falling Input Source Counter Count Up Enable)

This bit enables or disables GTCNT counter count up at the falling edge of GTETRGD pin input.

USCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Count Up Enable)

This bit enables or disables GTCNT counter count up on the rising edge of GTIOCA pin input, when GTIOCB input is 0.

USCARBH bit (GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Count Up Enable)

This bit enables or disables GTCNT counter count up on the rising edge of GTIOCA pin input, when GTIOCB input is 1.

USCAFBL bit (GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Count Up Enable)

This bit enables or disables GTCNT counter count up at the falling edge of GTIOCA pin input, when GTIOCB input is 0.

USCAFBH bit (GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Count Up Enable)

This bit enables or disables GTCNT counter count up at the falling edge of GTIOCA pin input, when GTIOCB input is 1.

USCBRAL bit (GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Count Up Enable)

This bit enables or disables GTCNT counter count up on the rising edge of GTIOCB pin input, when GTIOCA input is 0.

USCBRAH bit (GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Count Up Enable)

This bit enables or disables GTCNT counter count up on the rising edge of GTIOCB pin input, when GTIOCA input is 1.

USCBFAL bit (GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Count Up Enable)

This bit enables or disables GTCNT counter count up at the falling edge of GTIOCB pin input, when GTIOCA input is 0.

USCBFAH bit (GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Count Up Enable)

This bit enables or disables GTCNT counter count up at the falling edge of GTIOCB pin input, when GTIOCA input is 1.

USELCm bit (ELC_GPTm Event Source Counter Count Up Enable) (m = A to H)

This bit enables or disables GTCNT counter count up at the ELC_GPTm event input.

23.2.9 General PWM Timer Down Count Source Select Register (GTDNSR)

Address(es): GPT32m.GTDNSR 4007 8020h + 0100h × m (m = 0 to 9)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DSCBFAH	DSCBFAL	DSCBRAH	DSCBRAL	DSCAFBH	DSCAFBL	DSCARBH	DSCARBL	DSGTRGDF	DSGTRGDR	DSGTRGCF	DSGTRGCR	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	Bit name	Description	R/W
b0	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
b1	DSGTRGAF	GTETRGA Pin Falling Input Source Counter Count Down Enable	0: Counter count down disabled at the falling edge of GTETRGA input 1: Counter count down enabled at the falling edge of GTETRGA input.	R/W
b2	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
b3	DSGTRGBF	GTETRGB Pin Falling Input Source Counter Count Down Enable	0: Counter count down disabled at the falling edge of GTETRGB input 1: Counter count down enabled at the falling edge of GTETRGB input.	R/W
b4	DSGTRGCR	GTETRGC Pin Rising Input Source Counter Count Down Enable	0: Counter count down disabled on the rising edge of GTETRGC input 1: Counter count down enabled on the rising edge of GTETRGC input.	R/W
b5	DSGTRGCF	GTETRGC Pin Falling Input Source Counter Count Down Enable	0: Counter count down disabled at the falling edge of GTETRGC input 1: Counter count down enabled at the falling edge of GTETRGC input.	R/W
b6	DSGTRGDR	GTETRGD Pin Rising Input Source Counter Count Down Enable	0: Counter count down disabled on the rising edge of GTETRGD input 1: Counter count down enabled on the rising edge of GTETRGD input.	R/W
b7	DSGTRGDF	GTETRGD Pin Falling Input Source Counter Count Down Enable	0: Counter count down disabled at the falling edge of GTETRGD input 1: Counter count down enabled at the falling edge of GTETRGD input.	R/W
b8	DSCARBL	GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Count Down Enable	0: Counter count down disabled on the rising edge of GTIOCA input when GTIOCB input is 0 1: Counter count down enabled on the rising edge of GTIOCA input when GTIOCB input is 0.	R/W
b9	DSCARBH	GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Count Down Enable	0: Counter count down disabled on the rising edge of GTIOCA input when GTIOCB input is 1 1: Counter count down enabled on the rising edge of GTIOCA input when GTIOCB input is 1.	R/W
b10	DSCAFBL	GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Count Down Enable	0: Counter count down disabled at the falling edge of GTIOCA input when GTIOCB input is 0 1: Counter count down enabled at the falling edge of GTIOCA input when GTIOCB input is 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b11	DSCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Count Down Enable	0: Counter count down disabled at the falling edge of GTIOCA input when GTIOCB input is 1 1: Counter count down enabled at the falling edge of GTIOCA input when GTIOCB input is 1.	R/W
b12	DSCBRAL	GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Count Down Enable	0: Counter count down disabled on the rising edge of GTIOCB input when GTIOCA input is 0 1: Counter count down enabled on the rising edge of GTIOCB input when GTIOCA input is 0.	R/W
b13	DSCBRAH	GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Count Down Enable	0: Counter count down disabled on the rising edge of GTIOCB input when GTIOCA input is 1 1: Counter count down enabled on the rising edge of GTIOCB input when GTIOCA input is 1.	R/W
b14	DSCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Count Down Enable	0: Counter count down disabled at the falling edge of GTIOCB input when GTIOCA input is 0 1: Counter count down enabled at the falling edge of GTIOCB input when GTIOCA input is 0.	R/W
b15	DSCBFAH	GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Count Down Enable	0: Counter count down disabled at the falling edge of GTIOCB input when GTIOCA input is 1 1: Counter count down enabled at the falling edge of GTIOCB input when GTIOCA input is 1.	R/W
b16	DSELCA	ELC_GPTA Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTA event input 1: Counter count down enabled at the ELC_GPTA event input.	R/W
b17	DSELCB	ELC_GPTB Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTB event input 1: Counter count down enabled at the ELC_GPTB event input.	R/W
b18	DSELCC	ELC_GPTC Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTC event input 1: Counter count down enabled at the ELC_GPTC event input.	R/W
b19	DSELCD	ELC_GPTD Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTD event input 1: Counter count down enabled at the ELC_GPTD event input.	R/W
b20	DSELCE	ELC_GPTE Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTE event input 1: Counter count down enabled at the ELC_GPTE event input.	R/W
b21	DSELCF	ELC_GPTF Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTF event input 1: Counter count down enabled at the ELC_GPTF event input.	R/W
b22	DSELCG	ELC_GPTG Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTG event input 1: Counter count down enabled at the ELC_GPTG event input.	R/W
b23	DSELCH	ELC_GPTH Event Source Counter Count Down Enable	0: Counter count down disabled at the ELC_GPTH event input 1: Counter count down enabled at the ELC_GPTH event input.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

GTDNSR sets the source to count down the GTCNT counter.

When at least 1 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, but the GTCNT counter set by GTCR.TPCS does not perform the count.

DSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Down Enable)

This bit enables or disables GTCNT counter count down on the rising edge of GTETRGA pin input.

DSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Down Enable)

This bit enables or disables GTCNT counter count down at the falling edge of GTETRGA pin input.

DSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Down Enable)

This bit enables or disables GTCNT counter count down on the rising edge of GTETRGB pin input.

DSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Down Enable)

This bit enables or disables GTCNT counter count down at the falling edge of GTETRGB pin input.

DSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Count Down Enable)

This bit enables or disables GTCNT counter count down on the rising edge of GTETRGC pin input.

DSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Count Down Enable)

This bit enables or disables GTCNT counter count down at the falling edge of GTETRGC pin input.

DSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Count Down Enable)

This bit enables or disables GTCNT counter count down on the rising edge of GTETRGD pin input.

DSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Count Down Enable)

This bit enables or disables GTCNT counter count down at the falling edge of GTETRGD pin input.

DSCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source Counter Count Down Enable)

This bit enables or disables GTCNT counter count down on the rising edge of GTIOCA pin input, when GTIOCB input is 0.

DSCARBH bit (GTIOCA Pin Rising Input during GTIOCB Value High Source Counter Count Down Enable)

This bit enables or disables GTCNT counter count down on the rising edge of GTIOCA pin input, when GTIOCB input is 1.

DSCAFBL bit (GTIOCA Pin Falling Input during GTIOCB Value Low Source Counter Count Down Enable)

This bit enables or disables GTCNT counter count down at the falling edge of GTIOCA pin input, when GTIOCB input is 0.

DSCAFBH bit (GTIOCA Pin Falling Input during GTIOCB Value High Source Counter Count Down Enable)

This bit enables or disables GTCNT counter count down at the falling edge of GTIOCA pin input, when GTIOCB input is 1.

DSCBRAL bit (GTIOCB Pin Rising Input during GTIOCA Value Low Source Counter Count Down Enable)

This bit enables or disables GTCNT counter count down on the rising edge of GTIOCB pin input, when GTIOCA input is 0.

DSCBRAH bit (GTIOCB Pin Rising Input during GTIOCA Value High Source Counter Count Down Enable)

This bit enables or disables GTCNT counter count down on the rising edge of GTIOCB pin input, when GTIOCA input is 1.

DSCBFAL bit (GTIOCB Pin Falling Input during GTIOCA Value Low Source Counter Count Down Enable)

This bit enables or disables GTCNT counter count down at the falling edge of GTIOCB pin input, when GTIOCA input is 0.

DSCBFAH bit (GTIOCB Pin Falling Input during GTIOCA Value High Source Counter Count Down Enable)

This bit enables or disables GTCNT counter count down at the falling edge of GTIOCB pin input, when GTIOCA input is 1.

DSELCm bit (ELC_GPTm Event Source Counter Count Down Enable) (m = A to H)

This bit enables or disables GTCNT counter count down at the ELC_GPTm event input.

23.2.10 General PWM Timer Input Capture Source Select Register A(GTICASR)

Address(es): GPT32m.GTICASR 4007 8024h + 0100h × m (m = 0 to 9)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ASCBFAH	ASCBFAL	ASCBRAH	ASCBRAL	ASCAFBH	ASCAFBL	ASCARBH	ASCARBL	ASGTRGDF	ASGTRGDR	ASGTRGCF	ASGTRGCR	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	Bit name	Description	R/W
b0	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
b1	ASGTRGAF	GTETRGA Pin Falling Input Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the falling edge of GTETRGA input 1: GTCCRA input capture enabled at the falling edge of GTETRGA input.	R/W
b2	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
b3	ASGTRGBF	GTETRGB Pin Falling Input Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the falling edge of GTETRGB input 1: GTCCRA input capture enabled at the falling edge of GTETRGB input.	R/W
b4	ASGTRGCR	GTETRGC Pin Rising Input Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled on the rising edge of GTETRGC input 1: GTCCRA input capture enabled on the rising edge of GTETRGC input.	R/W
b5	ASGTRGCF	GTETRGC Pin Falling Input Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the falling edge of GTETRGC input 1: GTCCRA input capture enabled at the falling edge of GTETRGC input.	R/W
b6	ASGTRGDR	GTETRGD Pin Rising Input Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled on the rising edge of GTETRGD input 1: GTCCRA input capture enabled on the rising edge of GTETRGD input.	R/W
b7	ASGTRGDF	GTETRGD Pin Falling Input Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the falling edge of GTETRGD input 1: GTCCRA input capture enabled at the falling edge of GTETRGD input.	R/W
b8	ASCARBL	GTIOCA Pin Rising Input during GTIOCB Value Low Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled on the rising edge of GTIOCA input when GTIOCB input is 0 1: GTCCRA input capture enabled on the rising edge of GTIOCA input when GTIOCB input is 0.	R/W
b9	ASCARBH	GTIOCA Pin Rising Input during GTIOCB Value High Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled on the rising edge of GTIOCA input when GTIOCB input is 1 1: GTCCRA input capture enabled on the rising edge of GTIOCA input when GTIOCB input is 1.	R/W
b10	ASCAFBL	GTIOCA Pin Falling Input during GTIOCB Value Low Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the falling edge of GTIOCA input when GTIOCB input is 0 1: GTCCRA input capture enabled at the falling edge of GTIOCA input when GTIOCB input is 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b11	ASCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the falling edge of GTIOCA input when GTIOCB input is 1 1: GTCCRA input capture enabled at the falling edge of GTIOCA input when GTIOCB input is 1.	R/W
b12	ASCBRAL	GTIOCB Pin Rising Input during GTIOCA Value Low Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled on the rising edge of GTIOCB input when GTIOCA input is 0 1: GTCCRA input capture enabled on the rising edge of GTIOCB input when GTIOCA input is 0.	R/W
b13	ASCBRAH	GTIOCB Pin Rising Input during GTIOCA Value High Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled on the rising edge of GTIOCB input when GTIOCA input is 1 1: GTCCRA input capture enabled on the rising edge of GTIOCB input when GTIOCA input is 1.	R/W
b14	ASCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the falling edge of GTIOCB input when GTIOCA input is 0 1: GTCCRA input capture enabled at the falling edge of GTIOCB input when GTIOCA input is 0.	R/W
b15	ASCBFAH	GTIOCB Pin Falling Input during GTIOCA Value High Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the falling edge of GTIOCB input when GTIOCA input is 1 1: GTCCRA input capture enabled at the falling edge of GTIOCB input when GTIOCA input is 1.	R/W
b16	ASELCA	ELC_GPTA Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTA event input 1: GTCCRA input capture enabled at the ELC_GPTA event input.	R/W
b17	ASELCB	ELC_GPTB Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTB event input 1: GTCCRA input capture enabled at the ELC_GPTB event input.	R/W
b18	ASELCC	ELC_GPTC Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTC event input 1: GTCCRA input capture enabled at the ELC_GPTC event input.	R/W
b19	ASELCD	ELC_GPTD Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTD event input 1: GTCCRA input capture enabled at the ELC_GPTD event input.	R/W
b20	ASELCE	ELC_GPTE Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTE event input 1: GTCCRA input capture enabled at the ELC_GPTE event input.	R/W
b21	ASELCF	ELC_GPTF Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTF event input 1: GTCCRA input capture enabled at the ELC_GPTF event input.	R/W
b22	ASELCG	ELC_GPTG Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTG event input 1: GTCCRA input capture enabled at the ELC_GPTG event input.	R/W
b23	ASELCH	ELC_GPTH Event Source GTCCRA Input Capture Enable	0: GTCCRA input capture disabled at the ELC_GPTH event input 1: GTCCRA input capture enabled at the ELC_GPTH event input.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

GTICASR sets the source of input capture for GTCCRA.

ASGTRGAR bit (GTETRGA Pin Rising Input Source GTCCRA Input Capture Enable)

This bit enables or disables input capture for GTCCRA on the rising edge of GTETRGA pin input.

ASGTRGAF bit (GTETRGA Pin Falling Input Source GTCCRA Input Capture Enable)

This bit enables or disables input capture for GTCCRA at the falling edge of GTETRGA pin input.

ASGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRA Input Capture Enable)

This bit enables or disables input capture for GTCCRA on the rising edge of GTETRGB pin input.

ASGTRGBF bit (GTETRGB Pin Falling Input Source GTCCRA Input Capture Enable)

This bit enables or disables input capture for GTCCRA at the falling edge of GTETRGB pin input.

ASGTRGCR bit (GTETRGC Pin Rising Input Source GTCCRA Input Capture Enable)

This bit enables or disables input capture for GTCCRA on the rising edge of GTETRGC pin input.

ASGTRGCF bit (GTETRGC Pin Falling Input Source GTCCRA Input Capture Enable)

This bit enables or disables input capture for GTCCRA at the falling edge of GTETRGC pin input.

ASGTRGDR bit (GTETRGD Pin Rising Input Source GTCCRA Input Capture Enable)

This bit enables or disables input capture for GTCCRA on the rising edge of GTETRGD pin input.

ASGTRGDF bit (GTETRGD Pin Falling Input Source GTCCRA Input Capture Enable)

This bit enables or disables input capture for GTCCRA at the falling edge of GTETRGD pin input.

ASCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source GTCCRA Input Capture Enable)

This bit enables or disables input capture for GTCCRA on the rising edge of GTIOCA pin input, when GTIOCB input is 0.

ASCARBH bit (GTIOCA Pin Rising Input during GTIOCB Value High Source GTCCRA Input Capture Enable)

This bit enables or disables input capture for GTCCRA on the rising edge of GTIOCA pin input, when GTIOCB input is 1.

ASCAFBL bit (GTIOCA Pin Falling Input during GTIOCB Value Low Source GTCCRA Input Capture Enable)

This bit enables or disables input capture for GTCCRA at the falling edge of GTIOCA pin input, when GTIOCB input is 0.

ASCAFBH bit (GTIOCA Pin Falling Input during GTIOCB Value High Source GTCCRA Input Capture Enable)

This bit enables or disables input capture for GTCCRA at the falling edge of GTIOCA pin input, when GTIOCB input is 1.

ASCBRAL bit (GTIOCB Pin Rising Input during GTIOCA Value Low Source GTCCRA Input Capture Enable)

This bit enables or disables input capture for GTCCRA on the rising edge of GTIOCB pin input, when GTIOCA input is 0.

ASCBRAH bit (GTIOCB Pin Rising Input during GTIOCA Value High Source GTCCRA Input Capture Enable)

This bit enables or disables input capture for GTCCRA on the rising edge of GTIOCB pin input, when GTIOCA input is 1.

ASCBFAL bit (GTIOCB Pin Falling Input during GTIOCA Value Low Source GTCCRA Input Capture Enable)

This bit enables or disables input capture for GTCCRA at the falling edge of GTIOCB pin input, when GTIOCA input is 0.

ASCBFAH bit (GTIOCB Pin Falling Input during GTIOCA Value High Source GTCCRA Input Capture Enable)

This bit enables or disables input capture for GTCCRA at the falling edge of GTIOCB pin input, when GTIOCA input is 1.

ASELCm bit (ELC_GPTm Event Source Counter GTCCRA Input Capture Enable) (m = A to H)

This bit enables or disables input capture for GTCCRA at the ELC_GPTm event input.

23.2.11 General PWM Timer Input Capture Source Select Register B(GTICBSR)

Address(es): GPT32m.GTICBSR 4007 8028h + 0100h × m (m = 0 to 9)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSCBFAH	BSCBFAL	BSCBRAH	BSCBRAL	BSCAFBH	BSCAFBL	BSCARBH	BSCARBL	BSGTRGDF	BSGTRGDR	BSGTRGCF	BSGTRGCR	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	Bit name	Description	R/W
b0	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
b1	BSGTRGAF	GTETRGA Pin Falling Input Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the falling edge of GTETRGA input 1: GTCCRB input capture enabled at the falling edge of GTETRGA input.	R/W
b2	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
b3	BSGTRGBF	GTETRGB Pin Falling Input Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the falling edge of GTETRGB input 1: GTCCRB input capture enabled at the falling edge of GTETRGB input.	R/W
b4	BSGTRGCR	GTETRGC Pin Rising Input Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled on the rising edge of GTETRGC input 1: GTCCRB input capture enabled on the rising edge of GTETRGC input.	R/W
b5	BSGTRGCF	GTETRGC Pin Falling Input Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the falling edge of GTETRGC input 1: GTCCRB input capture enabled at the falling edge of GTETRGC input.	R/W
b6	BSGTRGDR	GTETRGD Pin Rising Input Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled on the rising edge of GTETRGD input 1: GTCCRB input capture enabled on the rising edge of GTETRGD input.	R/W
b7	BSGTRGDF	GTETRGD Pin Falling Input Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the falling edge of GTETRGD input 1: GTCCRB input capture enabled at the falling edge of GTETRGD input.	R/W
b8	BSCARBL	GTIOCA Pin Rising Input during GTIOCB Value Low Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled on the rising edge of GTIOCA input when GTIOCB input is 0 1: GTCCRB input capture enabled on the rising edge of GTIOCA input when GTIOCB input is 0.	R/W
b9	BSCARBH	GTIOCA Pin Rising Input during GTIOCB Value High Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled on the rising edge of GTIOCA input when GTIOCB input is 1 1: GTCCRB input capture enabled on the rising edge of GTIOCA input when GTIOCB input is 1.	R/W
b10	BSCAFBL	GTIOCA Pin Falling Input during GTIOCB Value Low Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the falling edge of GTIOCA input when GTIOCB input is 0 1: GTCCRB input capture enabled at the falling edge of GTIOCA input when GTIOCB input is 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b11	BSCAFBH	GTIOCA Pin Falling Input during GTIOCB Value High Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the falling edge of GTIOCA input when GTIOCB input is 1 1: GTCCRB input capture enabled at the falling edge of GTIOCA input when GTIOCB input is 1.	R/W
b12	BSCBRAL	GTIOCB Pin Rising Input during GTIOCA Value Low Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled on the rising edge of GTIOCB input when GTIOCA input is 0 1: GTCCRB input capture enabled on the rising edge of GTIOCB input when GTIOCA input is 0.	R/W
b13	BSCBRAH	GTIOCB Pin Rising Input during GTIOCA Value High Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled on the rising edge of GTIOCB input when GTIOCA input is 1 1: GTCCRB input capture enabled on the rising edge of GTIOCB input when GTIOCA input is 1.	R/W
b14	BSCBFAL	GTIOCB Pin Falling Input during GTIOCA Value Low Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the falling edge of GTIOCB input when GTIOCA input is 0 1: GTCCRB input capture enabled at the falling edge of GTIOCB input when GTIOCA input is 0.	R/W
b15	BSCBFAH	GTIOCB Pin Falling Input during GTIOCA Value High Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the falling edge of GTIOCB input when GTIOCA input is 1 1: GTCCRB input capture enabled at the falling edge of GTIOCB input when GTIOCA input is 1.	R/W
b16	BSELCA	ELC_GPTA Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTA event input 1: GTCCRB input capture enabled at the ELC_GPTA event input.	R/W
b17	BSELCB	ELC_GPTB Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTB event input 1: GTCCRB input capture enabled at the ELC_GPTB event input.	R/W
b18	BSELCC	ELC_GPTC Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTC event input 1: GTCCRB input capture enabled at the ELC_GPTC event input.	R/W
b19	BSELCD	ELC_GPTD Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTD event input 1: GTCCRB input capture enabled at the ELC_GPTD event input.	R/W
b20	BSELCE	ELC_GPTE Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTE event input 1: GTCCRB input capture enabled at the ELC_GPTE event input.	R/W
b21	BSELCF	ELC_GPTF Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTF event input 1: GTCCRB input capture enabled at the ELC_GPTF event input.	R/W
b22	BSELCG	ELC_GPTG Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTG event input 1: GTCCRB input capture enabled at the ELC_GPTG event input.	R/W
b23	BSELCH	ELC_GPTH Event Source GTCCRB Input Capture Enable	0: GTCCRB input capture disabled at the ELC_GPTH event input 1: GTCCRB input capture enabled at the ELC_GPTH event input.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

GTICBSR sets the source of input capture for GTCCRB.

BSGTRGAR bit (GTETRGA Pin Rising Input Source GTCCRB Input Capture Enable)

This bit enables or disables input capture for GTCCRB on the rising edge of GTETRGA pin input.

BSGTRGAF bit (GTETRGA Pin Falling Input Source GTCCRB Input Capture Enable)

This bit enables or disables input capture for GTCCRB at the falling edge of GTETRGA pin input.

BSGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRB Input Capture Enable)

This bit enables or disables input capture for GTCCRB on the rising edge of GTETRGB pin input.

BSGTRGBF bit (GTETRGB Pin Falling Input Source GTCCRB Input Capture Enable)

This bit enables or disables input capture for GTCCRB at the falling edge of GTETRGB pin input.

BSGTRGCR bit (GTETRGC Pin Rising Input Source GTCCRB Input Capture Enable)

This bit enables or disables input capture for GTCCRB on the rising edge of GTETRGC pin input.

BSGTRGCF bit (GTETRGC Pin Falling Input Source GTCCRB Input Capture Enable)

This bit enables or disables input capture for GTCCRB at the falling edge of GTETRGC pin input.

BSGTRGDR bit (GTETRGD Pin Rising Input Source GTCCRB Input Capture Enable)

This bit enables or disables input capture for GTCCRB on the rising edge of GTETRGD pin input.

BSGTRGDF bit (GTETRGD Pin Falling Input Source GTCCRB Input Capture Enable)

This bit enables or disables input capture for GTCCRB at the falling edge of GTETRGD pin input.

BSCARBL bit (GTIOCA Pin Rising Input during GTIOCB Value Low Source GTCCRB Input Capture Enable)

This bit enables or disables input capture for GTCCRB on the rising edge of GTIOCA pin input, when GTIOCB input is 0.

BSCARBH bit (GTIOCA Pin Rising Input during GTIOCB Value High Source GTCCRB Input Capture Enable)

This bit enables or disables input capture for GTCCRB on the rising edge of GTIOCA pin input, when GTIOCB input is 1.

BSCAFBL bit (GTIOCA Pin Falling Input during GTIOCB Value Low Source GTCCRB Input Capture Enable)

This bit enables or disables input capture for GTCCRB at the falling edge of GTIOCA pin input, when GTIOCB input is 0.

BSCAFBH bit (GTIOCA Pin Falling Input during GTIOCB Value High Source GTCCRB Input Capture Enable)

This bit enables or disables input capture for GTCCRB at the falling edge of GTIOCA pin input, when GTIOCB input is 1.

BSCBRAL bit (GTIOCB Pin Rising Input during GTIOCA Value Low Source GTCCRB Input Capture Enable)

This bit enables or disables input capture for GTCCRB on the rising edge of GTIOCB pin input, when GTIOCA input is 0.

BSCBRAH bit (GTIOCB Pin Rising Input during GTIOCA Value High Source GTCCRB Input Capture Enable)

This bit enables or disables input capture for GTCCRB on the rising edge of GTIOCB pin input, when GTIOCA input is 1.

BSCBFAL bit (GTIOCB Pin Falling Input during GTIOCA Value Low Source GTCCRB Input Capture Enable)

This bit enables or disables input capture for GTCCRB at the falling edge of GTIOCB pin input, when GTIOCA input is 0.

BSCBFAH bit (GTIOCB Pin Falling Input during GTIOCA Value High Source GTCCRB Input Capture Enable)

This bit enables or disables input capture for GTCCRB at the falling edge of GTIOCB pin input, when GTIOCA input is 1.

BSELCm bit (ELC_GPTm Event Source Counter GTCCRB Input Capture Enable) (m = A to H)

This bit enables or disables input capture for GTCCRB at the ELC_GPTm event input.

23.2.12 General PWM Timer Control Register (GTCR)

Address(es): GPT32m.GTCR 4007 802Ch + 0100h × m (m = 0 to 9)

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

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

Bit	Symbol	Bit name	Description	R/W
b0	CST	Count Start	0: Count operation is stopped 1: Count operation is performed.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	MD[2:0]	Mode Select	b18 b16 0 0 0: Saw-wave PWM mode (single buffer or double buffer possible) 0 0 1: Saw-wave one-shot pulse mode (fixed buffer operation) 0 1 0: Setting prohibited 0 1 1: Setting prohibited 1 0 0: Triangle-wave PWM mode 1 (32-bit transfer at trough) (single buffer or double buffer possible) 1 0 1: Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) (single buffer or double buffer possible) 1 1 0: Triangle-wave PWM mode 3 (64-bit transfer at trough) (fixed buffer operation) 1 1 1: Setting prohibited.	R/W
b23 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26 to b24	TPCS[2:0]	Timer Prescaler Select	b26 b24 0 0 0: PCLKD/1 0 0 1: PCLKD/4 0 1 0: PCLKD/16 0 1 1: PCLKD/64 1 0 0: PCLKD/256 1 0 1: PCLKD/1024.	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

GTCR controls GTCNT.

CST bit (Count Start)

This bit controls the GTCNT counter start and stop.

[Setting conditions]

- GTSTR value where the channel number associated with the bit number is set to 1 with the GTSSR.CSTRT bit being 1.
- The ELC event input or the GTIOCA/GTIOCB/GTETRGn port input that are enabled by GTSSR for the starting counter source, occurs.
- 1 is written by software directly.

[Clearing conditions]

- GTSTP value where the channel number associated with the bit number is set to 1 with the GTSSR.CSTOP bit being 1.
- The ELC event input or the GTIOCA/GTIOCB/GTETRGn port input that are enabled by GTSSR for the stopping counter source, occurs.
- 0 is written by software directly.

MD[2:0] bits (Mode Select)

These bits select the GPT operating mode. The MD[2:0] bits must be set while the GTCNT operation is stopped.

TPCS[2:0] bits (Timer Prescaler Select)

These bits select the clock for GTCNT. A clock prescaler can be selected independently for each channel. TPCS[2:0] bits must be set while GTCNT operation is stopped.

23.2.13 General PWM Timer Count Direction and Duty Setting Register (GTUDDTYC)

Address(es): GPT32m.GTUDDTYC 4007 8030h + 0100h × m (m = 0 to 9)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	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

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

Bit	Symbol	Bit name	Description	R/W
b0	UD	Count Direction Setting	0: GTCNT counts down 1: GTCNT counts up.	R/W
b1	UDF	Forcible Count Direction Setting	0: Not forcibly set 1: Forcibly set.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, 16	OADTY[1:0]	GTIOCA Output Duty Setting	b17 b16 0 x: GTIOCA pin duty depends on compare match 1 0: GTIOCA pin duty 0% 1 1: GTIOCA pin duty 100%.	R/W
b18	OADTYF	Forcible GTIOCA Output Duty Setting	0: Not forcibly set 1: Forcibly set.	R/W
b19	OADTYR	GTIOCA Output Value Selecting after Releasing 0%/100% Duty Setting	0: Apply output value set in 0%/100% duty to GTIOA[3:2] function after releasing 0%/100% duty setting. 1: Apply masked compare match output value to GTIOA[3:2] function after releasing 0%/100% duty setting.	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	OBDTY[1:0]	GTIOCB Output Duty Setting	b25 b24 0 x: GTIOCB pin duty is depend on compare match 1 0: GTIOCB pin duty 0% 1 1: GTIOCB pin duty 100%.	R/W
b26	OBDTYF	Forcible GTIOCB Output Duty Setting	0: Not forcibly set 1: Forcibly set.	R/W
b27	OBDTYR	GTIOCB Output Value Selecting after Releasing 0%/100% Duty Setting	0: Apply output value set in 0%/100% duty to GTIOB[3:2] function after releasing 0%/100% duty setting 1: Apply masked compare match output value to GTIOB[3:2] function after releasing 0%/100% duty setting.	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

GTUDDTYC sets the direction in which GTCNT counts (up-counting or down-counting), and sets the duty of GTIOCA/GTIOCB pin output.

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 GTCNT value becomes 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 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 GTCNT value becomes 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 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 changed 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 at that time is reflected in the count direction when counting starts.

UD bit (Count Direction Setting)

This bit sets the count direction (up-counting or down-counting) for GTCNT.

UDF bit (Forcible Count Direction Setting)

This 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, this bit should be returned 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 changes during down-counting, the duty is reflected at an underflow (GTCNT = 0).

When the OADTY/OBDTY value changes to 1 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 changes to 0 with the OADTYF/OBDTYF bit being 1 and while counting is stopped, the output duty is reflected at the 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 changes to 1 with the OADTYF/OBDTYF bit being 0 and while counting stops, the output duty is not reflected at the starting counter operation, However, the output duty is reflected at an underflow. When the OADTY/OBDTY value changes to 0 with the OADTYF/OBDTYF bit being 1 and while counting stops, the output duty is reflected at the starting counter operation.

OmDTY[1:0] bits (GTIOCm Output Duty Setting) (m = A, B)

These bits set the output duty (0%, 100% or compare match control) of the GTIOCm pin.

OmDTYF bit (ForcibleGTIOCm Output Duty Setting) (m = A, B)

This bit forcibly sets the output duty cycle to the OmDTY setting. This bit should be set to 0 during counter operation. When this bit is set to 1 while counting stops, this bit should be returned to 0 until the first period ends after the counter starts.

OmDTYR bit (GTIOCm Output Value Selecting after Releasing 0%/100% Duty Setting) (m = A, B)

These bits select the value that is the object of output retained or toggled at cycle end, when the control changes from 0%/100% duty setting to compare match for GTIOCm pin and GTIOR.GTIOm[3:2] are set to 00b (output retained at cycle end) or GTIOR.GTIOm[3:2] are set to 11b (output toggled at cycle end).

GPT internally continues to perform compare match operation in performing 0%/100% duty operation. When OmDTYR bit is set to 1, the value of compare match at cycle end is applied to GTIOR.GTIOm[3:2].

23.2.14 General PWM Timer I/O Control Register (GTIOR)

Address(es): GPT32m.GTIOR 4007 8034h + 0100h × m (m = 0 to 9)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NFCSB[1:0]		NFBEN	—	—	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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NFCSA[1:0]		NFAEN	—	—	OADF[1:0]		OAE	OAHLD	OADFLT	—	GTIOA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b4 to b0	GTIOA[4:0]	GTIOCA Pin Function Select	See Table 23.5.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	OADFLT	GTIOCA Pin Output Value Setting at the Count Stop	0: The GTIOCA pin outputs low when counting stops 1: The GTIOCA pin outputs high when counting stops.	R/W
b7	OAHLD	GTIOCA Pin Output Setting at the Start/Stop Count	0: The GTIOCA pin output level at the start/stop of counting depends on the register setting 1: The GTIOCA pin output level is retained at the start/stop of counting.	R/W
b8	OAE	GTIOCA Pin Output Enable	0: Output is disabled 1: Output is enabled.	R/W
b10, b9	OADF[1:0]	GTIOCA Pin Disable Value Setting	b10 b9 0 0: Output disable is prohibited 0 1: GTIOCA pin is set to Hi-Z when output disable is performed 1 0: GTIOCA pin is set to 0 when output disable is performed 1 1: GTIOCA pin is set to 1 when output disable is performed.	R/W
b12, b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	NFAEN	Noise Filter A Enable	0: The noise filter for the GTIOCA pin is disabled 1: The noise filter for the GTIOCA pin is enabled.	R/W
b15, b14	NFCSA[1:0]	Noise Filter A Sampling Clock Select	b15 b14 0 0: PCLKD/1 0 1: PCLKD/4 1 0: PCLKD/16 1 1: PCLKD/64.	R/W
b20 to b16	GTIOB[4:0]	GTIOCB Pin Function Select	See Table 23.5.	R/W
b21	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b22	OBDFLT	GTIOCB Pin Output Value Setting at the Count Stop	0: The GTIOCB pin outputs low when counting stops 1: The GTIOCB pin outputs high when counting stops.	R/W
b23	OBHLD	GTIOCB Pin Output Setting at the Start/Stop Count	0: The GTIOCB pin output level at the start/stop of counting depends on the register setting 1: The GTIOCB pin output level is retained at the start/stop of counting.	R/W
b24	OBE	GTIOCB Pin Output Enable	0: Output is disabled 1: Output is enabled.	R/W
b26, b25	OBDF[1:0]	GTIOCB Pin Disable Value Setting	b26 b25 0 0: Output disable is prohibited 0 1: GTIOCB pin is set to Hi-Z when output disable is performed 1 0: GTIOCB pin is set to 0 when output disable is performed 1 1: GTIOCB pin is set to 1 when output disable is performed.	R/W

Bit	Symbol	Bit name	Description	R/W
b28, b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b29	NFBEN	Noise Filter B Enable	0: The noise filter for the GTIOCB pin is disabled 1: The noise filter for the GTIOCB pin is enabled.	R/W
b31, b30	NFCSB[1:0]	Noise Filter B Sampling Clock Select	b31 b30 0 0: PCLKD/1 0 1: PCLKD/4 1 0: PCLKD/16 1 1: PCLKD/64.	R/W

GTIOR sets the functions of the GTIOCA and GTIOCB pins.

GTIOA[4:0] bits (GTIOCA Pin Function Select)

These bits select the GTIOCA pin function. For details, see Table 23.5.

OADFLT bit (GTIOCA Pin Output Value Setting at the Count Stop)

This bit sets whether the GTIOCA pin outputs high or low when counting stops.

OAHLD bit (GTIOCA Pin Output Setting at the Start/Stop Count)

This bit specifies whether the GTIOCA pin output level is retained or the level at the start/stop of counting depends on the register setting.

[When the OAHLD bit is set to 0]

- The value specified by bit [4] of the GTIOA[4:0] bits is output when counting starts
- The value specified by the OADFLT bit is output when counting stops
- If the OADFLT bit is modified while counting stops, it 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 (GTIOCA Pin Output Enable)

This bit disables or enables the GTIOCA 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 GTIOCA pin does not output independently of the OAE bit value.

OADF[1:0] bits (GTIOCA Pin Disable Value Setting)

These bits select the output value of GTIOCA pin when an output disable request occurs.

NFAEN bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the GTIOCA pin. Because changing the value of the bit might lead to 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)

These bits set the sampling interval for the noise filter of the GTIOCA pin. When setting these bits, wait for 2 cycles of the selected sampling interval before setting the input-capture function.

GTIOB[4:0] bits (GTIOCB Pin Function Select)

These bits select the GTIOCB pin function. For details, see Table 23.5.

OBDFLT bit (GTIOCB Pin Output Value Setting at the Count Stop)

This bit sets whether the GTIOCB pin outputs high or low when counting stops.

OBHLD bit (GTIOCB Pin Output Setting at the Start/Stop Count)

This bit specifies whether the GTIOCB pin output level is retained or the level at the start/stop of counting depends on the register setting.

[When the OBHLD bit is set to 0]

- The value specified by bit [4] of the GTIOB[4:0] bits is output when counting starts
- The value specified by the OBDFLT bit is output when counting stops
- If the OBDFLT bit is modified while counting stops, it 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 (GTIOCB Pin Output Enable)

This bit disables or enables the GTIOCB pin output.

When GTCCRB register is used as the input capture register (at least one bit in GTICBSR register is set to 1), the GTIOCB pin does not output independently of the OBE bit value.

OBDF[1:0] bits (GTIOCB Pin Disable Value Setting)

These bits select the output value of GTIOCB pin when an output disable request occurs.

NFBEN bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the GTIOCB 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)

These bits set the sampling interval for the noise filter of the GTIOCB pin. When setting these bits, wait for 2 cycles of the selected sampling interval before setting the input-capture function.

Table 23.5 Settings of GTIOA[4:0] and GTIOB[4:0] bits (1 of 2)

GTIOA/GTIOB[4:0] bits					Function		
b4	b3	b2	b1	b0	b4	b3, b2	b1, b0
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 23.5 Settings of GTIOA[4:0] and GTIOB[4:0] bits (2 of 2)

GTIOA/GTIOB[4:0] bits					Function		
b4	b3	b2	b1	b0	b4	b3, b2	b1, b0
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 is changed from GTPR to 0 in up-counting) or underflow (GTCNT is changed from 0 to GTPR in down-counting). The GTCNT counter is cleared for saw waves and for the trough (GTCNT changes from 0 to 1) for triangle waves.

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, 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 1 bit in GTUPSR or GTDNSR is set to 1, the setting of b3 and b2 is ignored.

23.2.15 General PWM Timer Interrupt Output Setting Register (GTINTAD)

Address(es): GPT32m.GTINTAD 4007 8038h + 0100h × m (m = 0 to 9)

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

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

Bit	Symbol	Bit name	Description	R/W
b23 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	GRP[1:0]	Output Disable Source Select	b25 b24 0 0: Group A output disable request 0 1: Group B output disable request 1 0: Group C output disable request 1 1: Group D output disable request.	R/W
b28 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b29	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
b30	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
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

GTINTAD enables or disables interrupt requests and output-disable requests.

GRP[1:0] bits (Output Disable Source Select)

The GRP[1:0] bits select the GTIOCA or GTIOCB pin output-disable sources.

The output disable request to POEG outputs to the group which is selected in the GRP[1:0] bits when same time output level high or same time output level low occurs based on each output disable request enable bit.

GTST.ODF shows the request of output-disable source group that is selected in the GRP[1:0] bits. The GRP[1:0] bits should be set when both GTIOR.OAE and GTIOR.OBE are 0.

GRPABH bit (Same Time Output Level High Disable Request Enable)

This bit enables or disables output disable request when GTIOCA pin and GTIOCB pin output 1 at the same time.

GRPABL bit (Same Time Output Level Low Disable Request Enable)

This bit enables or disables output disable request when GTIOCA pin and GTIOCB pin output 0 at the same time.

23.2.16 General PWM Timer Status Register (GTST)

Address(es): GPT32m.GTST 4007 803Ch + 0100h × m (m = 0 to 9)

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TUCF	—	—	—	—	—	—	—	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	Bit name	Description	R/W
b0	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
b1	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
b2	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
b3	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
b4	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
b5	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
b6	TCFPO	Overflow Flag	0: No overflow (crest) has occurred 1: An overflow (crest) has occurred.	R/(W)*1
b7	TCFPU	Underflow Flag	0: No underflow (trough) has occurred 1: An underflow (trough) has occurred.	R/(W)*1
b14 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	TUCF	Count Direction Flag	0: The GTCNT counter counts downward 1: The GTCNT counter counts upward.	R
b23 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b24	ODF	Output Disable Flag	0: No output disable request is generated 1: An output disable request is generated.	R
b28 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b29	OABHF	Same Time Output Level High Flag	0: GTIOCA pin and GTIOCB pin do not output 1 at the same time 1: GTIOCA pin and GTIOCB pin output 1 at the same time.	R
b30	OABLF	Same Time Output Level Low Flag	0: GTIOCA pin and GTIOCB pin do not output 0 at the same time 1: GTIOCA pin and GTIOCB pin output 0 at the same time.	R
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit. Do not write 1.

GTST indicates the status of the GPT.

TCFA flag (Input Capture/Compare Match Flag A)

This bit is the status flag 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 bit.

TCFB flag (Input Capture/Compare Match Flag B)

This bit is the status flag 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 bit.

TCFC flag (Input Compare Match Flag C)

This bit is the status flag for the compare match of GTCCRC.

[Setting condition]

- GTCNT = GTCCRC.

[Clearing condition]

- 0 is written to this bit.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (GTCCRC performs buffer operation).

TCFD flag (Input Compare Match Flag D)

This bit is the status flag for the compare match of GTCCRD.

[Setting condition]

- GTCNT = GTCCRD.

[Clearing condition]

- 0 is written to this bit.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (Triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 10b, 11b (GTCCRD performs buffer operation).

TCFE flag (Input Compare Match Flag E)

This bit is the status flag for the compare match of GTCCRE.

[Setting condition]

- GTCNT = GTCCRE.

[Clearing condition]

- 0 is written to this bit.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (Triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (GTCCRE performs buffer operation).

TCFF flag (Input Compare Match Flag F)

This bit is the status flag for the compare match of GTCCRF.

[Setting condition]

- GTCNT = GTCCRF.

[Clearing condition]

- 0 is written to this bit.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (Triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 10b, 11b (GTCCRF performs buffer operation).

TCFPO flag (Overflow Flag)

This bit indicates when an overflow or a crest has occurred.

[Setting conditions]

- In saw-wave mode, an overflow (GTCNT changes from GTPR to 0 in up count) 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 count) has occurred.

[Clearing condition]

- 0 is written to this bit.

TCFPU flag (Underflow Flag)

This bit indicates when an underflow or a 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 crest (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.

TUCF flag (Count Direction Flag)

This flag indicates the count direction of GTCNT. In event count operation, this flag is set to 1 in up-counting and is set to 0 in down-counting.

ODF flag (Output Disable Flag)

This flag shows the request of the output disable source group that is selected by GRP[1:0] bits.

When output is disabled, an output disable control is not released within the same in which an output disable request is negated. It is released in the next cycle.

OABHF flag (Same Time Output Level High Flag)

This flag indicates that GTIOCA pin and GTIOCB pin output 1 at the same time.

When GTIOCA pin or GTIOCB pin output 0, this flag is returned to 0. This flag is read only. Writing 0 to clear the flag is prohibited.

When an interrupt by the OABHF flag is enabled (GTINTAD.GRPABH = 1), the OABHF flag is output to POEG as the output disable request.

[Setting condition]

- GTIOCA pin and GTIOCB pin output 1 at the same time when both OAE bit and OBE bit are set to 1.

[Clearing conditions]

- GTIOCA pin output value is different from GTIOCB pin output value when both OAE bit and OBE bit are set to 1
- GTIOCA pin and GTIOCB pin output 0 at the same time when both OAE bit and OBE bit are set to 1
- At least either OAE bit or OBE bit is set to 0.

OABLF flag (Same Time Output Level Low Flag)

This flag indicates that GTIOCA pin and GTIOCB pin output 0 at the same time.

When GTIOCA pin or GTIOCB pin output 1, this flag is returned to 0. This flag is read only. Writing 0 to clear the flag is not prohibited. When an interrupt by the OABLF flag is enabled (GTINTAD.GRPABL = 1), the OABLF flag is output to POEG as the output disable request.

[Setting condition]

- GTIOCA pin and GTIOCB pin output 0 at the same time when both OAE bit and OBE bit are set to 1.

[Clearing conditions]

- GTIOCA pin output value is different from GTIOCB pin output value when both OAE bit and OBE bit are set to 1
- GTIOCA pin and GTIOCB pin output 1 at the same time when both OAE bit and OBE bit are set to 1
- Either the OAE bit or 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 masked by the output disable function. When the output disable state is performed, a compare match is also performed continuously in the GPT and the OABHF/OABLF flag is updated in association with the result of the compared value.

23.2.17 General PWM Timer Buffer Enable Register (GTBER)

Address(es): GPT32m.GTBER 4007 8040h + 0100h × m (m = 0 to 9)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—		—	—	—		—	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

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

Bit	Symbol	Bit name	Description	R/W
b0	BD[0]	GTCCR Buffer Operation Disable	0: Buffer operation is enabled 1: Buffer operation is disabled.	R/W
b1	BD[1]	GTPR Buffer Operation Disable		R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b17, b16	CCRA[1:0]	GTCCRA Buffer Operation	b17 b16 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTCCRA ↔ GTCCRC) 1 x: Double buffer operation (GTCCRA ↔ GTCCRC ↔ GTCCRD).	R/W
b19, b18	CCRB[1:0]	GTCCRB Buffer Operation	b19 b18 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTCCRB ↔ GTCCRE) 1 x: Double buffer operation (GTCCRB ↔ GTCCRE ↔ GTCCRF).	R/W
b21, b20	PR[1:0]	GTPR Buffer Operation	b21 b20 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTPBR → GTPR) 1 x: Setting Prohibited.	R/W

Bit	Symbol	Bit name	Description	R/W
b22	CCRSWT	GTCCRA and GTCCRB Forcible Buffer Operation	Writing 1 to this bit forcibly performs a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after 1 is written. This bit is read as 0.	R/W
b31 to b23	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

GTBER provides settings for the buffer operation and must be set while the GTCNT operation stops.

BD[0] bit (GTCCR Buffer Operation Disable)

This bit disables the buffer operation using GTCCRA, GTCCRC, and GTCCRD combined and the buffer operation using GTCCRB, GTCCRE, and GTCCRF combined.

When GTDTCR.TDE is 1 and when BD[0] is set to 0, GTCCRB does not perform buffer operation and the GTCCRB register is automatically set to a compare match value for a negative-phase waveform with dead time.

BD[1] bit (GTPR Buffer Operation Disable)

This bit disables buffer operation using GTPR and GTPBR combined.

CCRA[1:0] bits (GTCCRA Buffer Operation)

These bits set buffer operation using GTCCRA, GTCCRC and GTCCRD combined. When buffer operation is restricted by the operating mode set in GTCR, the GTCR setting is given priority.*1

CCRB[1:0] bits (GTCCRB Buffer Operation)

These bits set buffer operation using GTCCRB, GTCCRE, and GTCCRF combined. When buffer operation is restricted by the operating mode set in GTCR, the GTCR setting is given priority.*1

PR[1:0] bits (GTPR Buffer Operation)

These bits set buffer operation using GTPR and GTPBR of the combined GPT.

CCRSWT bit (GTCCRA and GTCCRB Forcible Buffer Operation)

Writing 1 to the CCRSWT bit forcibly performs a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after 1 is written. This bit is read as 0 and is valid only when counting is stopped with a specified compare match operation.

Note 1. The buffer operation mode is fixed in saw-wave one-shot pulse mode, or triangle-wave PWM mode 3 (64-bit transfer at trough).

23.2.18 General PWM Timer Counter (GTCNT)

Address(es): GPT32m.GTCNT 4007 8048h + 0100h × m (m = 0 to 9)

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

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

GTCNT is a 32-bit read/write counter and can only be written to after counting stops. GTCNT should be accessed in 32-bit units. Access in 8-bit/16-bit units is prohibited. GTCNT should be set within the range of 0 ≤ GTCNT ≤ GTPR.

23.2.19 General PWM Timer Compare Capture Register n (GTCCRn) (n = A to F)

Address(es): GPT32m.GTCCRA 4007 804Ch + 0100h × m (m = 0 to 9)
GPT32m.GTCCRB 4007 8050h + 0100h × m (m = 0 to 9)
GPT32m.GTCCRC 4007 8054h + 0100h × m (m = 0 to 9)
GPT32m.GTCCRD 4007 805Ch + 0100h × m (m = 0 to 9)
GPT32m.GTCCRE 4007 8058h + 0100h × m (m = 0 to 9)
GPT32m.GTCCRF 4007 8060h + 0100h × m (m = 0 to 9)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

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

GTCCRn registers are read/write registers.

GTCCRA and GTCCRB are registers used for both output compare and input capture. GTCCRC and GTCCRE are compare match registers that can also function as buffer registers for GTCCRA and GTCCRB.

GTCCRD and GTCCRF are compare match registers that can also function as buffer registers for GTCCRC and GTCCRE (double-buffer registers for GTCCRA and GTCCRB).

23.2.20 General PWM Timer Cycle Setting Register (GTPR)

Address(es): GPT32m.GTPR 4007 8064h + 0100h × m (m = 0 to 9)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

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

GTPR is a read/write register that sets the maximum count value of GTCNT. For saw waves, the value of (GTPR + 1) is the cycle. For triangle waves, the value of (GTPR value × 2) is the cycle.

23.2.21 General PWM Timer Cycle Setting Buffer Register (GTPBR)

Address(es): GPT32m.GTPBR 4007 8068h + 0100h × m (m = 0 to 9)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

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

GTPBR is a readable/writable register that functions as a buffer register for GTPR.

23.2.22 General PWM Timer Dead Time Control Register (GTDTCR)

Address(es): GPT32m.GTDTCR 4007 8088h + 0100h × m (m = 0 to 9)

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

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

Bit	Symbol	Bit name	Description	R/W
b0	TDE	Negative-Phase Waveform Setting	0: GTCCRB is set without using GTDVU 1: GTDVU sets the compare match value for negative-phase waveform with automatic dead time in GTCCRB.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

GTDTCR enables automatic setting of a compare match value for negative-phase waveform with dead time. GPT has a dead time control function and the GTDVU register is used for setting dead time value.

TDE bit (Negative-Phase Waveform Setting)

This bit specifies whether to use GTDVU. When GTDVU is 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), is automatically set in GTCCRB. The TDE bit setting is ignored in saw-wave PWM mode, and automatic setting does not take place.

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.

- 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.

23.2.23 General PWM Timer Dead Time Value Register U (GTDVU)

Address(es): GPT32m.GTDVU 4007 808Ch + 0100h x m (m = 0 to 9)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

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

GTDVU is a read/write register that sets the dead time for generating PWM waveforms.

Setting a dead time value that exceeds the cycle is not allowed. The set value can be confirmed by reading from GTCCRB. When GTDVU is used, writing to GTCCRB is prohibited. When this register is set to 0, waveforms without dead time are output. While the GPT runs, changing the GTDVU values is not allowed. To change GTDVU to a new value, the GPT must be stopped with the CST bit in the GTCR register. GTDVU should be accessed in 32-bit units. Access in 8-bit/16-bit units is not allowed.

23.2.24 Output Phase Switching Control Register (OPSCR)

Address(es): GPT_OPS.OPSCR 4007 8FF0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	NFCS[1:0]		NFEN	—	—	GODF	GRP[1:0]		—	—	ALIGN	—	INV	N	P	FB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	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	Bit name	Description	R/W
b0	UF	Input Phase Soft Setting	These bits set the input phase from the software settings. Setting these bits is valid when the OPSCR.FB bit = 1.	R/W
b1	VF			R/W
b2	WF			R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	U	Input U-Phase Monitor	These bits monitor the state of the input phase. OPSCR.FB = 0: External input monitoring by PCLKD OPSCR.FB = 1: Software settings (UF/VF/WF).	R
b5	V	Input V-Phase Monitor		R
b6	W	Input W-Phase Monitor		R
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	EN	Enable-Phase Output Control	0: Not Output ("Hi-Z" external pin). 1: Output.*1	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	FB	External Feedback Signal Enable	This bit selects the input phase from the software settings and external input. 0: Select the external input 1: Select the soft setting (OPSCR.UF, VF, WF).	R/W
b17	P	Positive-Phase Output (P) Control	0: Level signal output 1: PWM signal output (PWM of GPT320).	R/W
b18	N	Negative-Phase Output (N) Control	0: Level signal output 1: PWM signal output (PWM of GPT320).	R/W

Bit	Symbol	Bit name	Description	R/W
b19	INV	Invert-Phase Output Control	0: Positive logic (active-high) output 1: Negative logic (active-low) output.	R/W
b20	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b21	ALIGN	Input Phase Alignment	0: Input phase is aligned to PCLKD 1: Input phase is aligned PWM.	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25, b24	GRP[1:0]	Output Disabled Source Selection	b25 b24 0 0: Select Group A output disable source 0 1: Select Group B output disable source 1 0: Select Group C output disable source 1 1: Select Group D output disable source.	R/W
b26	GODF	Group Output Disable Function	0: This bit function is ignored 1: Group disable clears the OPSCR.EN bit.*1	R/W
b28, b27	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b29	NFEN	External Input Noise Filter Enable	0: Do not use a noise filter to the external input 1: Use a noise filter to the external input.	R/W
b31, b30	NFCS[1:0]	External Input Noise Filter Clock Selection	Noise filter sampling clock setting of the external input. b31 b30 0 0: PCLKD/1 0 1: PCLKD/4 1 0: PCLKD/16 1 1: PCLKD/64.	R/W

Note 1. When OPSCR.GODF = 1 and the signal value selected by the OPSCR.GRP bit 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)

These bits set the input phase from the software settings. When OPSCR.FB bit = 1, these bits are valid. The set value of the UF/VF/WF take the place of the U/V/W external input.

U, V, W bits (Input Phase Monitor)

When OPSCR.FB bit is 0, external inputs that are synchronized by PCLKD are monitored by these bits. When 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 (Enable-Phase Output Control)

This bit controls the output enable signal output phase (positive phase/reverse phase).

When OPSCR.EN bit = 1, the signal waveform is output.

When OPSCR.EN bit = 0, first set OPSCR.FB, OPSCR.UF/VF/WF (software setting is selected), OPSCR.P/N, OPSCR.INV, OPSCR.RV, OPSCR.ALIGN, OPSCR.NFCS, OPSCR.GRP, OPSCR.GODF, OPSCR.NFEN, OPSCR.NFCS. Then, set this bit to 1. Also when OPSCR.GODF = 1 and the signal value selected by the OPSCR.GRP bit is high, the OPSCR.EN bit is set to 0.

FB bit (External Feedback Signal Enable)

This 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)

This bit selects one of the Level signal output (PWM of GPT320) or PWM signal output for the positive-phase output (GTOUUP pin, GTOVUP pin, GTOWUP pin).

N bit (Negative-Phase Output (N) Control)

This bit selects one of the Level signal output (PWM of GPT320) or PWM signal output for the negative-phase output (GTOULO pin, GTOVLO pin, GTOWLO pin).

INV bit (Invert-Phase Output Control)

This bit selects one of the positive logic (active-high) output or negative logic (active-low) output for the output phase.

ALIGN bit (Input Phase Alignment)

This bit selects the PCLKD or PWM for the sampling of the input phase (input phase is specified in the OPSCR.FB bit).

When OPSCR.ALIGN bit = 0, input phase is aligned to PCLKD.

Note: When PWM output is selected (OPSCR.P/N = 1) and the PCLKD input phase is aligned, the PWM pulse may be short-pulsed.

Note: When OPSCR.ALIGN bit = 1, input phase is aligned with PWM output.

GRP[1:0] bits (Output Disabled Source Selection)

These bits select the output disable source (A to D).

GODF bit (Group Output Disable Function)

When OPSCR.GODF = 1 and signal value selected by the OPSCR.GRP bit is high, the OPSCR.EN bit is set to 0. When OPSCR.GODF bit = 0, this bit is ignored.

NFEN bit (External Input Noise Filter Enable)

This bit selects the noise filter for external input.

When OPSCR.NFEN bit = 0, a noise filter to the external input is not used.

Note: When this bit is switched because of an unintentional internal edge, set OPSCR.EN bit to 0.

NFCS[1:0] bits (External Input Noise Filter Clock Selection)

When OPSCR.NFEN bit = 1, noise filter sampling clock setting of the external input is enabled.

1. Set the NFCS.
2. Wait for 2 cycles.
3. Set the OPSCR.EN bit to 1.

23.3 Operation

23.3.1 Basic Operation

Each channel has a 32-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 controls the count cycle.

When the GTCNT counter value matches the value in GTCCRA or GTCCRB, the output from the corresponding pin GTIOCA or GTIOCB can be changed. 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.

23.3.1.1 Counter Operation

(1) Counter Start/Stop

The counter of each channel starts the count operation by setting GTCR.CST to 1. The GTCR.CST bit value is changed by 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.

(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 00000000h. When the GTCNT value changes from the GTPR value to 0 (overflow), the GTST.TCFPO flag is set to 1. After GTCNT overflows, up-counting resumes from 00000000h.

Figure 23.3 shows an example of a periodic count operation in up-counting.

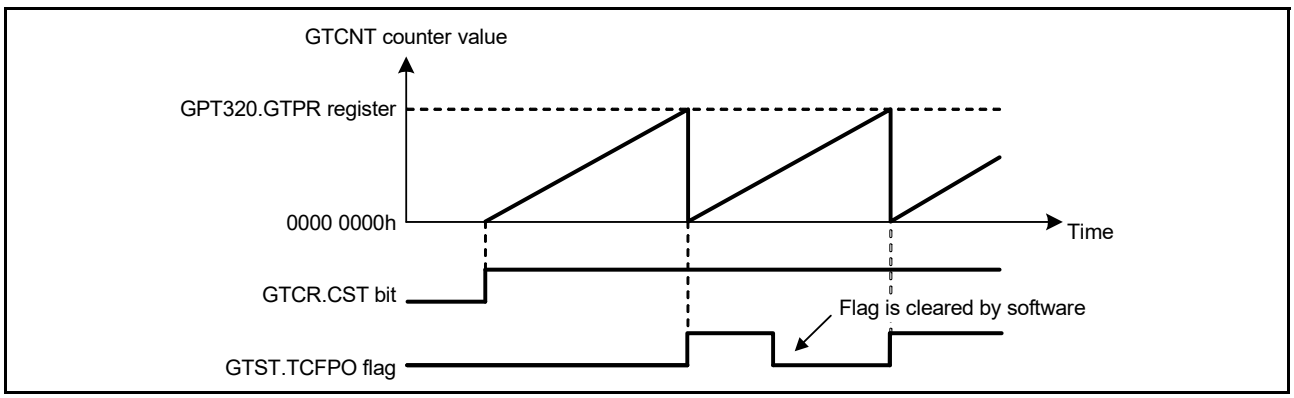


Figure 23.3 Example of periodic count operation in up-counting by the count clock

Figure 23.4 shows an example for setting periodic count operation in up-counting.

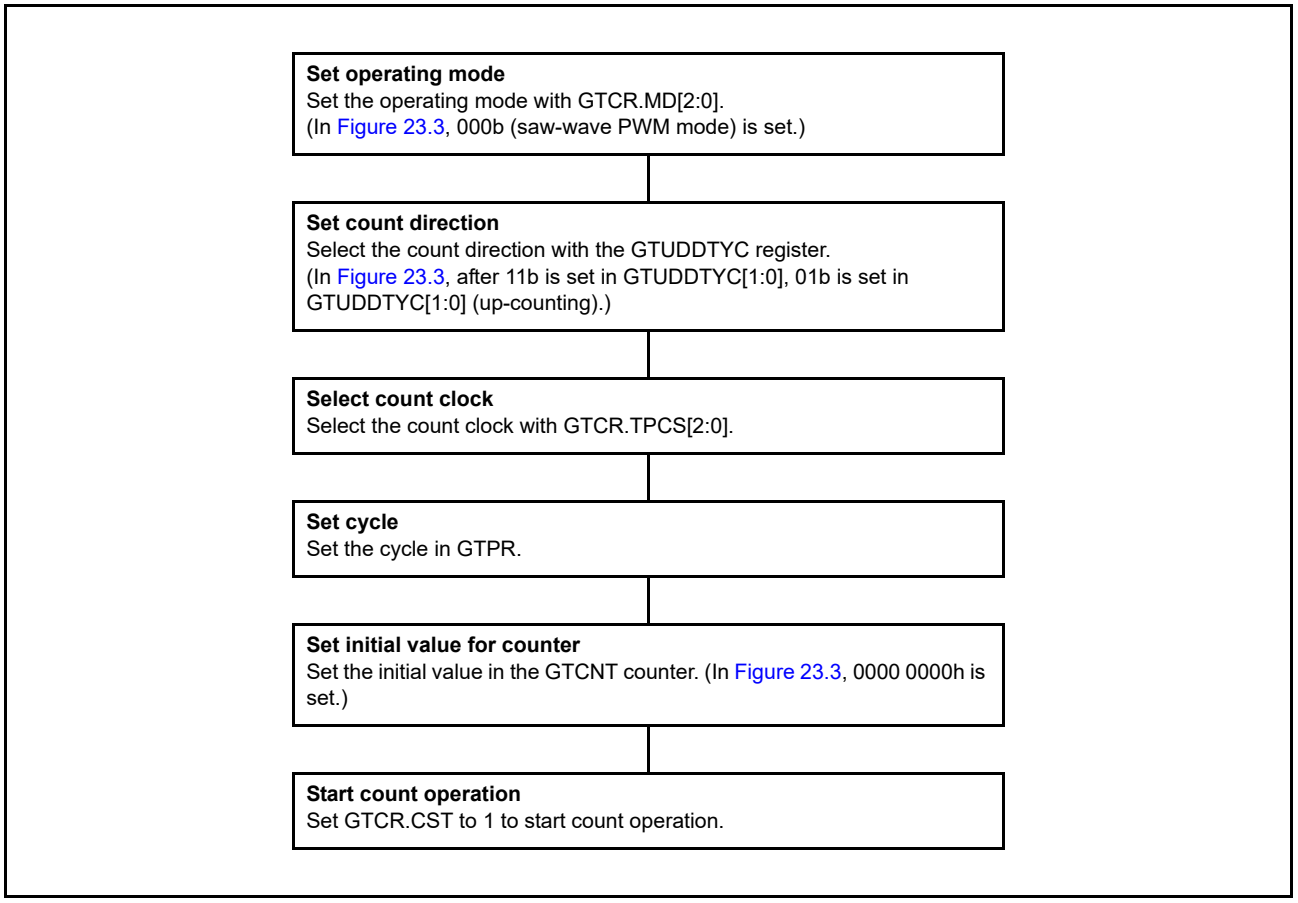


Figure 23.4 Example for setting a periodic count operation in up-counting by the count clock

(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 00000000h. When GTCNT changes from 0 to the GTPR value (underflow), GTST.TCFPU is set to 1. When the GTCNT counter underflows, down-counting resumes from the GTPR value.

Figure 23.5 shows an example of periodic count operation in down-counting by the count clock.

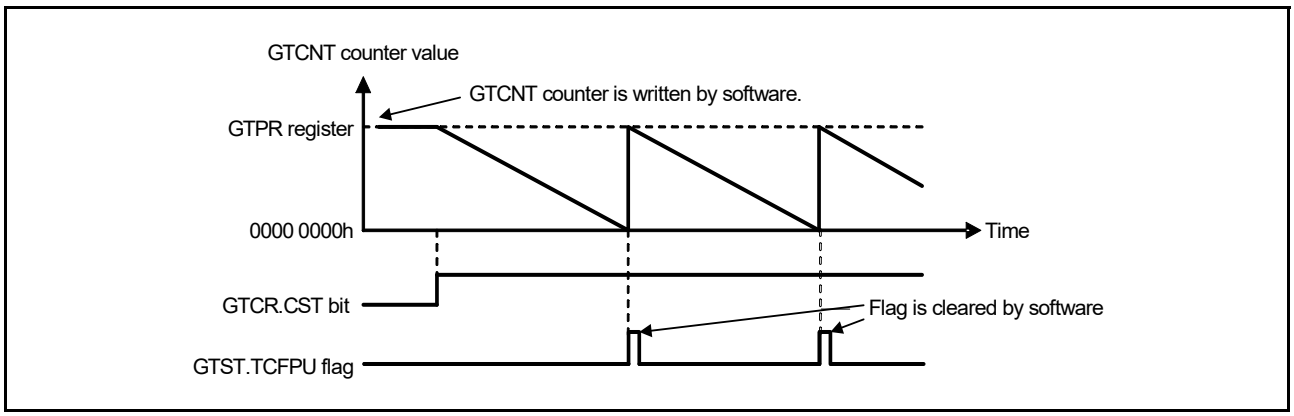


Figure 23.5 Example of periodic count operation in down-counting by the count clock

Figure 23.6 shows an example for setting periodic count operation in down-counting by the count clock.

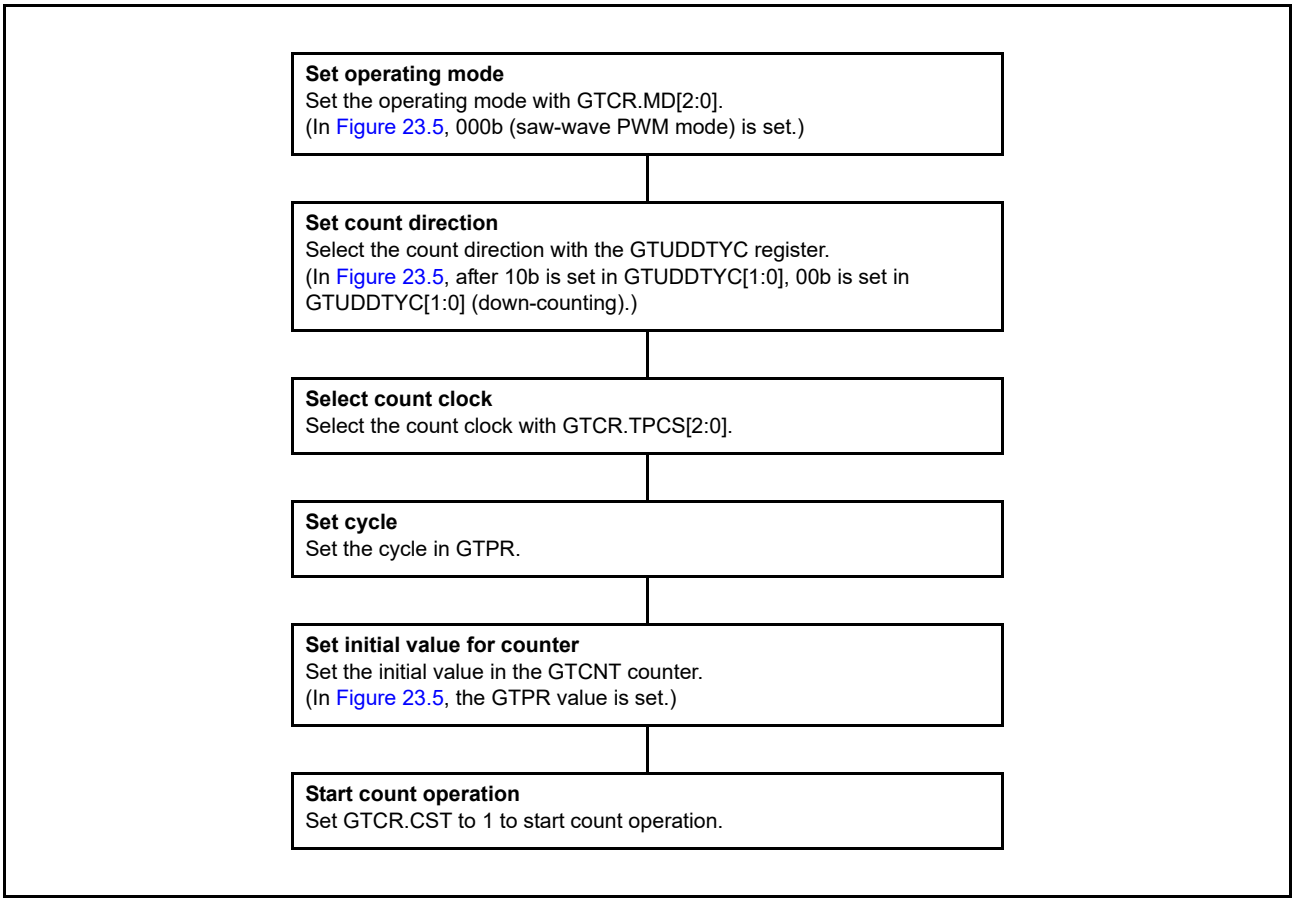


Figure 23.6 Example for setting periodic count operation in down-counting by count clock

(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[2:0] 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 for up-counting using hardware sources is the same as for up-counting by the count clock.

When GTCR.CST bit is set to 1 to count up using hardware sources, the count operation is enabled. When GTCR.CST is set to 1, the counter cannot count up for 1 clock cycle as specified by GTCR.TPCS[2:0] because the count operation is synchronized by the count clock selected by GTCR.TPCS[2:0]. Set GTCR.TPCS[2:0] to 000b to count up with a 1 PCLKD delay after GTCR.CST is set to 1.

Figure 23.7 shows an example of a periodic count operation in up-counting by a hardware resource (rising edge of GTETRGA pin).

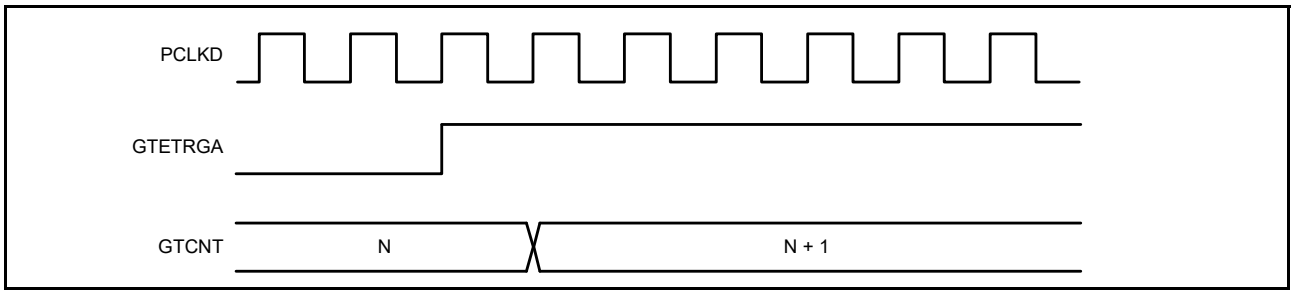


Figure 23.7 Example of periodic count operation in up-counting using hardware sources

Figure 23.8 shows an example for setting periodic count operation in down-counting by the count clock.

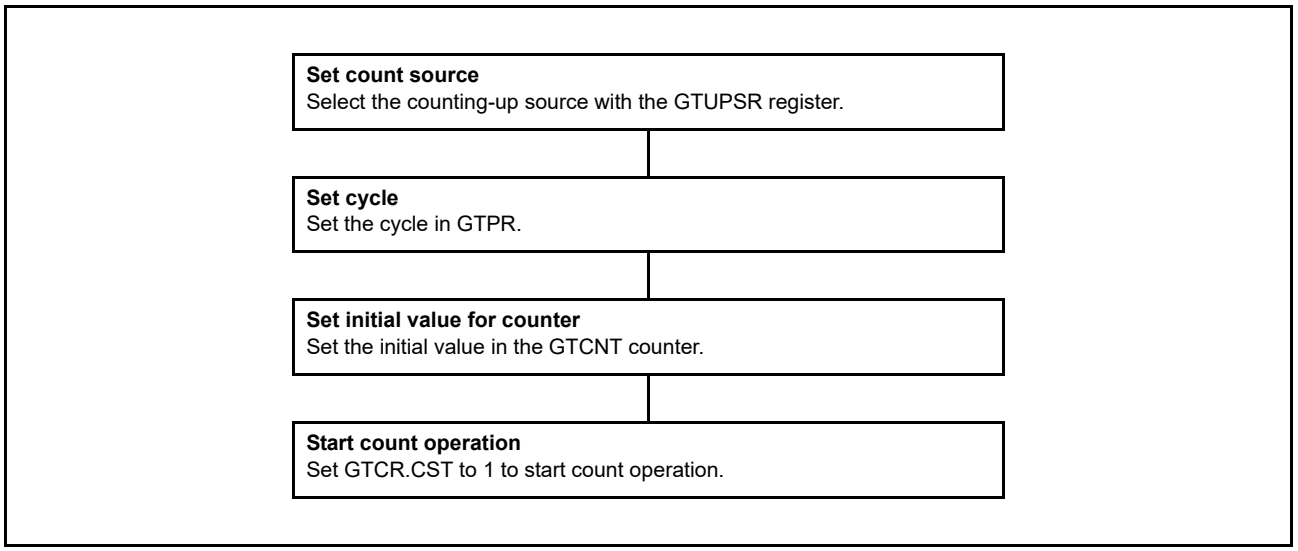


Figure 23.8 Example for setting an event count operation in up-counting using hardware sources

(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[2:0] and the count direction selected in GTUDDTYC.UD are ignored. If up-counting and down-counting using hardware sources occur at the same time, GTCNT counter value does not change. The underflow behavior for down-counting using hardware sources is the same as for 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. When GTCR.CST is set to 1, the counter cannot count down for 1 clock cycle as specified by GTCR.TPCS[2:0] because the count operation is synchronized with the count clock selected by GTCR.TPCS[2:0]. Set GTCR.TPCS[2:0] to 000b to count down with a 1 PCLKD delay after GTCR.CST is set to 1.

Figure 23.9 shows an example of a periodic count operation in down-counting by a hardware resource (rising edge of GTETRGA pin).

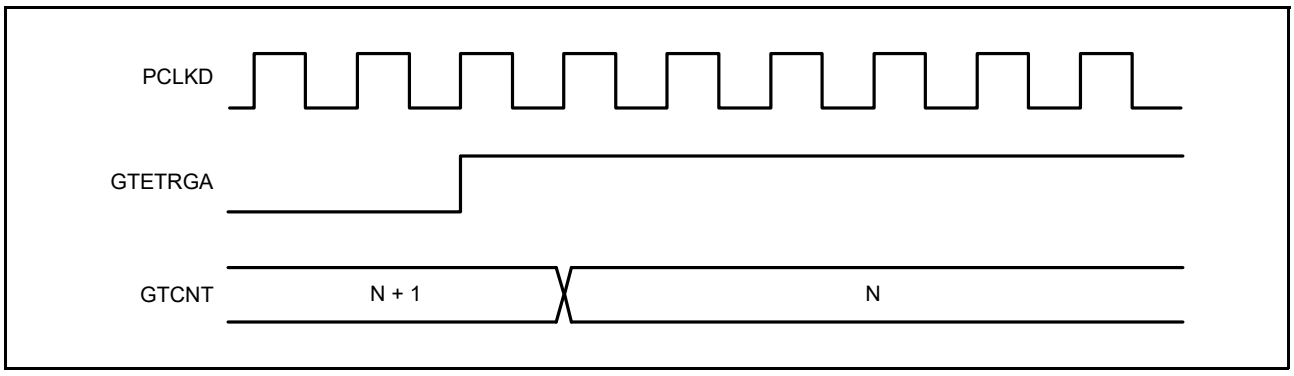


Figure 23.9 Example of event count operation in down-counting using hardware sources

Figure 23.10 shows an example for setting a periodic count operation in down-counting using a hardware resource.

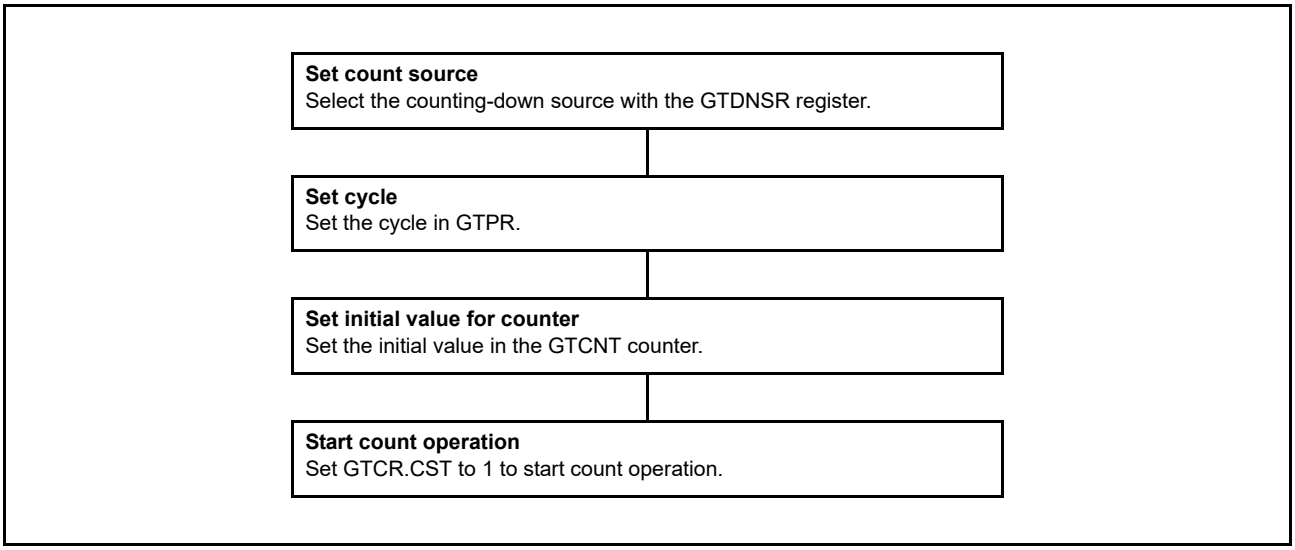


Figure 23.10 Example for setting an event count operation in down-counting using hardware sources

(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. 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 = 1) or not (GTCR.CST = 0).

For saw waves selected by setting GTCR.MD[2:0] and the count direction flag showing down-counting (GTST.TUCF = 0), 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. When not in saw waves 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 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 PCLKD. If other settings are used, clear is synchronized with the counter clock selected by GTCR.TPCS[2:0].

23.3.1.2 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 corresponding GTIOCA or GTIOCB output pin. In addition, the GTIOCA or GTIOCB pin output can be low, high, or toggled at the cycle end which is determined by GTPR.

The cycle end is:

- For saw waves in up-counting – when GTCNT changes from the GTPR value to 0 (overflow)
- For saw waves in down-counting – when GTCNT changes from 0 to GTPR value (underflow)
- For saw waves – when the GTCNT counter is cleared
- For triangle waves – when the GTCNT changes from 0 to 1 (trough).

(1) Low Output and High Output

Figure 23.11 shows an example of low output and high output operation by a compare match of GTCCRA and GTCCRB.

In this example, the GPT320.GTCNT counter performs up-counting, and settings are made so that high is output from the GTIOC0A pin by a GPT320.GTCCRA compare match, and low is output from the GTOC0B pin by a GPT320.GTCCRB compare match. The pin level does not change when the specified level and pin level match.

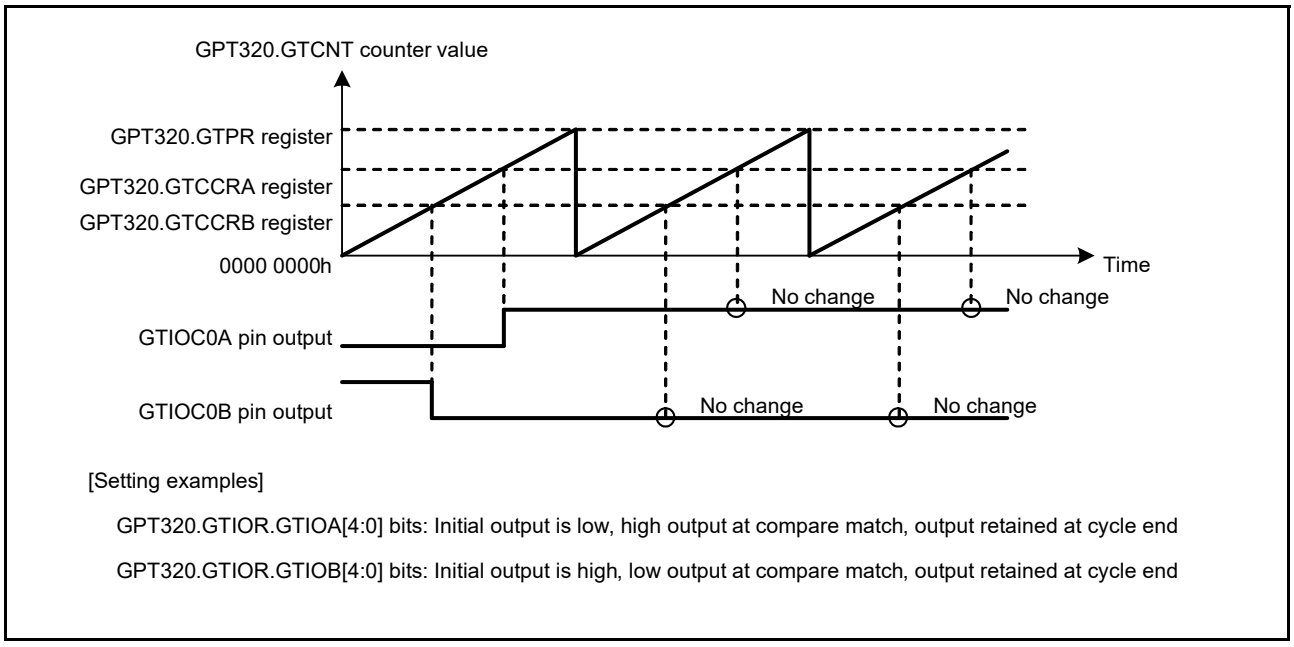


Figure 23.11 Example of low output and high output operation

Figure 23.12 shows an example for setting low output and high output operation.

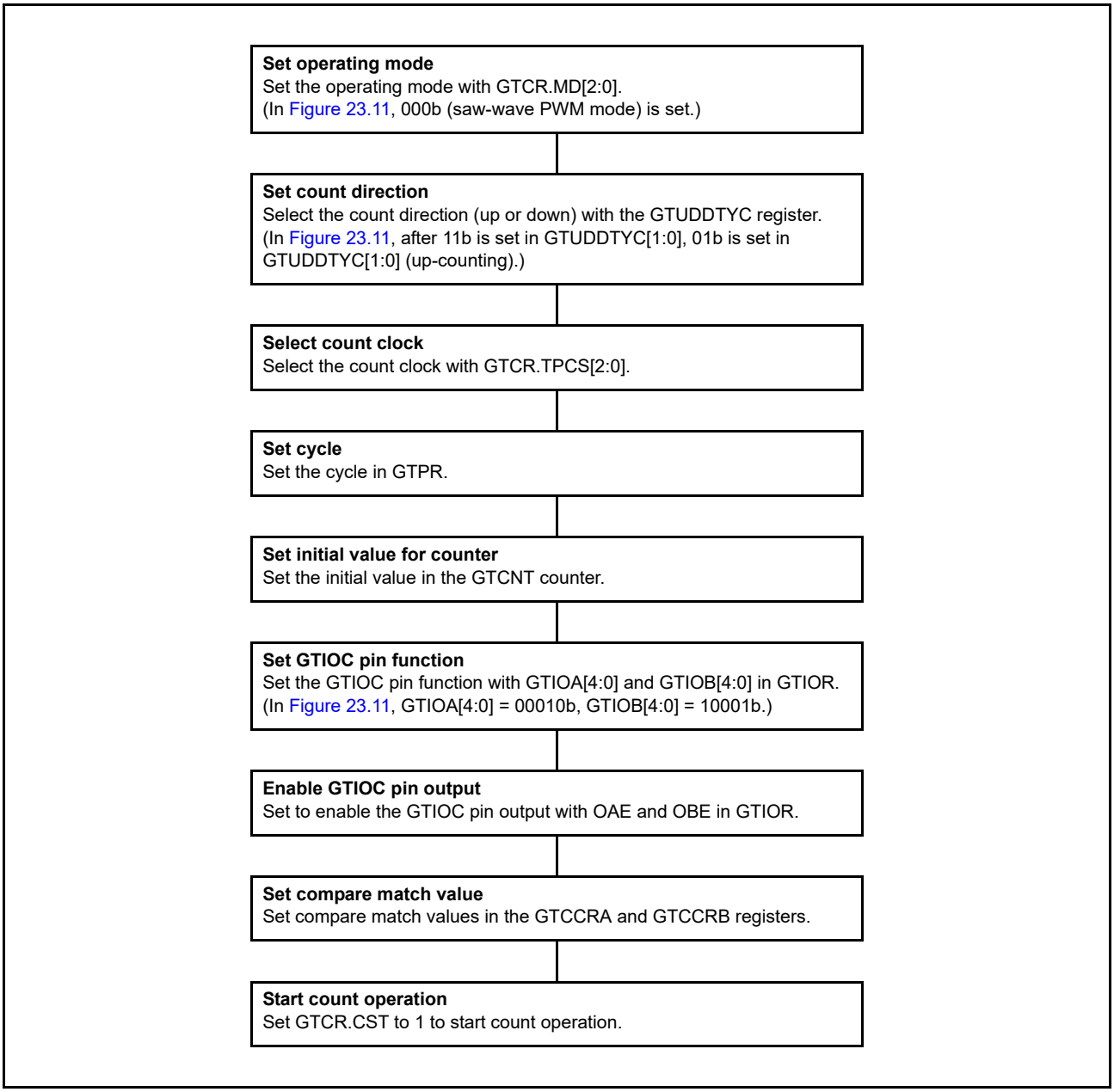
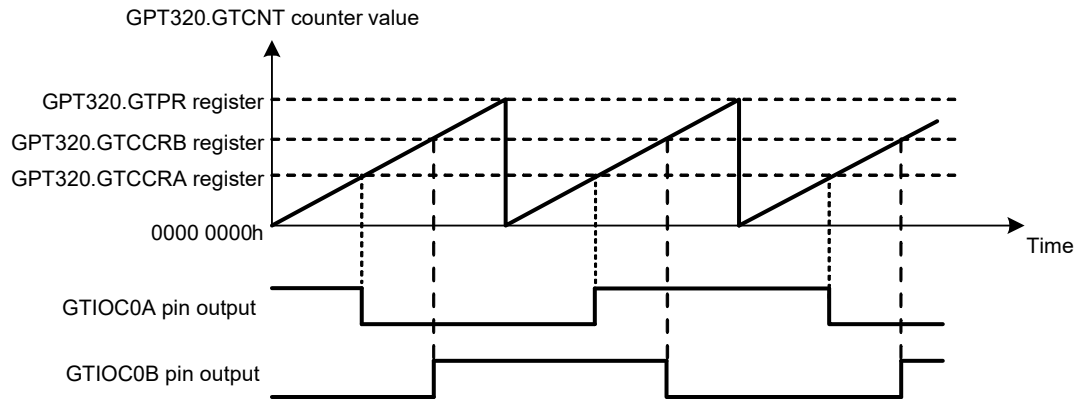


Figure 23.12 Example for setting low output and high output operation

(2) Toggled Output

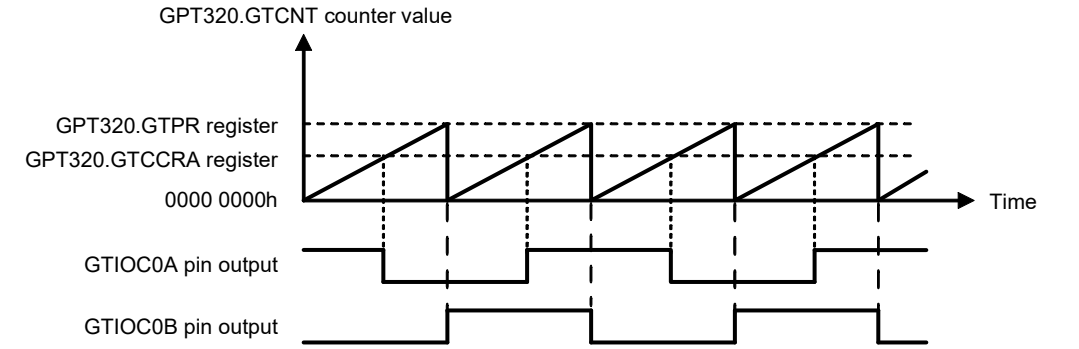
Figure 23.13 and Figure 23.14 show examples of toggled output operation by compare matches of GTCCRA and GTCCRB. In Figure 23.13, the GPT320.GTCNT counter performs up-counting, and settings are made so that the GTIOC0A pin output by a GPT320.GTCCRA compare match and GTIOC0B pin output by a GPT320.GTCCRB compare match are toggled.

In Figure 23.14, the GPT320.GTCNT counter performs up-counting, and settings are made so that the GTIOC0A output is toggled by a compare match of GPT320.GTCCRA and the GTIOC0B output is toggled at the cycle end.



[Setting examples]
GPT320.GTIOR.GTIOA[4:0] bits: Initial output is high, output toggled at compare match, output retained at cycle end
GPT320.GTIOR.GTIOB[4:0] bits: Initial output is low, output toggled at compare match, output retained at cycle end

Figure 23.13 Example of toggled output operation (1)



[Setting examples]
GPT320.GTIOR.GTIOA[4:0] bits: Initial output is high, output toggled at compare match, output retained at cycle end
GPT320.GTIOR.GTIOB[4:0] bits: Initial output is low, output retained at compare match, output toggled at cycle end

Figure 23.14 Example of toggled output operation (2)

Figure 23.15 shows an example for setting toggled output operation.

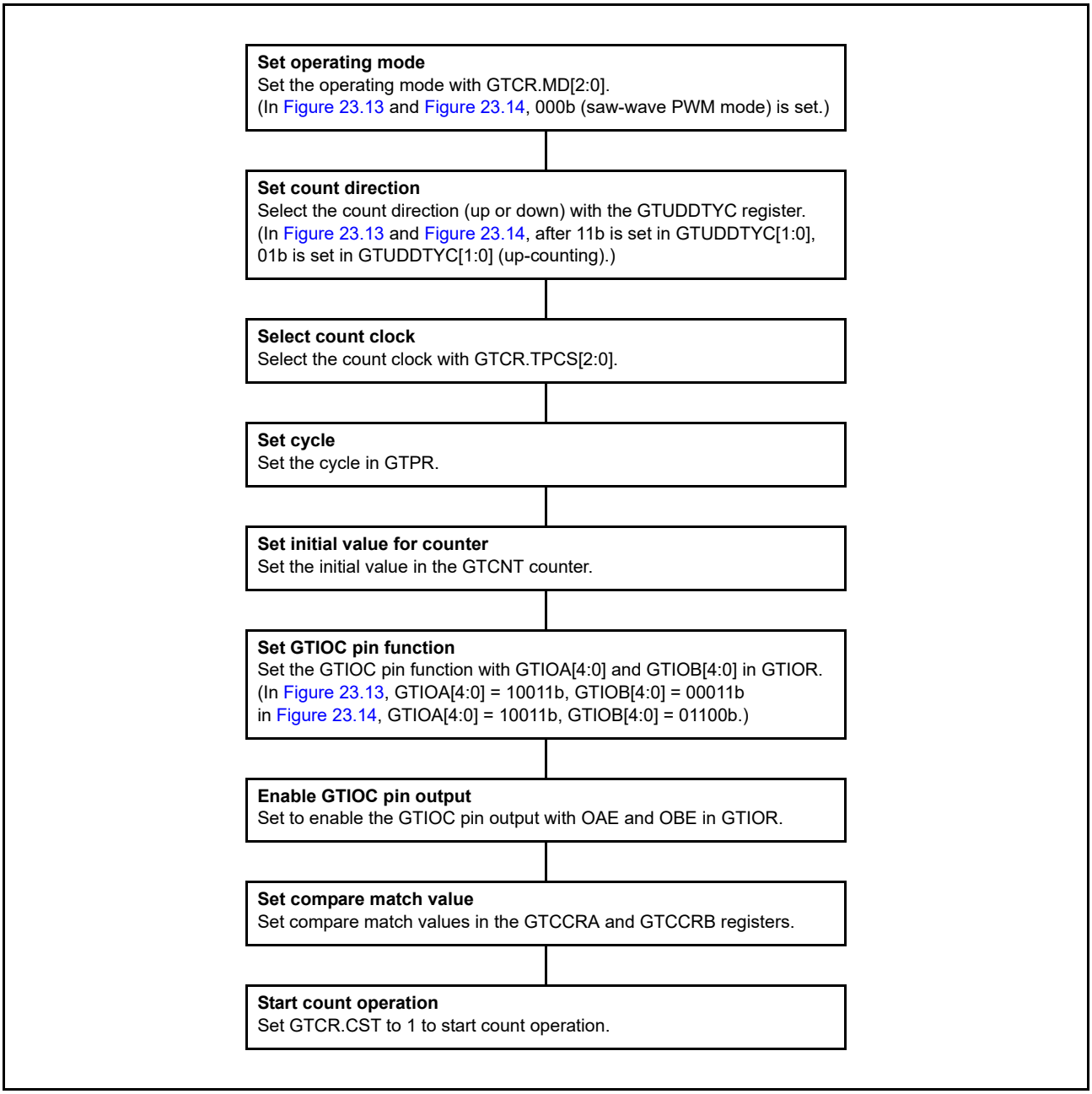


Figure 23.15 Example for setting toggled output operation

23.3.1.3 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.

Figure 23.16 shows an example of the input capture function.

In this example, the GPT320.GTCNT counter performs up-counting by the count clock, and settings are made so that an input capture is performed to GTICCRA at both edges of the GTIOC0A input pin and to GTICCRB on the rising edge of the GTIOC0B input pin.

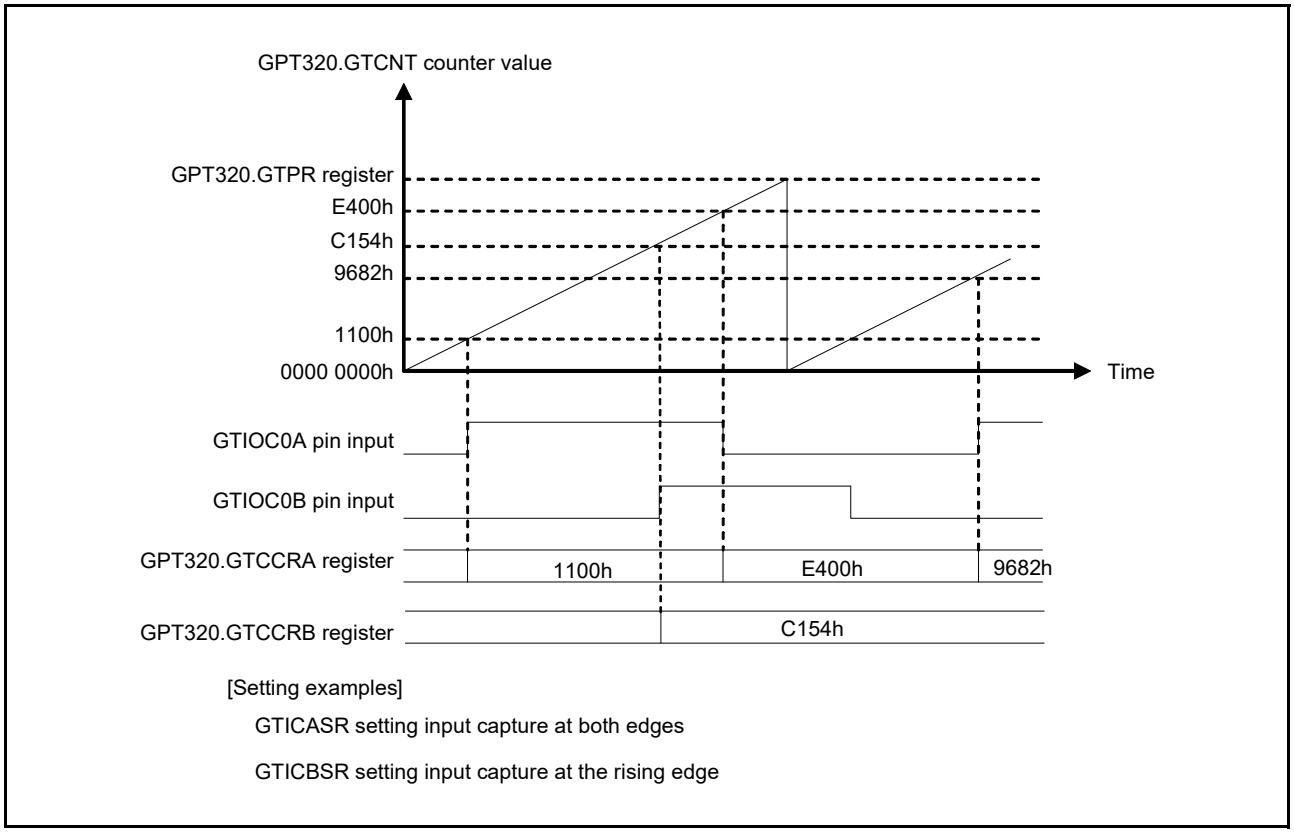


Figure 23.16 Example of input capture operation

Figure 23.17 shows an example for setting an input capture operation with count operation by the count clock.

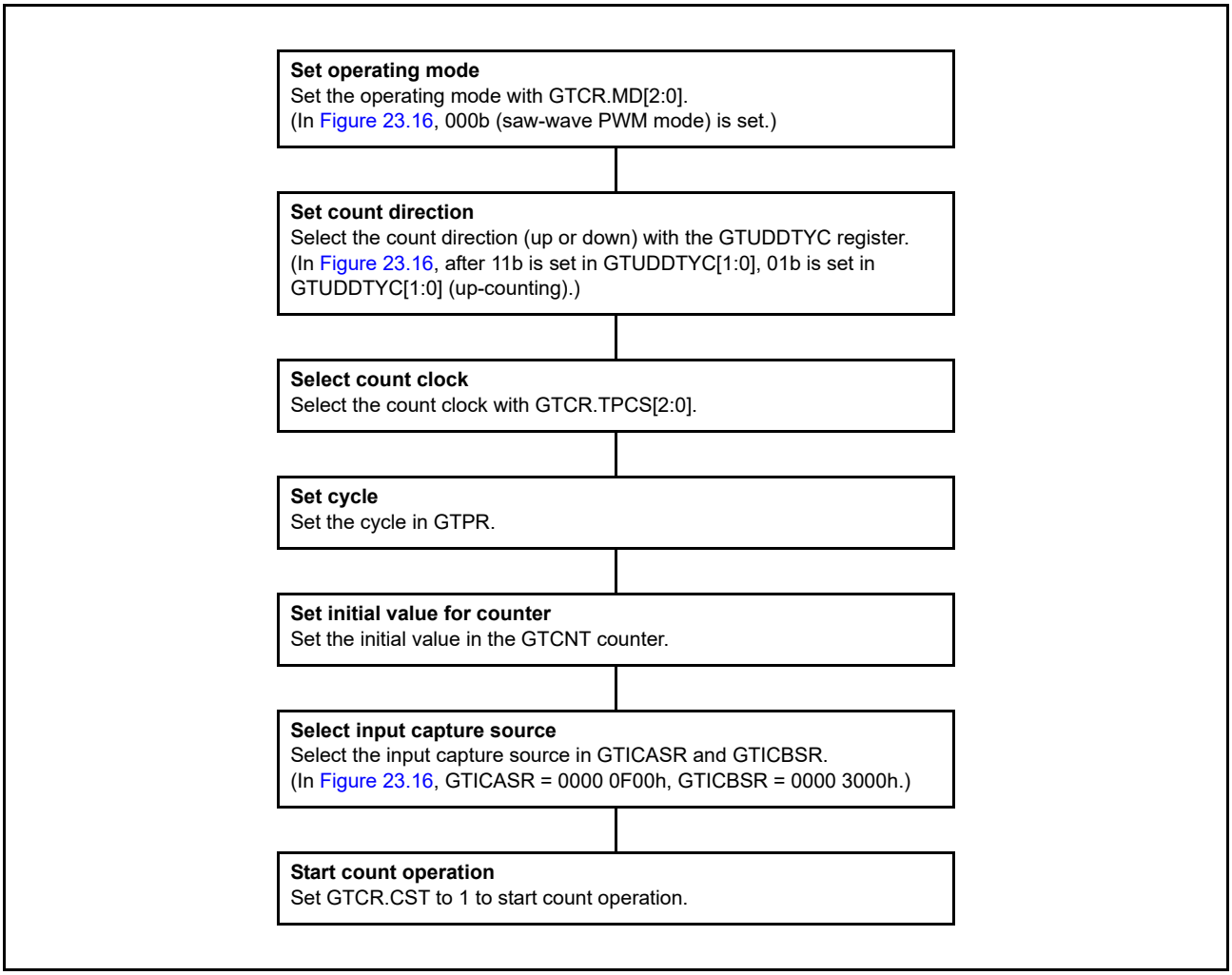


Figure 23.17 Example for setting input capture operation

23.3.2 Buffer Operation

The following buffer operations can be set with GTBER:

- GTPR and GTPBR
- GTCCRA, GTCCRC, and GTCCRD
- GTCCRB, GTCCRE, and GTCCRF.

23.3.2.1 GTPR Register Buffer Operation

GTPBR can function as a buffer register for GTPR. 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[23:0])
- Clear by software (when GTCSR.CCLR bit is 1 and GTCLR[n] bit is set to 1, n = channel number)

Figure 23.18 to Figure 23.20 show examples of GTPR buffer operation and Figure 23.21 shows an example for setting GTPR buffer operation.

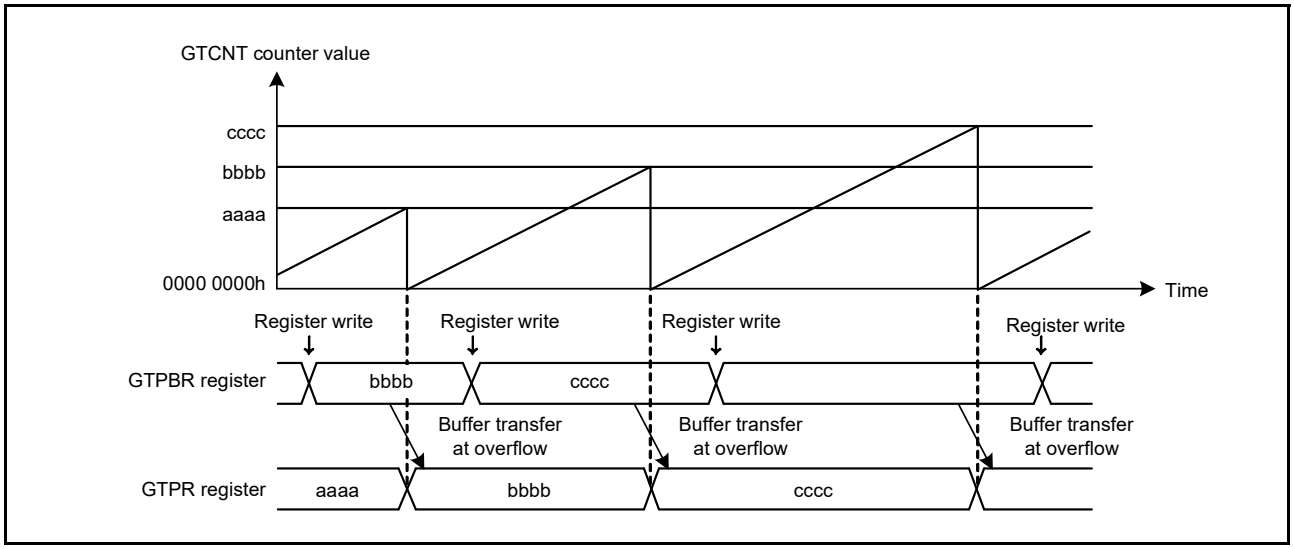


Figure 23.18 Example of GTPR buffer operation in saw waves in up-counting

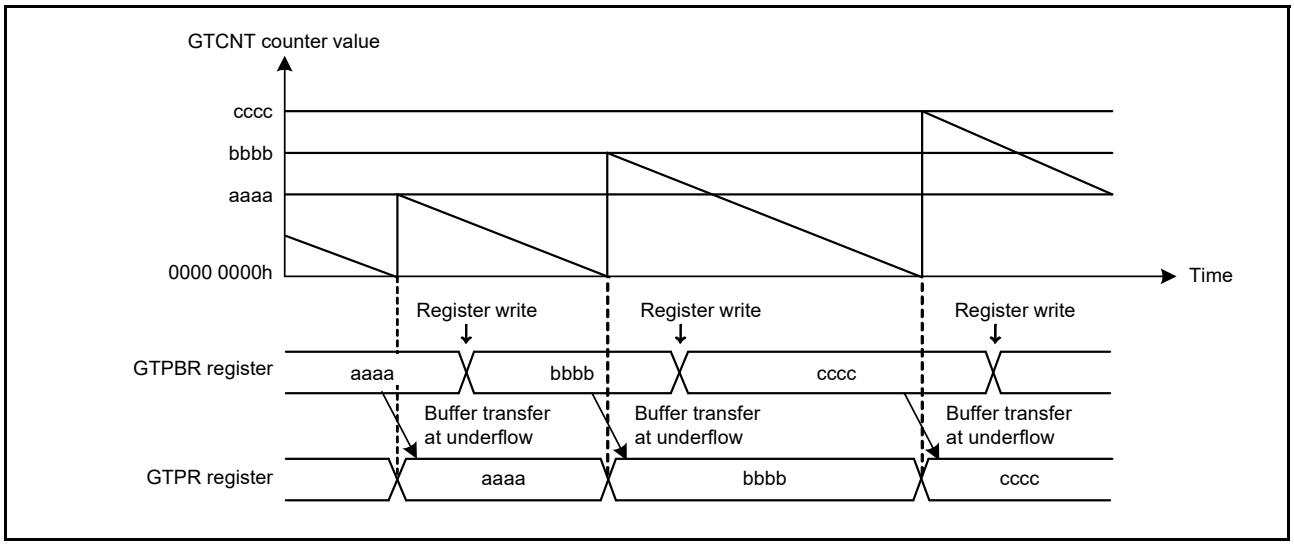


Figure 23.19 Example of GTPR buffer operation in saw waves in down-counting

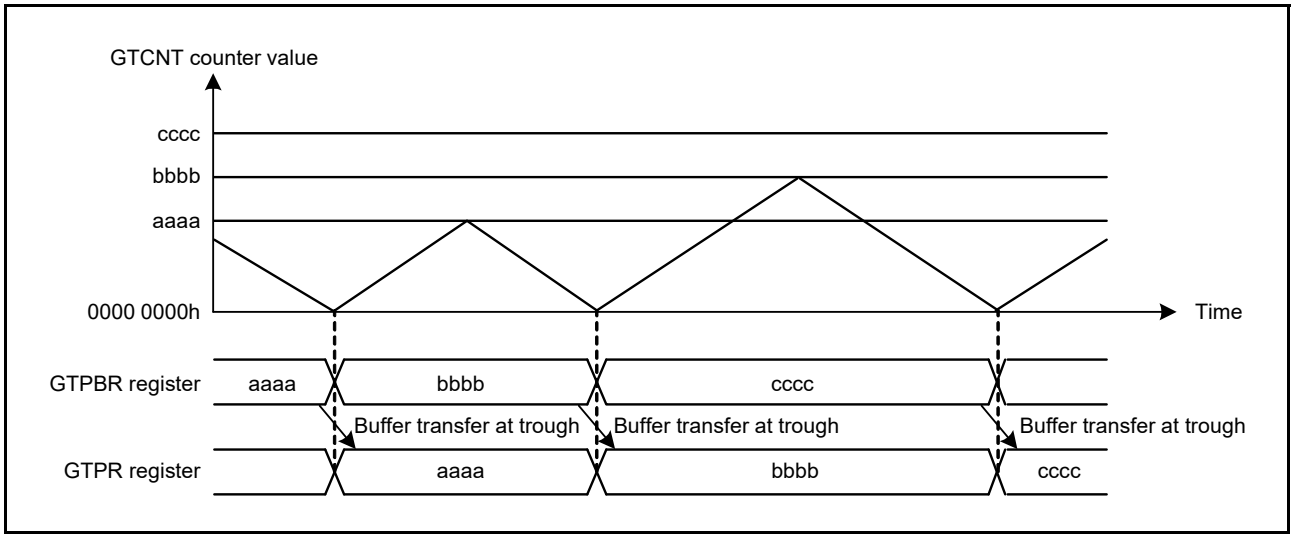


Figure 23.20 Example of GTPR buffer operation in triangle waves

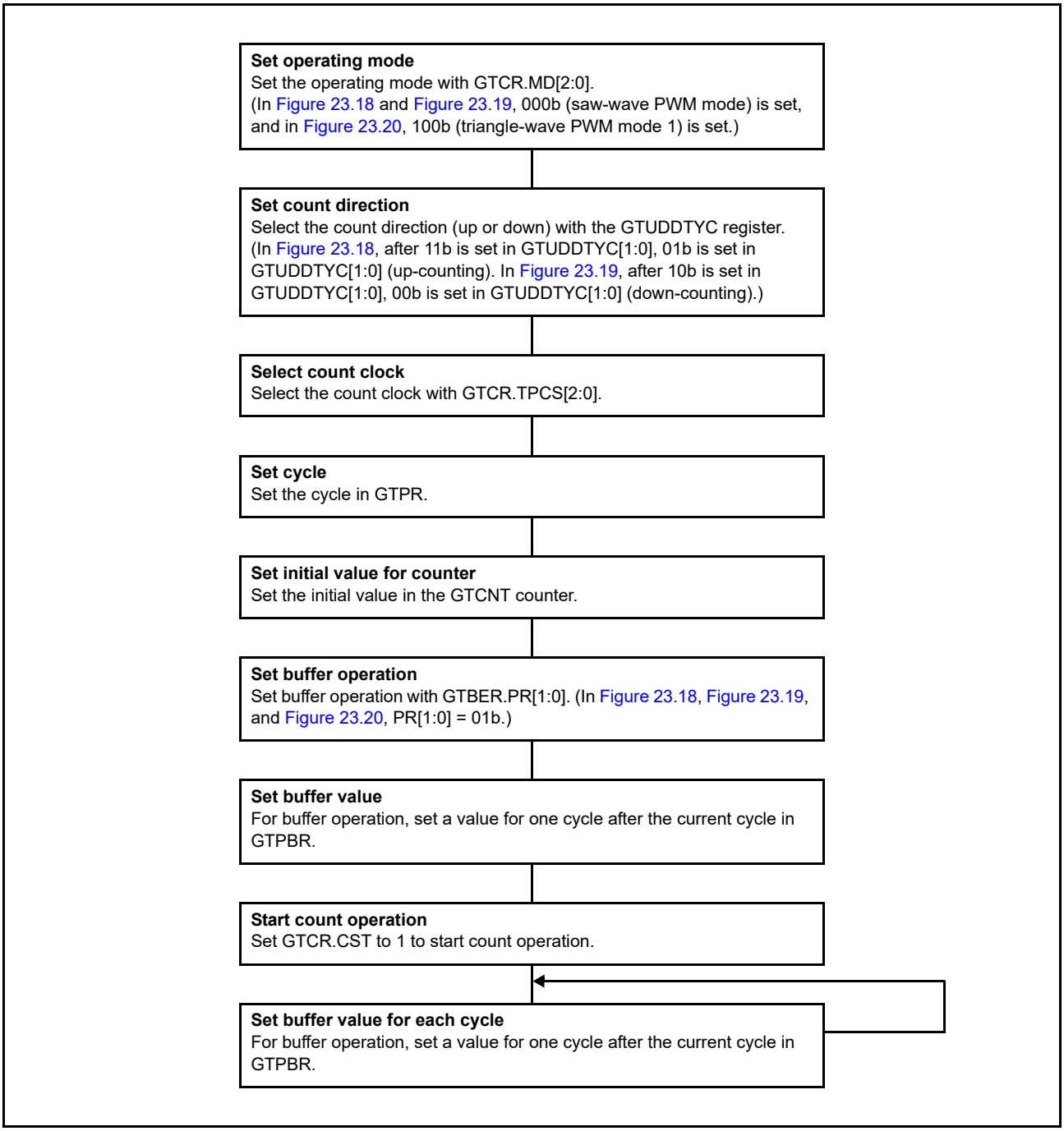


Figure 23.21 Example for setting GTPR buffer operation

23.3.2.2 Buffer Operation for GTCCRA and GTCCRB

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 GTBER.CCRA[1:0] or GTBER.CCRB[1:0] to 01b. To set GTCCRA or GTCCRB to not function as a buffer, set GTBER.CCRA[1:0] or GTBER.CCRB[1:0] to 00b.

(1) When GTCCRA or GTCCRB Functions as Output Compare Register

Buffer transfer has the following cases:

- 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 the same as in the case of section 23.3.2.1, GTPR Register Buffer Operation. In triangle-wave mode, buffer transfer is not performed by the counter clear.
- Forcible buffer transfer
  When GTBER.CCRSWT bit is set to 1 while the count operation stops, the GTCCRA and GTCCRB register buffer transfers are performed forcibly in saw-wave mode, in event count operation, and in triangle-wave mode. Additionally, buffer transfers from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B are performed in saw-wave 1 shot pulse mode or triangle-wave PWM mode 3.

Figure 23.22 to Figure 23.24 show examples of GTCCRA and GTCCRB buffer operation and Figure 23.25 shows an example for setting GTCCRA and GTCCRB buffer operation.

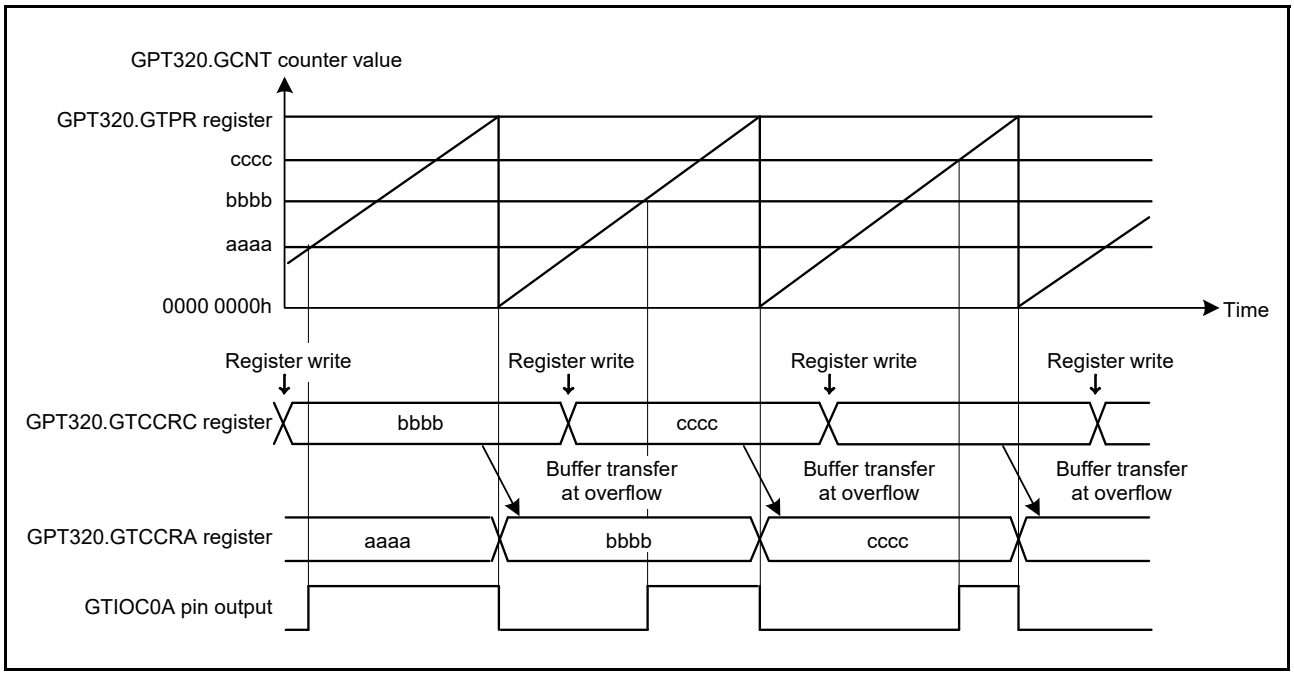


Figure 23.22 Example of GTCCRA and GTCCRB buffer operation (output compare, saw waves in up-counting, high output at GTCCRA compare match, low output at cycle end)

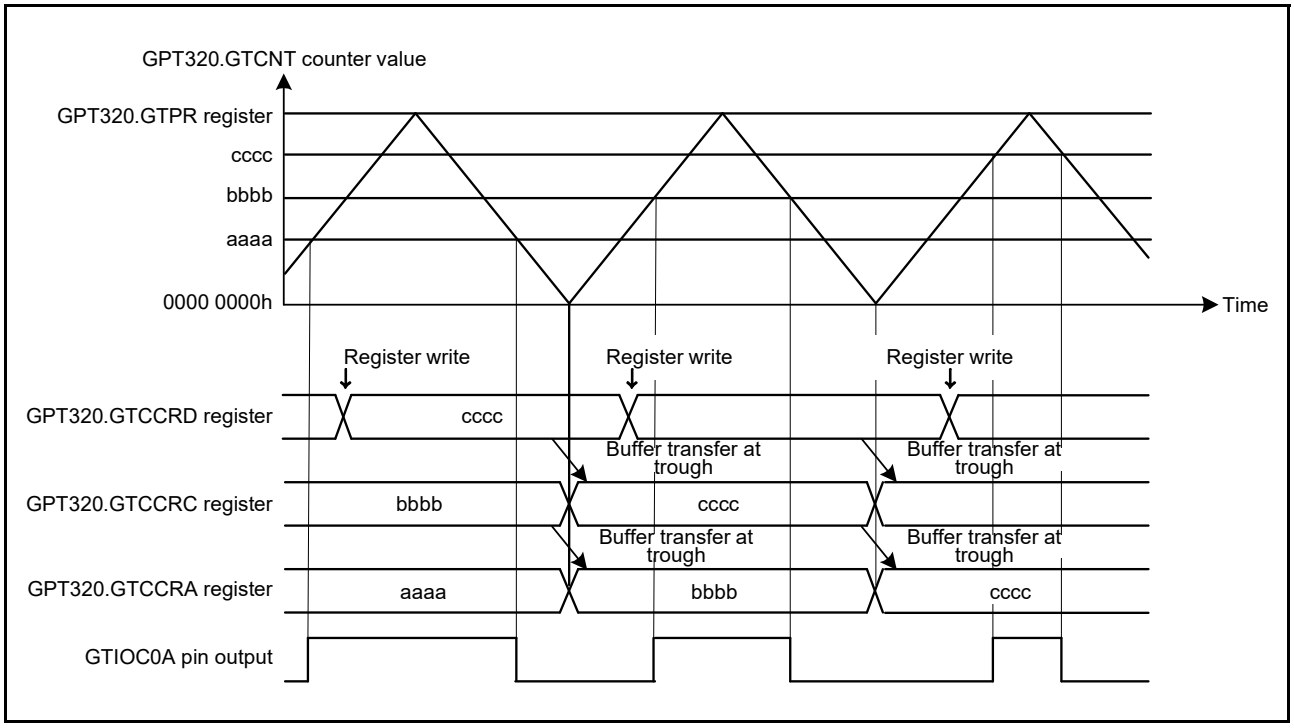


Figure 23.23 Example of GTCCRA and GTCCRB double buffer operation (output compare, triangle waves, buffer operation at trough, output toggled at GTCCRA compare match, output retained at cycle end)

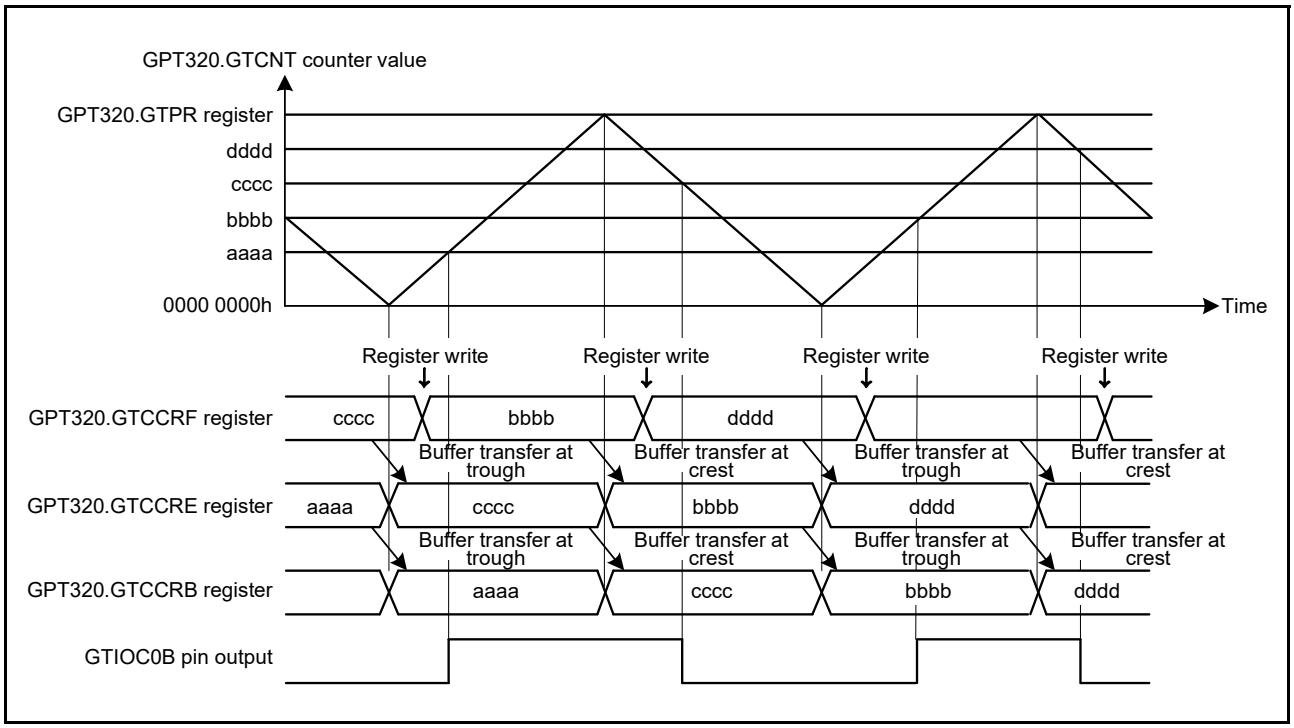


Figure 23.24 Example of GTCCRA and GTCCRB double buffer operation (output compare, triangle waves, buffer operation at both troughs and crests, output toggled at GTCCRB compare match, output retained at cycle end)

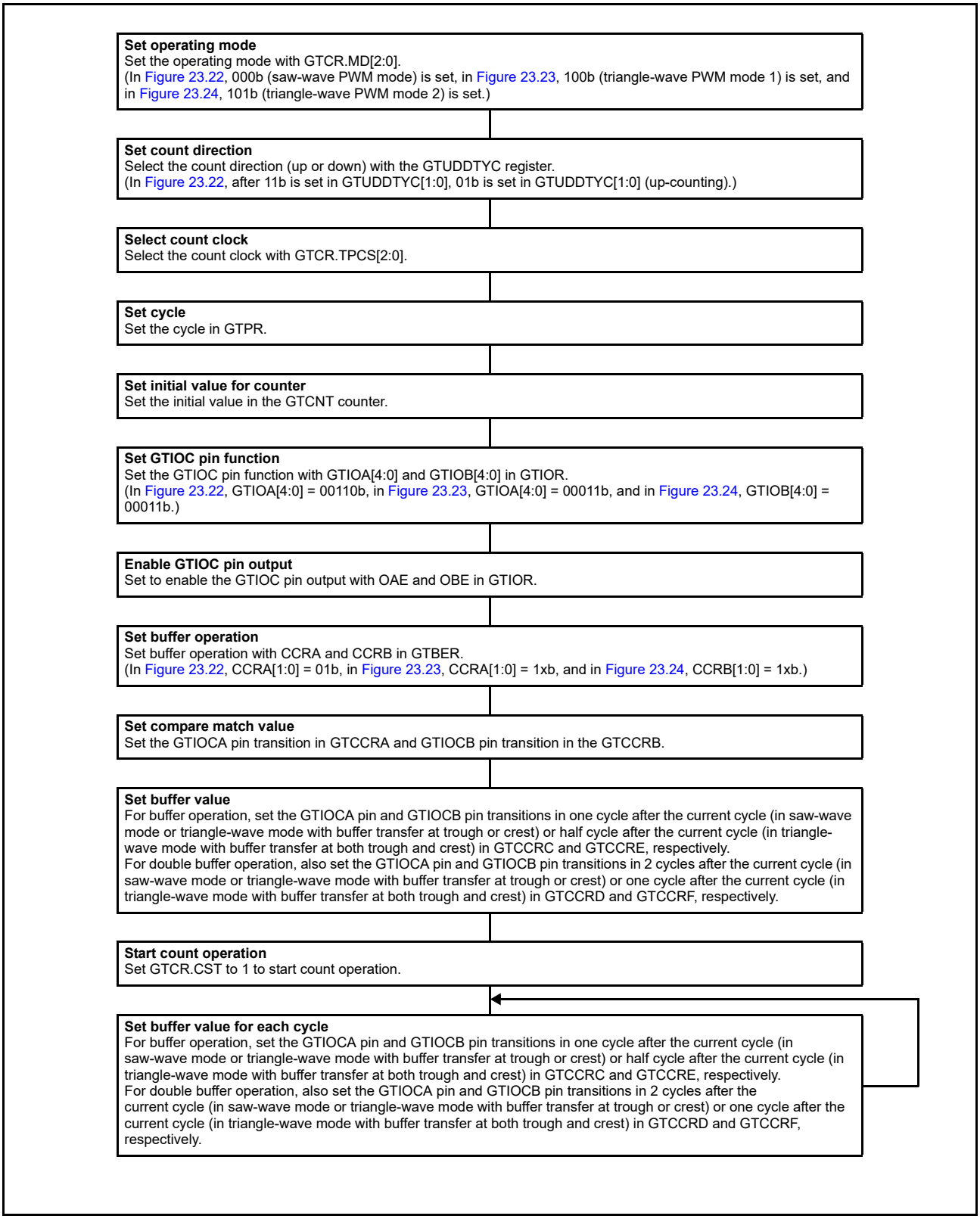


Figure 23.25 Example for setting GTCCRA and GTCCRB buffer operation for output compare

(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 23.26 and Figure 23.27 show examples of GTCCRA and GTCCRB buffer operation and Figure 23.28 shows an example for setting GTCCRA and GTCCRB buffer operation.

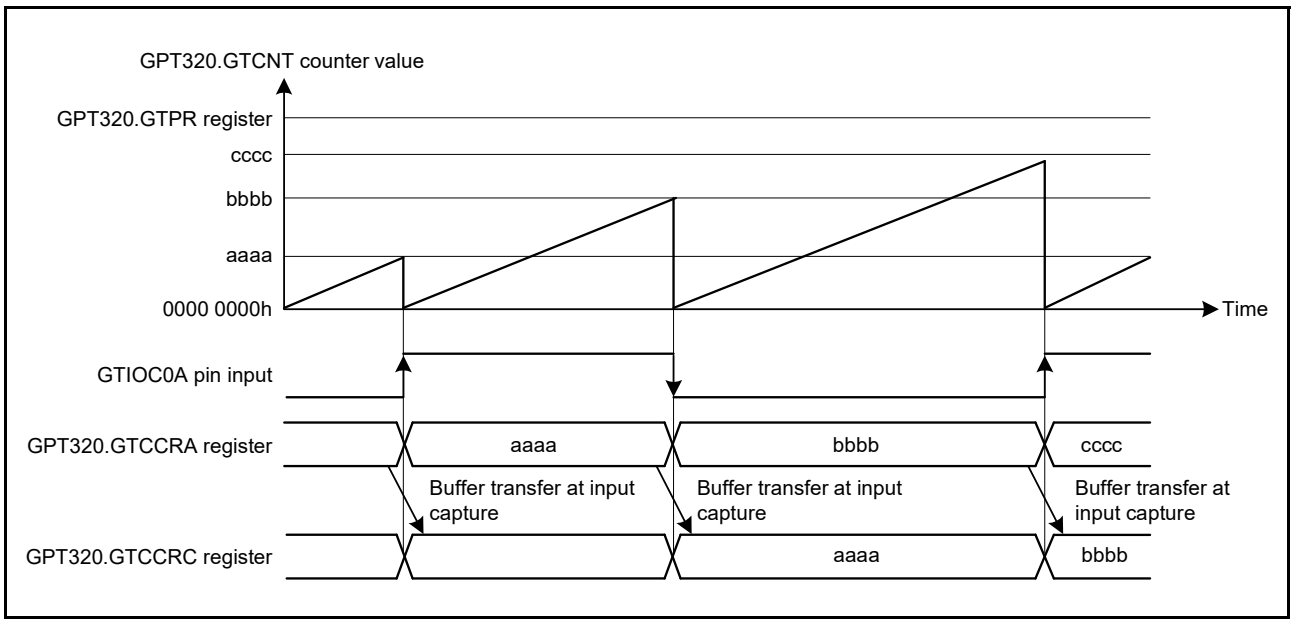


Figure 23.26 Example of GTCCRA and GTCCRB buffer operation (input capture at both edges of GTIOC0A input, saw waves in up-counting, GTCNT counter cleared at both edges of GTIOC0A input)

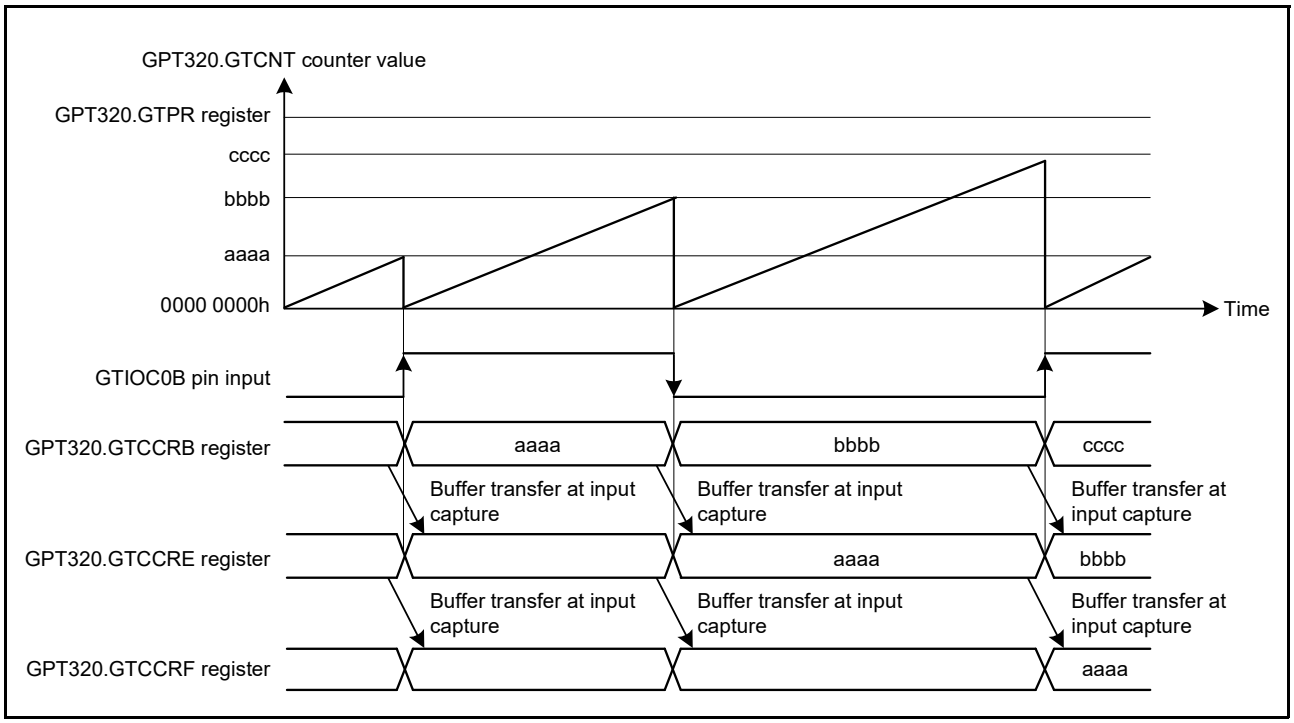


Figure 23.27 Example of GTCCRA and GTCCRB double buffer operation (input capture at both edges of GTIOC0B input, saw waves in up-counting, GTCNT counter cleared at both edges of GTIOC0B input)

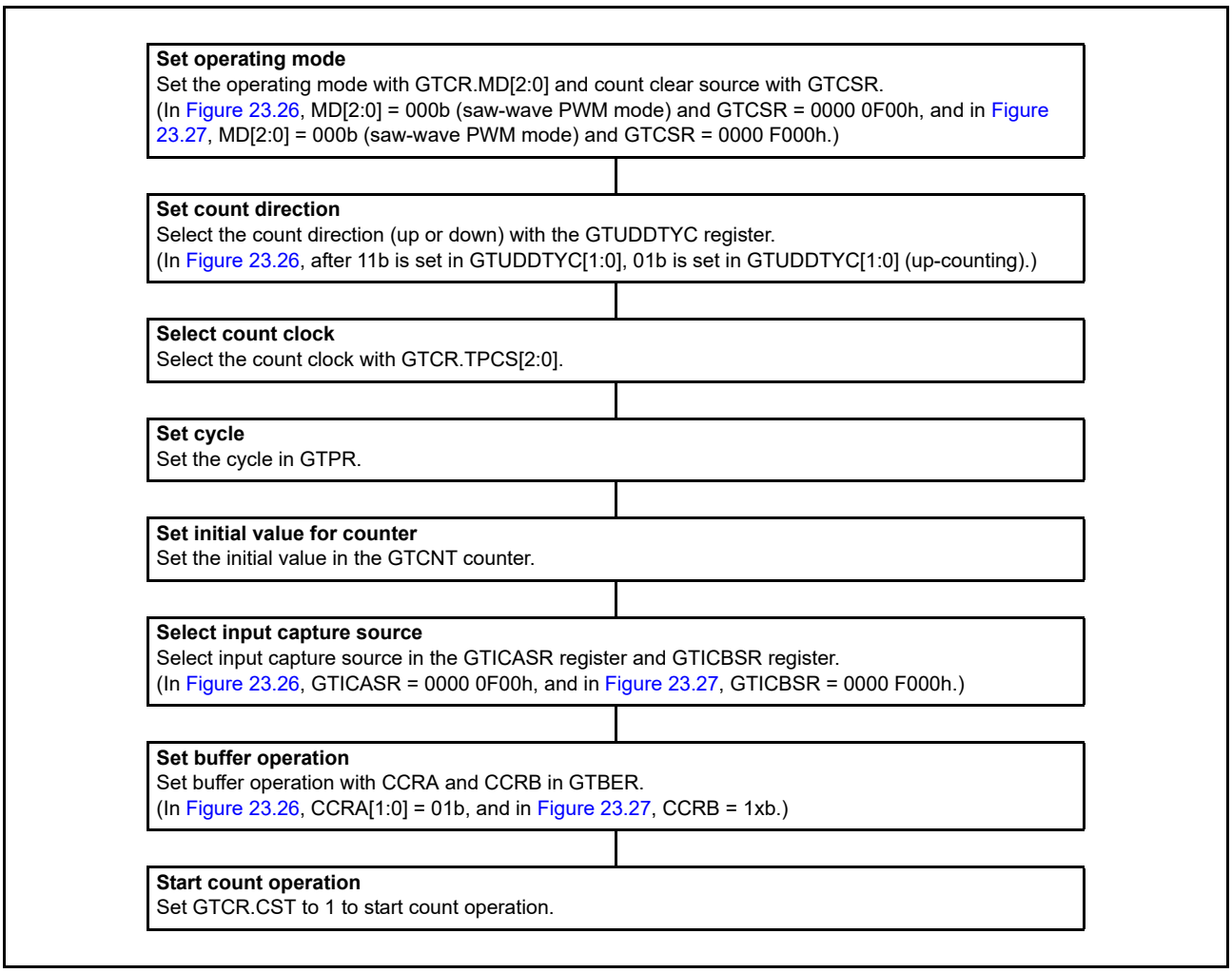


Figure 23.28 Example for setting GTCCRA and GTCCRB buffer operation (for input capture)

23.3.3 PWM Output Operating Mode

The GPT can output PWM waveforms to the GTIOCA or GTIOCB pin by a compare match between the GTCNT counter and GTCCRA or GTCCRB. By setting GTDTCR and GTDVU, the compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

23.3.3.1 Saw-Wave PWM Mode

In saw-wave PWM mode, GTCNT performs saw-wave (half-wave) operation by setting the cycle in GTPR. A PWM waveform is output to the GTIOCA or GTIOCB pin 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

Figure 23.29 shows an example of saw-wave PWM mode operation, and Figure 23.30 shows an example for setting saw-wave PWM mode.

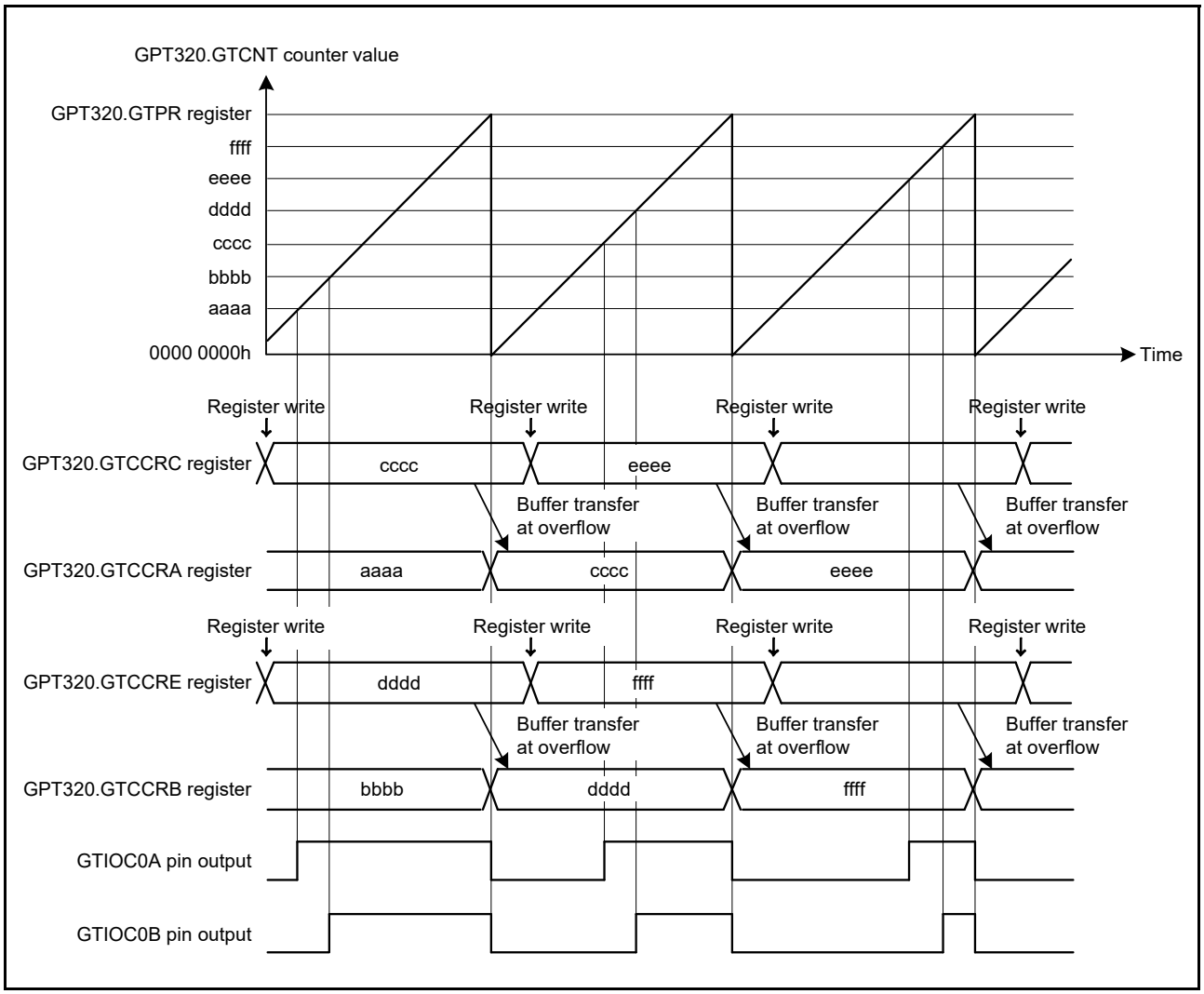


Figure 23.29 Example of saw-wave PWM mode operation (up-counting, buffer operation, high output at GTCCRA/GTCCRB compare match, low output at cycle end)

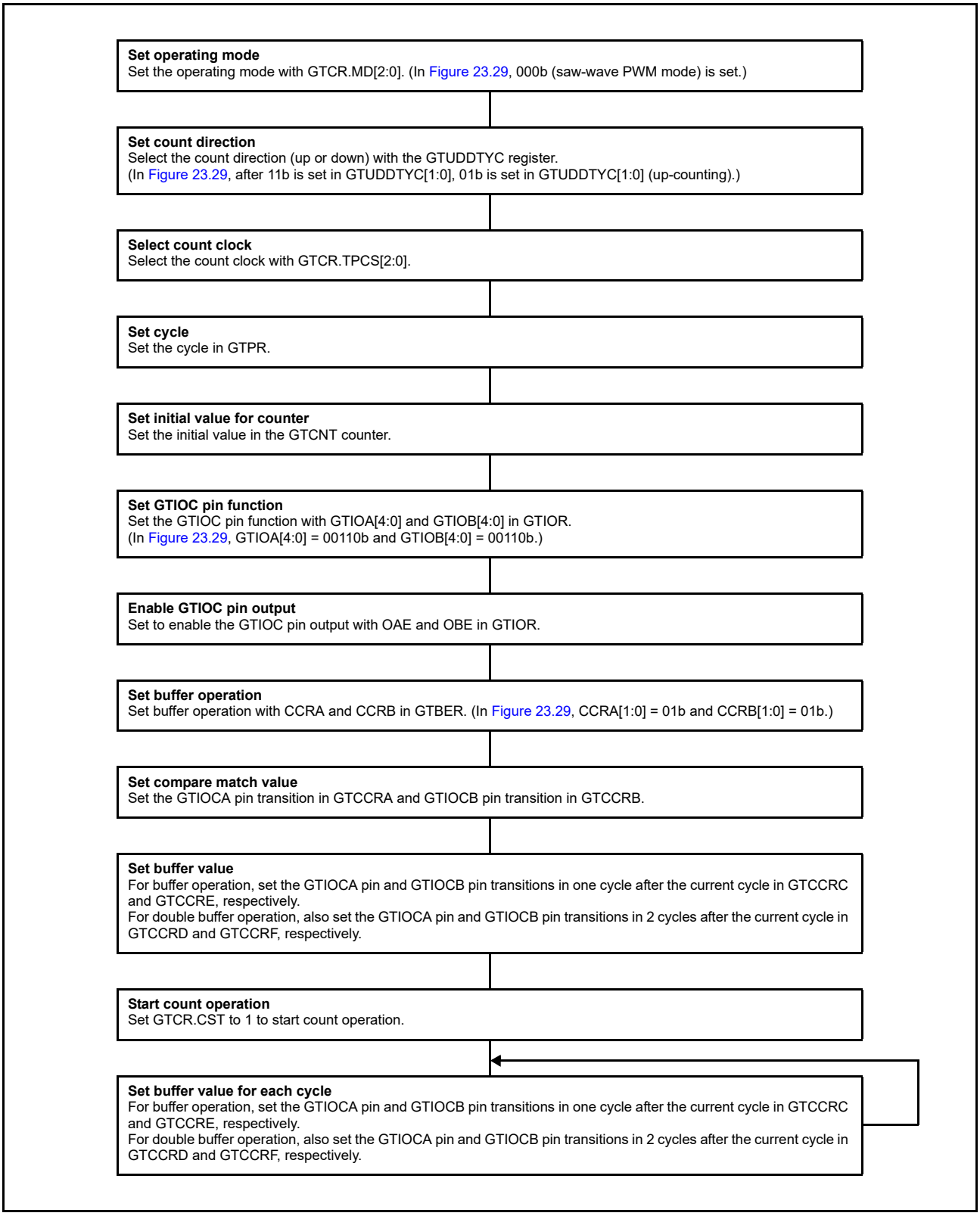


Figure 23.30 Example for setting saw-wave PWM mode

23.3.3.2 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 GTIOCA or GTIOCB pin 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 the following:

- 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 toggle 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 the count operation stops, 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 and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 23.31 shows an example of saw-wave one-shot pulse mode operation, and Figure 23.32 shows an example for setting saw-wave one-shot pulse mode.

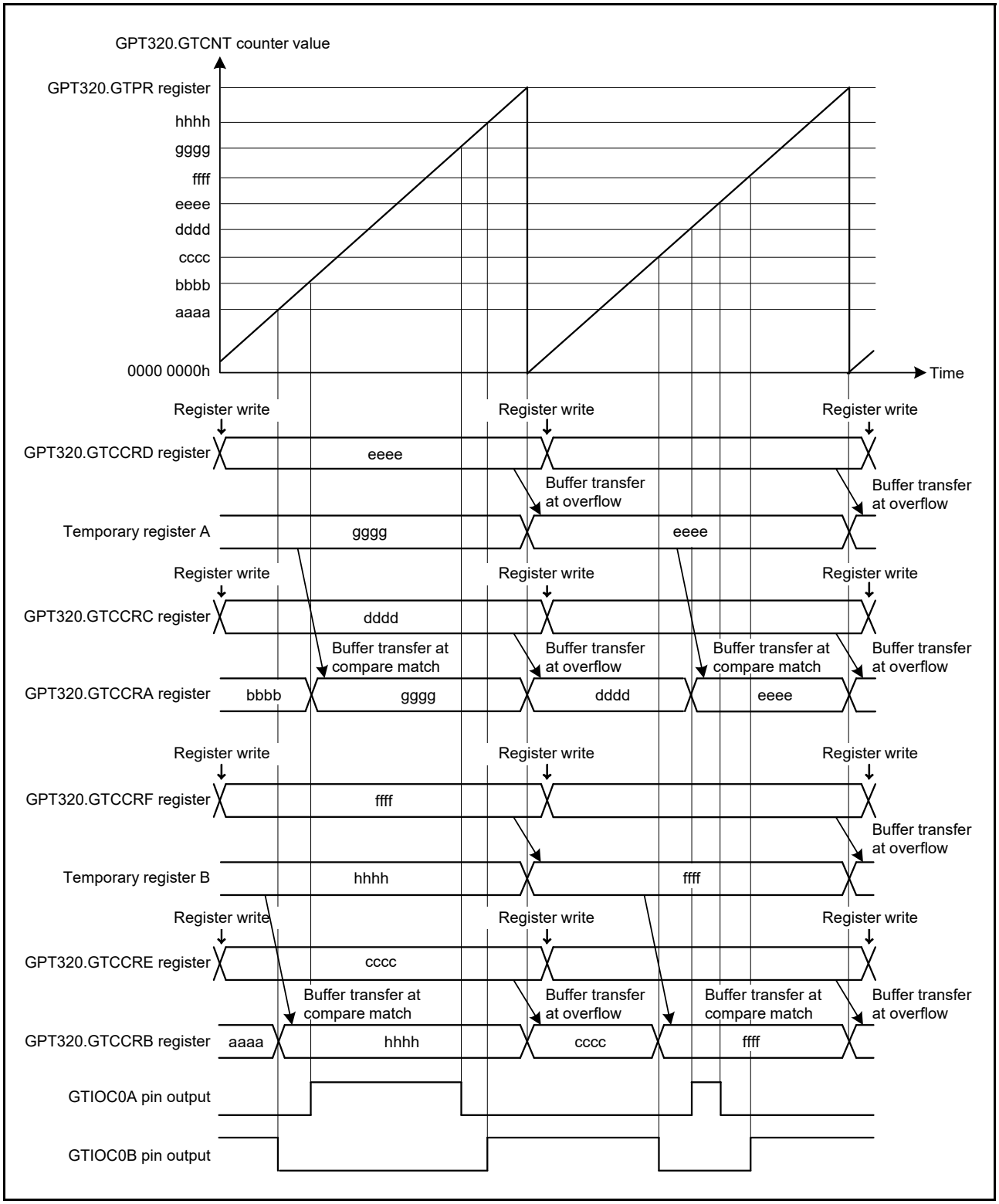


Figure 23.31 Example of saw-wave one-shot pulse mode operation (up-counting, low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/ GTCCRB compare match, output retained at cycle end)

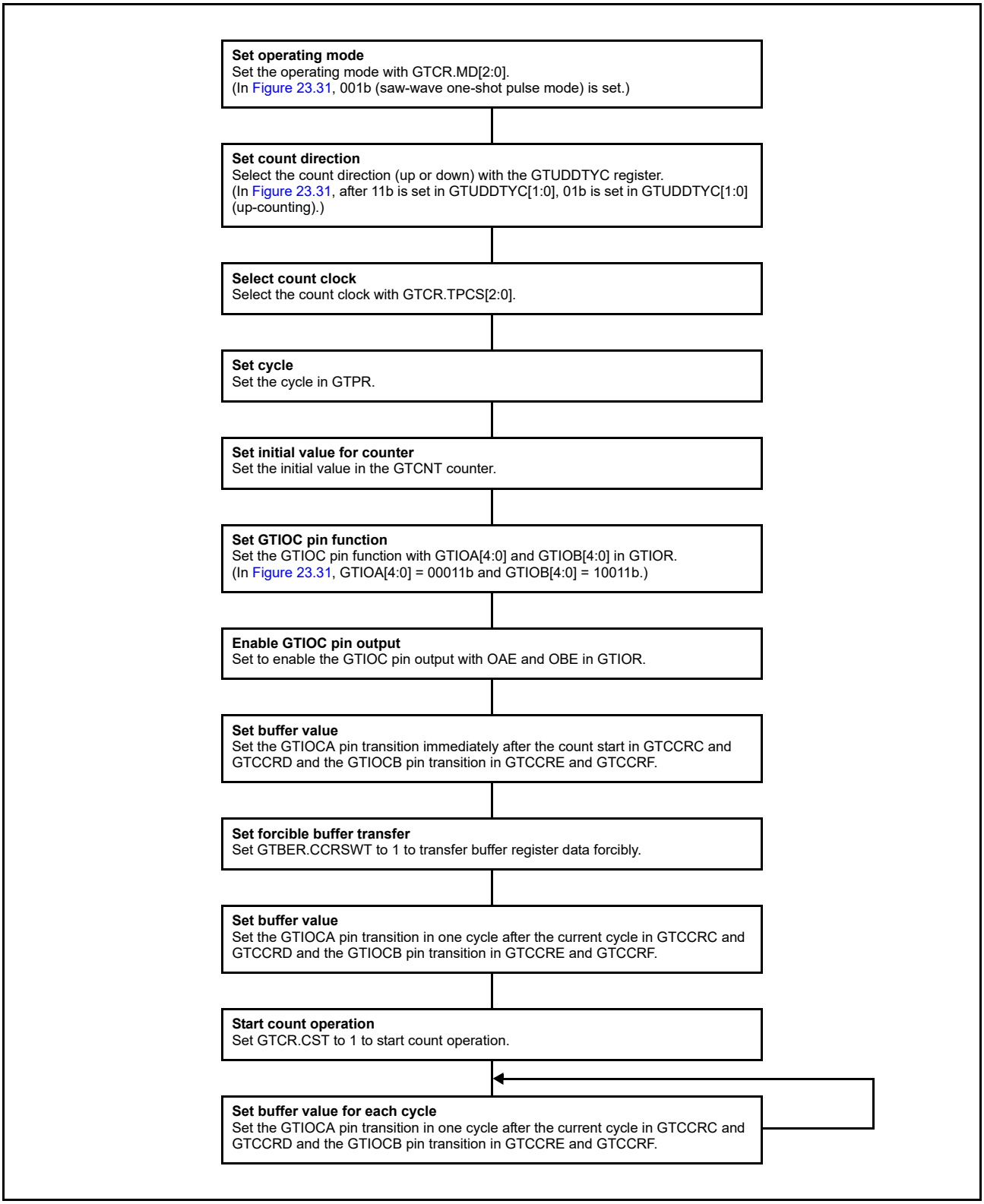


Figure 23.32 Example for setting saw-wave one-shot pulse mode

23.3.3.3 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 GTIOCA or GTIOCB pin 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 toggle output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 23.33 shows an example of a triangle-wave PWM mode 1 operation, and Figure 23.34 shows an example for setting a triangle-wave PWM mode 1.

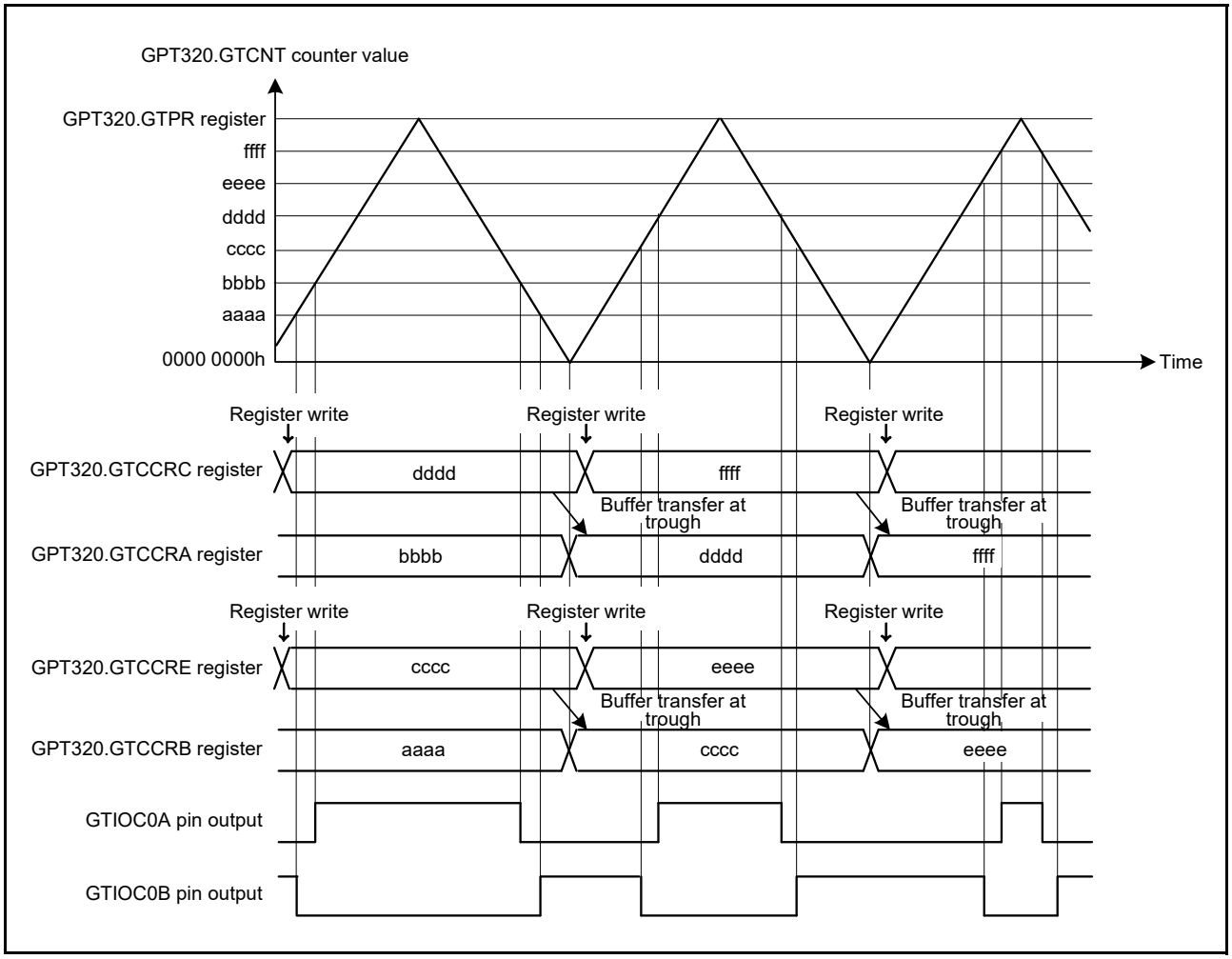


Figure 23.33 Example of triangle-wave PWM mode 1 operation (buffer operation, low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/GTCCRB register compare match, output retained at cycle end)

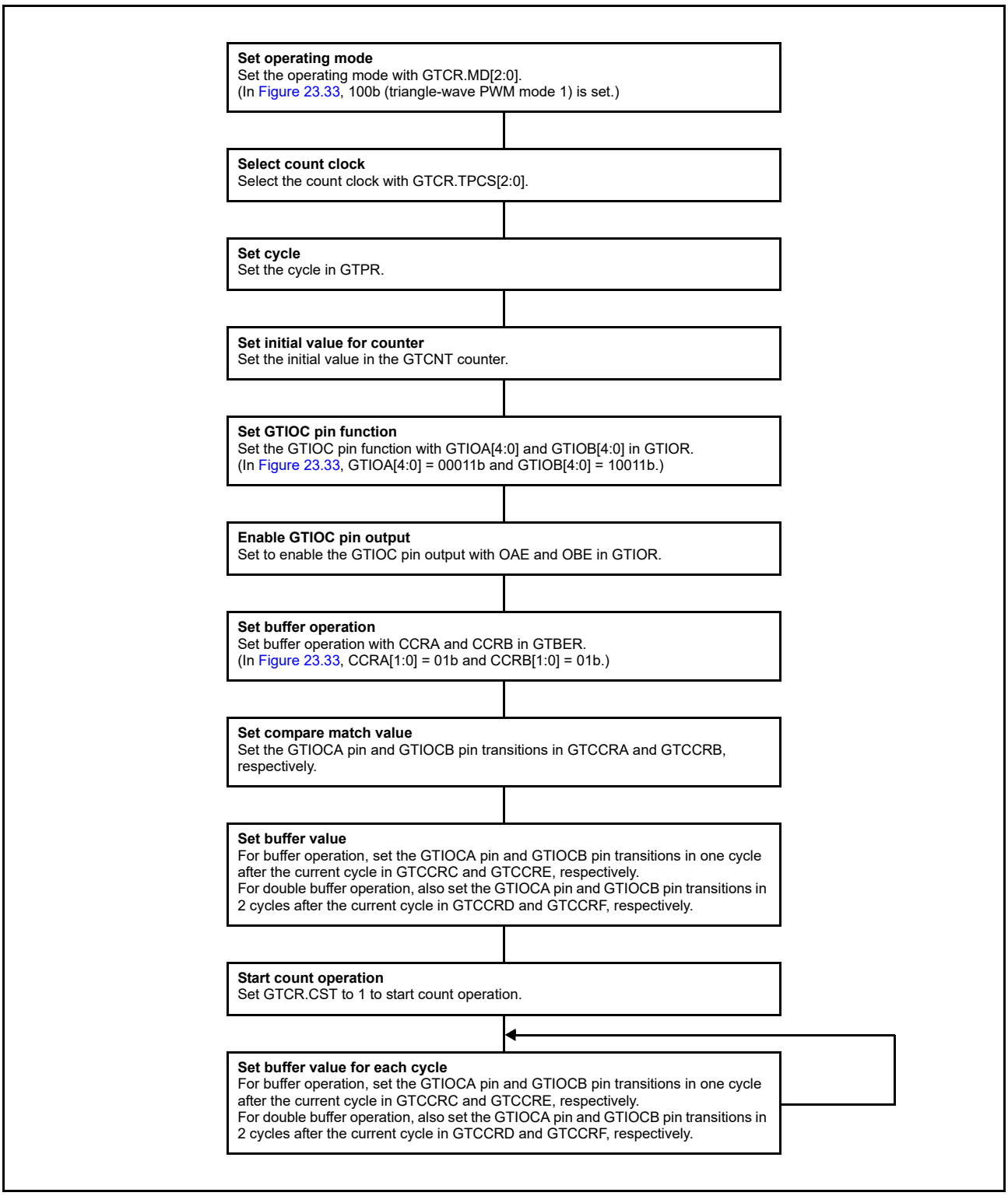


Figure 23.34 Example for setting triangle-wave PWM mode 1

23.3.3.4 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 GTIOCA or GTIOCB pin 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 and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 23.35 shows an example of triangle-wave PWM mode 2 operation, and Figure 23.36 shows an example for setting triangle-wave PWM mode 2.

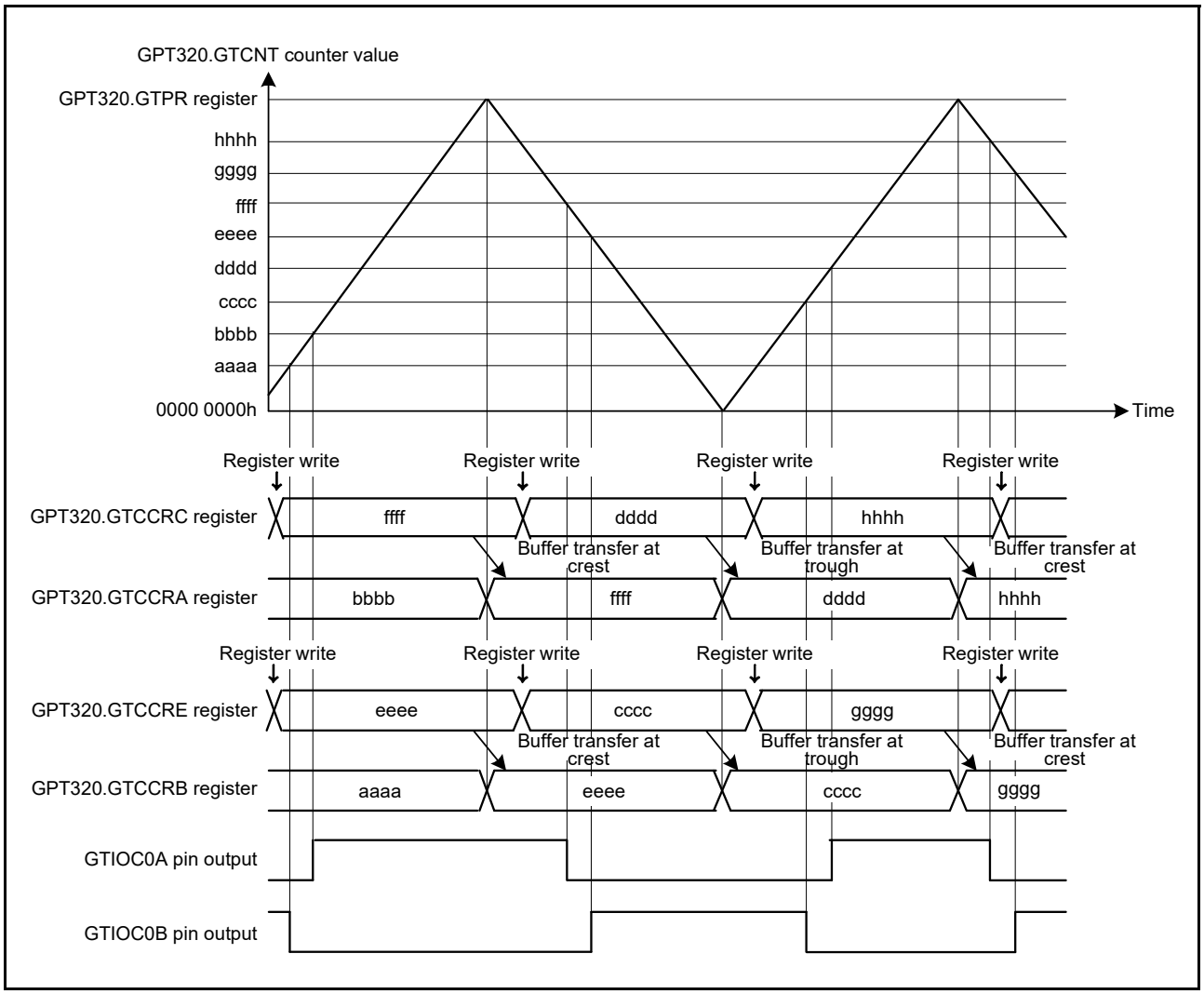


Figure 23.35 Example of triangle-wave PWM mode 2 operation (buffer operation, low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/GTCCRB compare match, output retained at cycle end)

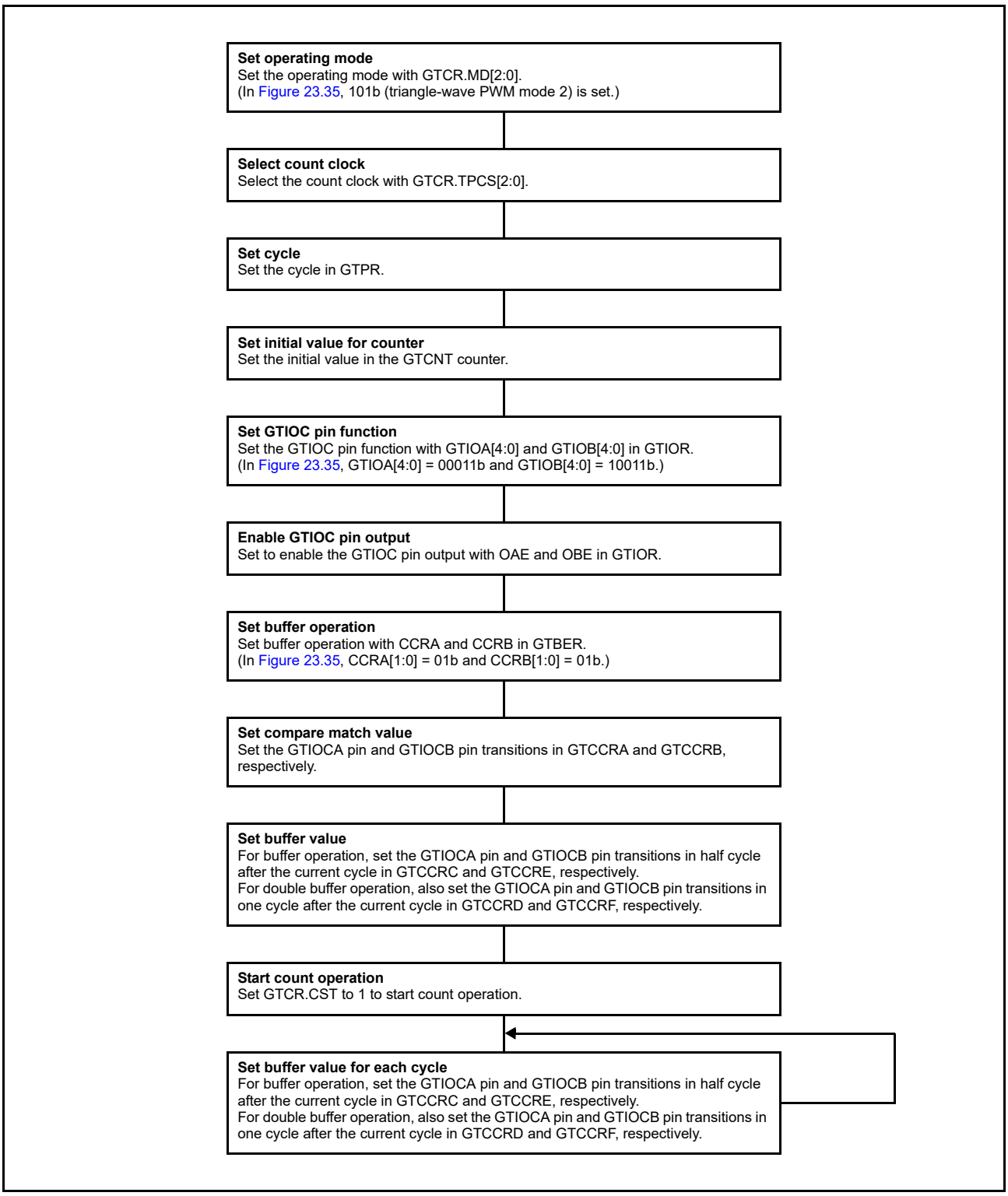


Figure 23.36 Example for setting triangle-wave PWM mode 2

23.3.3.5 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 GTIOCA or GTIOCB pin 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 toggle output separately for a compare match and for the cycle end according to the GTIOR setting. By setting GTDTCR and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 23.37 shows an example of triangle-wave PWM mode 3 operation, and Figure 23.38 shows an example for setting triangle-wave PWM mode 3.

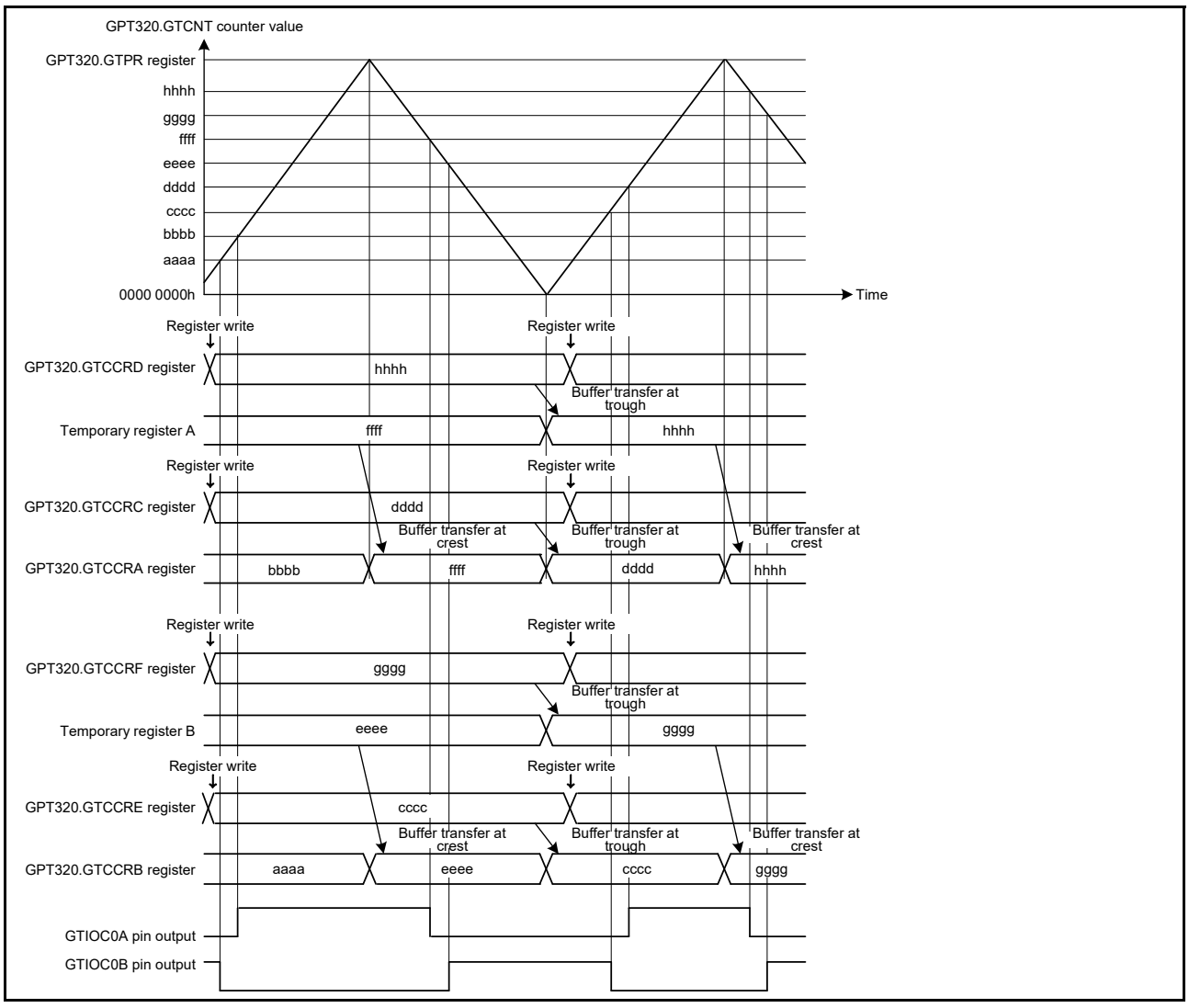


Figure 23.37 Example of triangle-wave PWM mode 3 operation (low output from the GTIOC0A pin and high output from the GTIOC0B pin at count start, output toggled at GTCCRA/GTCCRB compare match, output retained at cycle end)

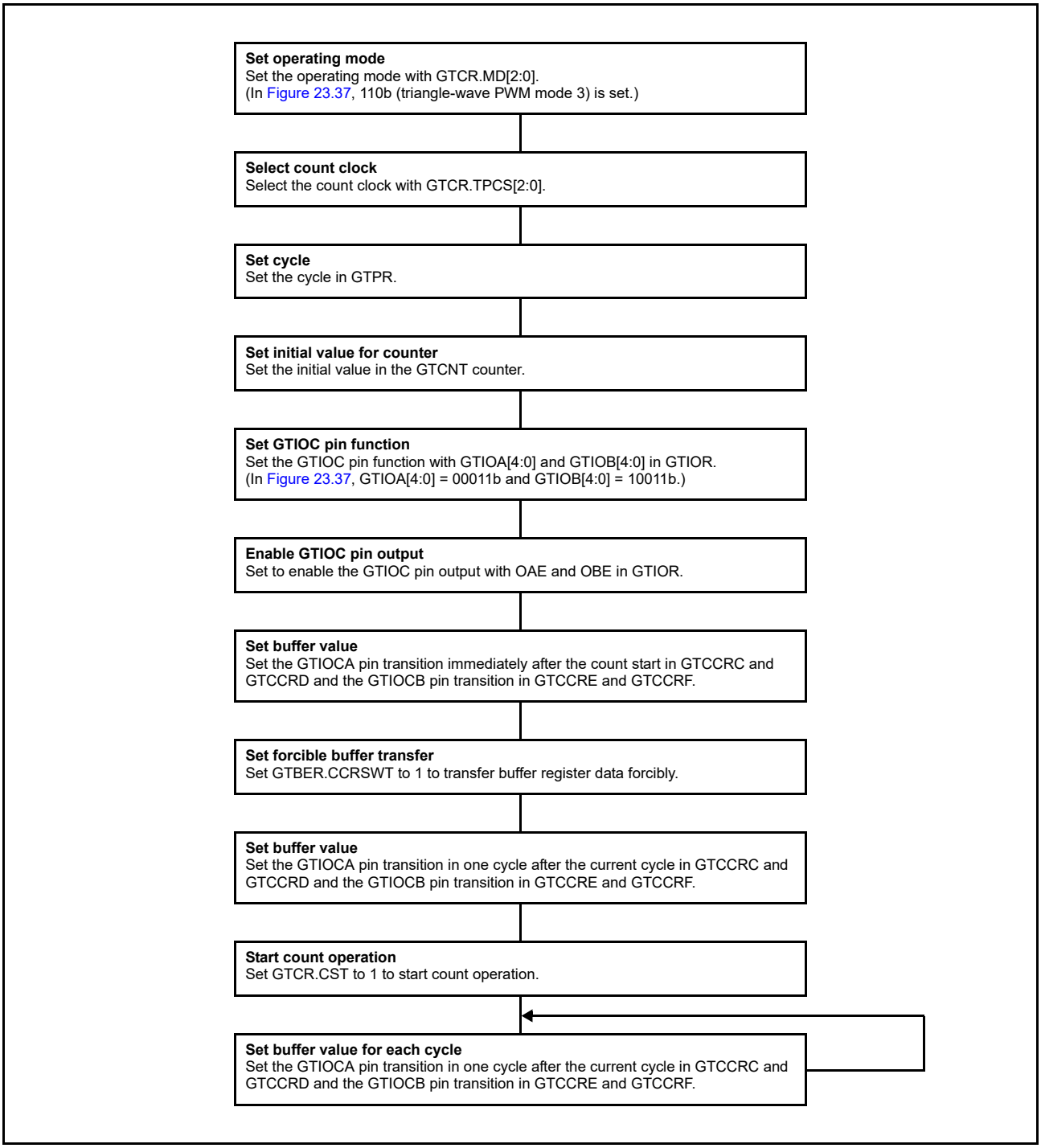


Figure 23.38 Example for setting triangle-wave PWM mode 3

23.3.4 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 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.

Writing to GTCCRB is prohibited when the automatic dead time setting function is used. Dead time setting beyond the cycle is also prohibited. Values for automatic dead time setting can be read from GTCCRB. The automatic dead time value setting to GTCCRB is performed at the next count clock cycle when registers that are used for calculating the automatic dead time value are updated.

Figure 23.39 to Figure 23.42 show examples of automatic dead time setting function operation. Figure 23.43 and Figure 23.44 show the setting examples.

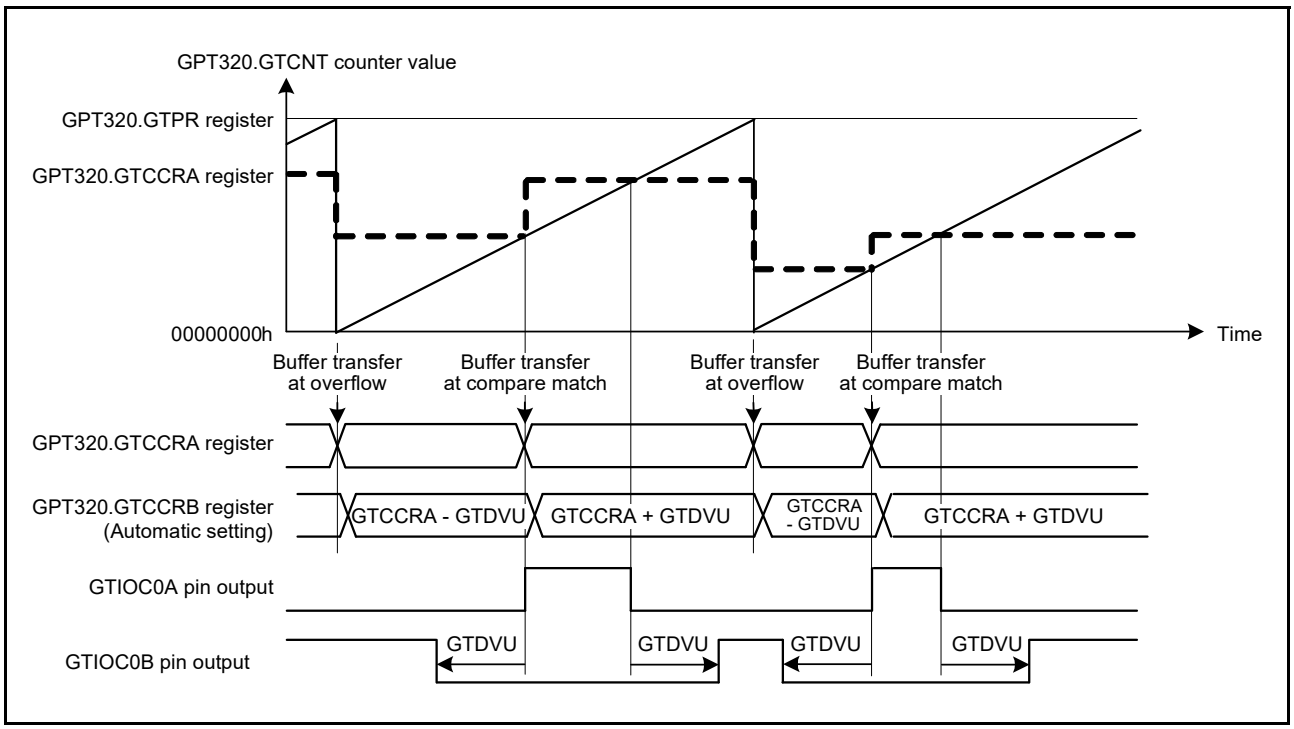


Figure 23.39 Example of automatic dead time setting function operation (saw-wave one-shot pulse mode, up-counting, active-high)

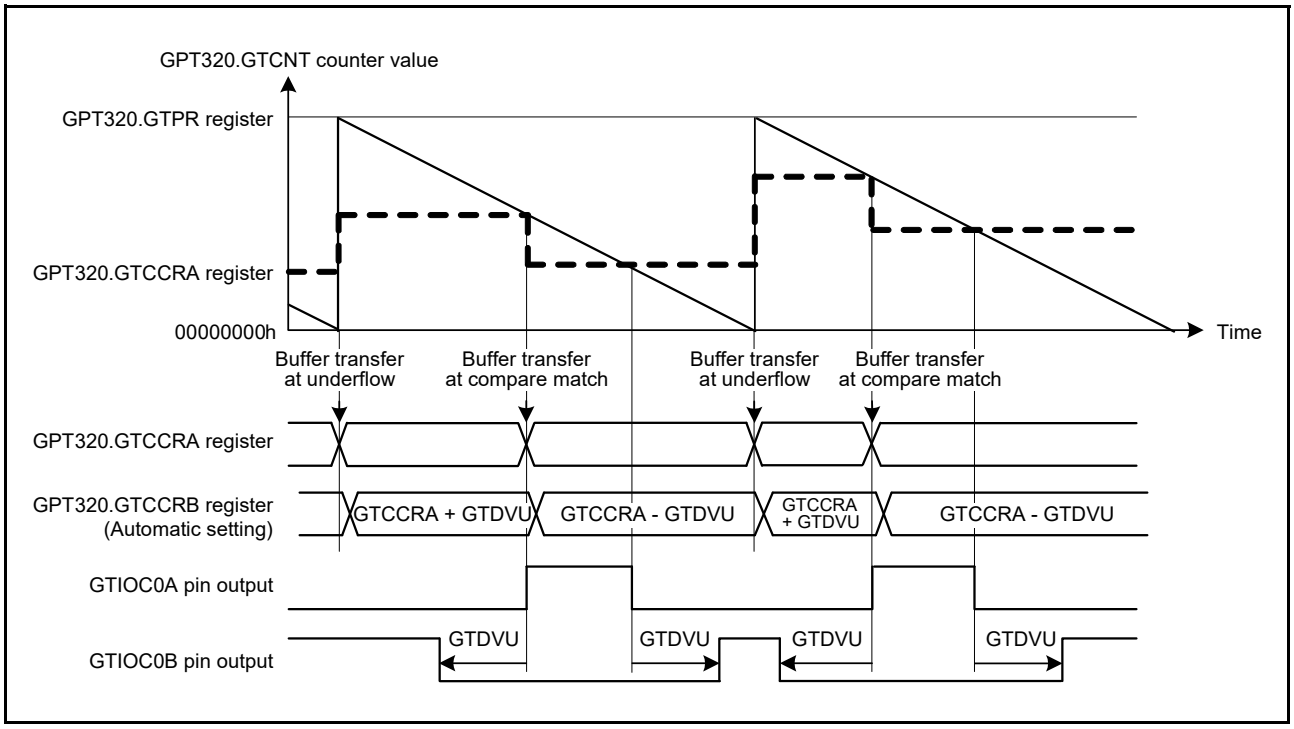


Figure 23.40 Example of automatic dead time setting function operation (saw-wave one-shot pulse mode, down-counting, active-high)

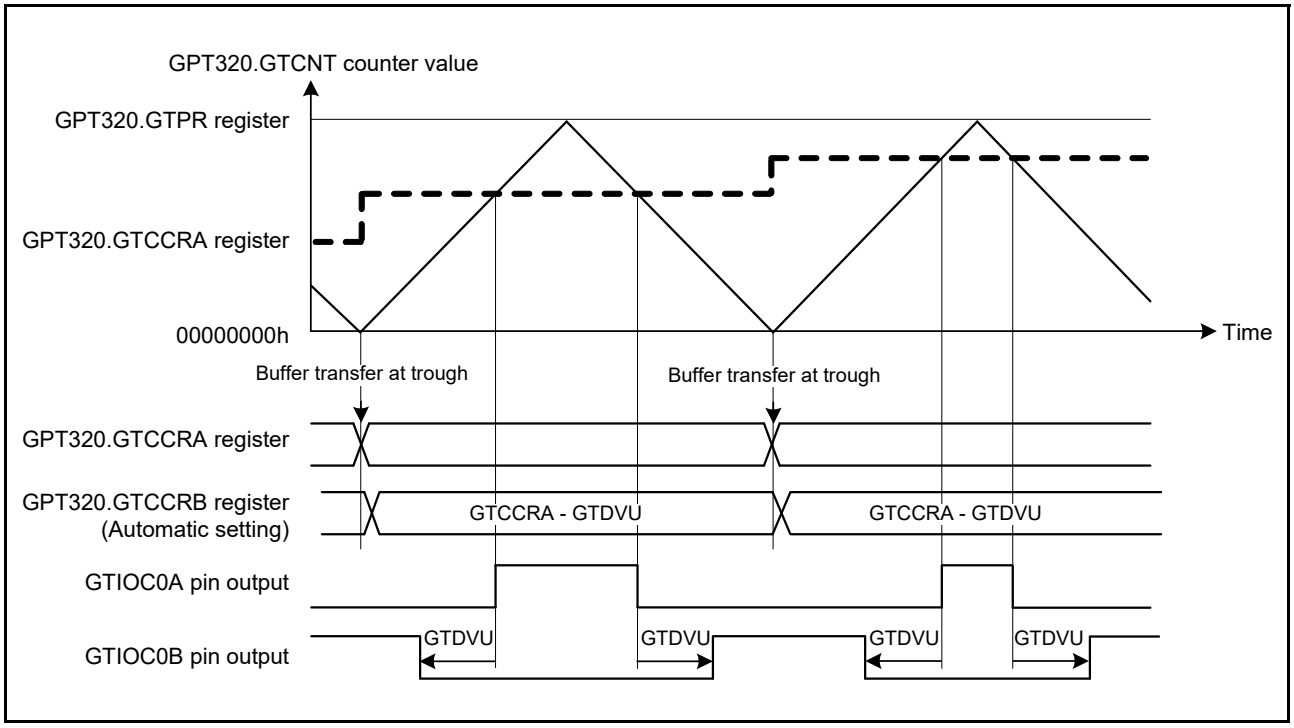


Figure 23.41 Example of automatic compare-match value setting function with dead time (triangle-wave PWM mode 1, active-high)

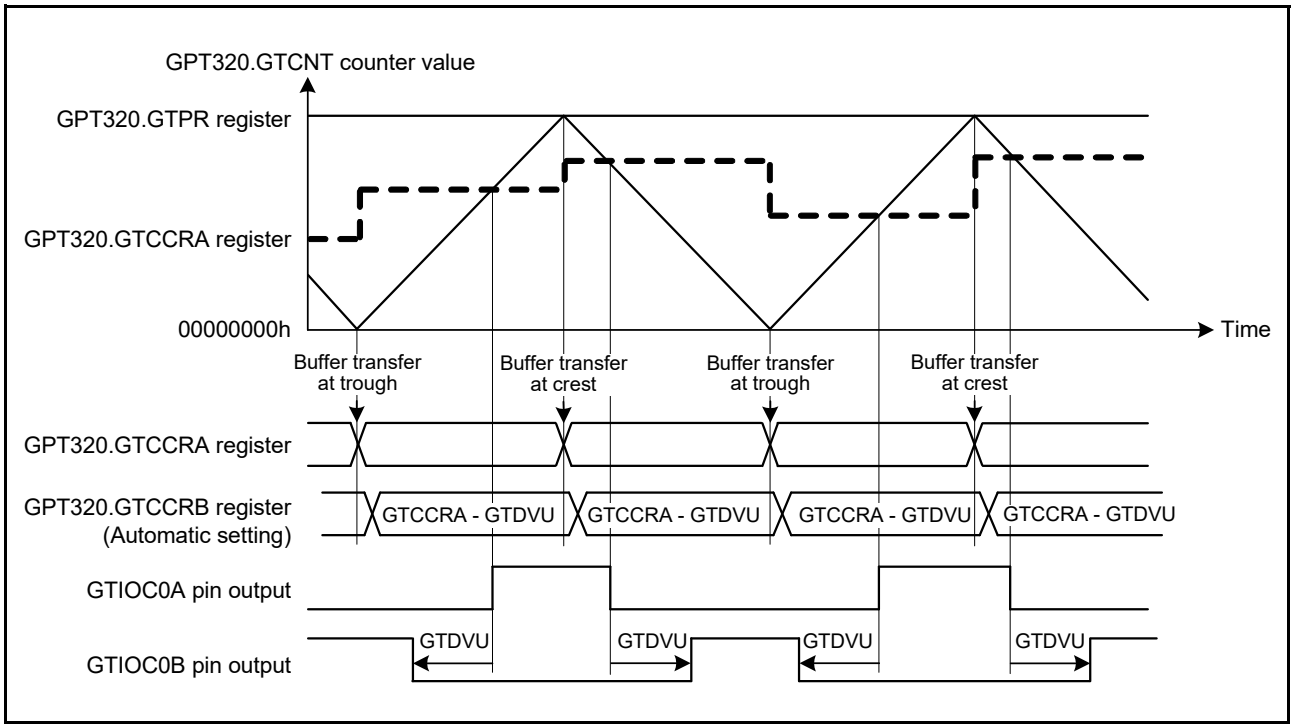


Figure 23.42 Example of automatic compare-match value setting function with dead time (triangle-wave PWM mode 2 or 3, active-high)

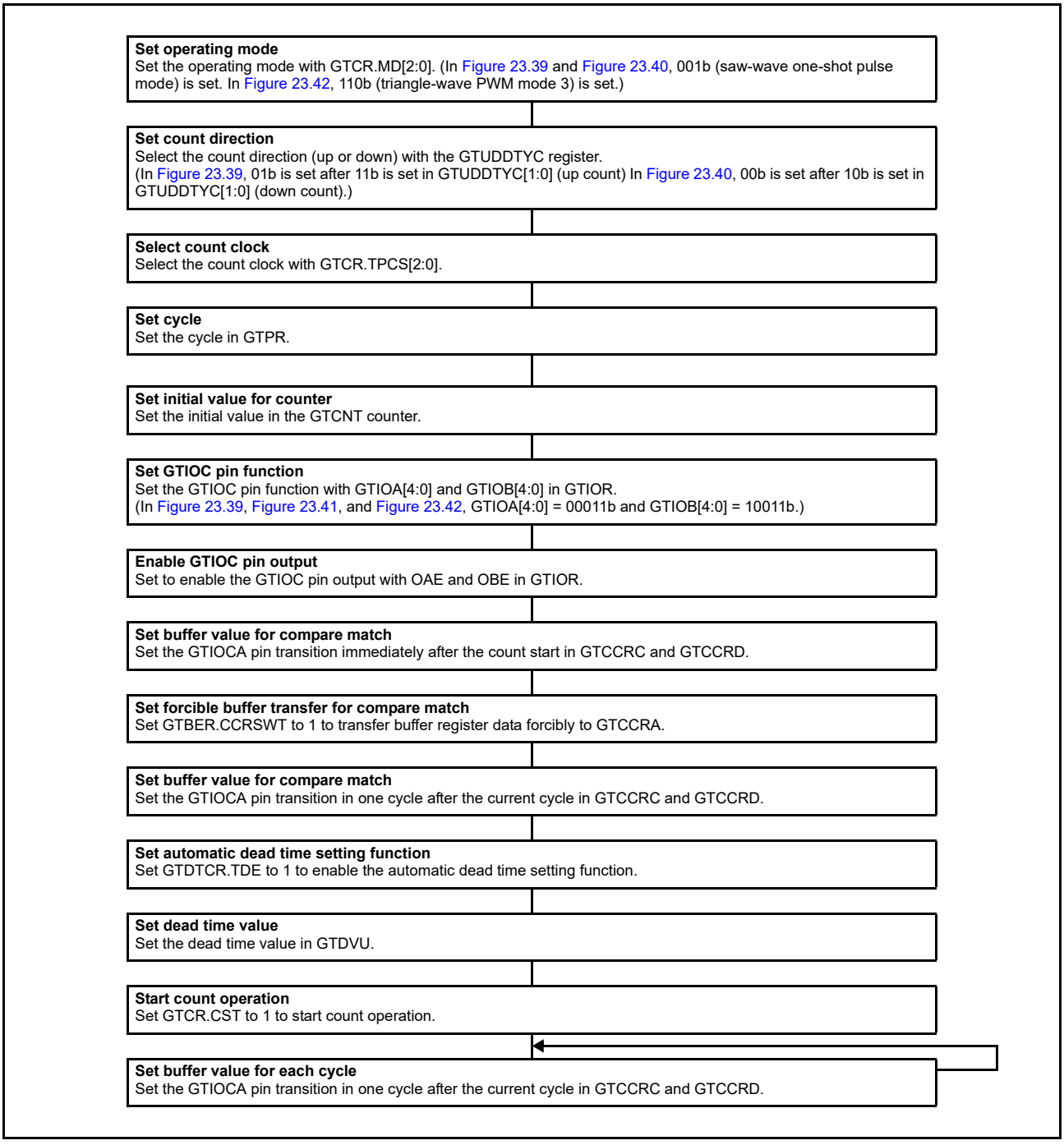


Figure 23.43 Example for setting automatic dead time setting function (saw-wave one-shot pulse mode, triangle-wave PWM mode 3)

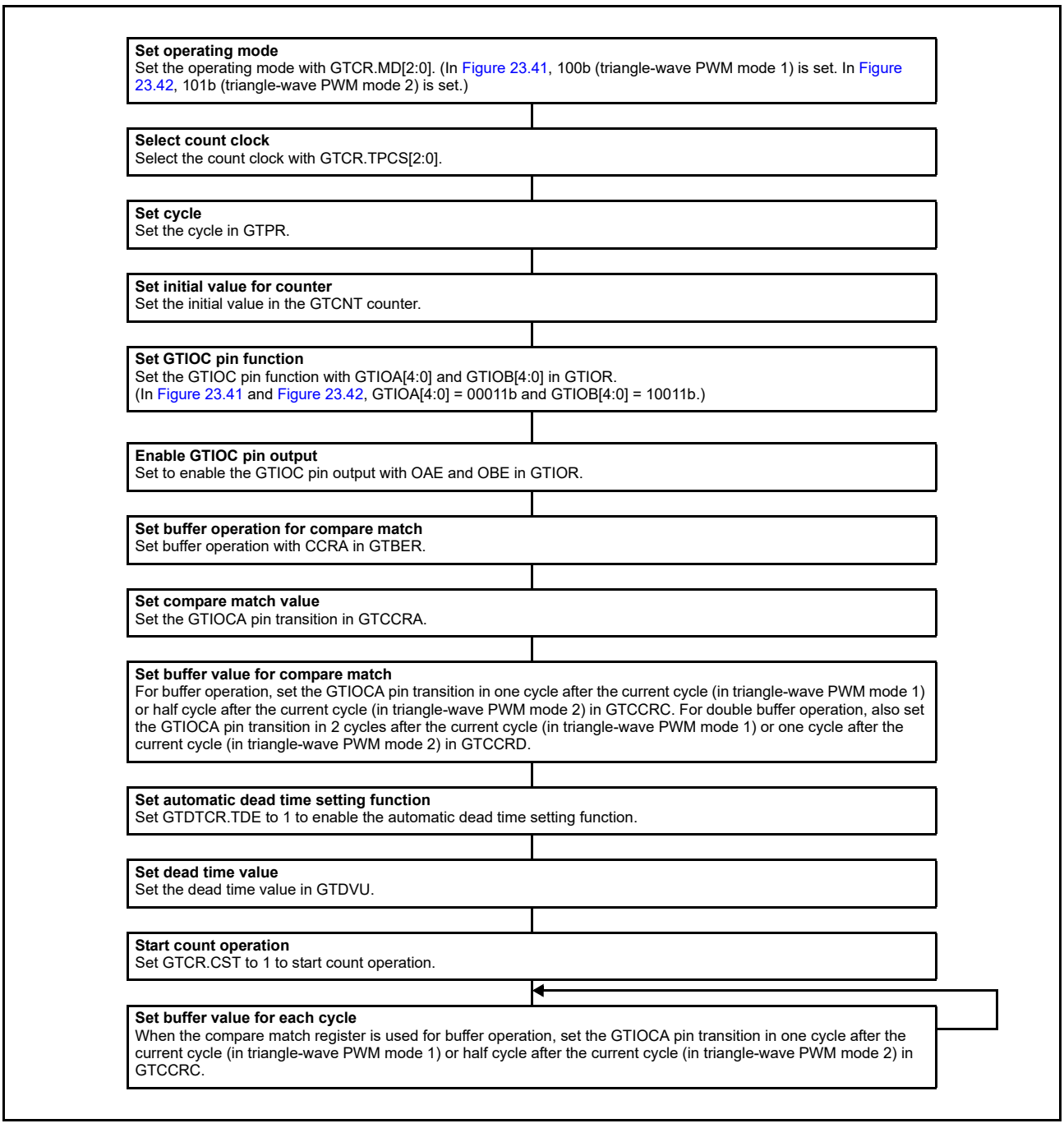


Figure 23.44 Example for setting automatic dead time setting function (triangle-wave PWM mode 1 or 2)

23.3.5 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 changes 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 stops, 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 to the count cycle during up-counting and the GTPR value before the start of down-counting is reflected during down-counting.

Figure 23.45 shows an example of count direction changing function operation.

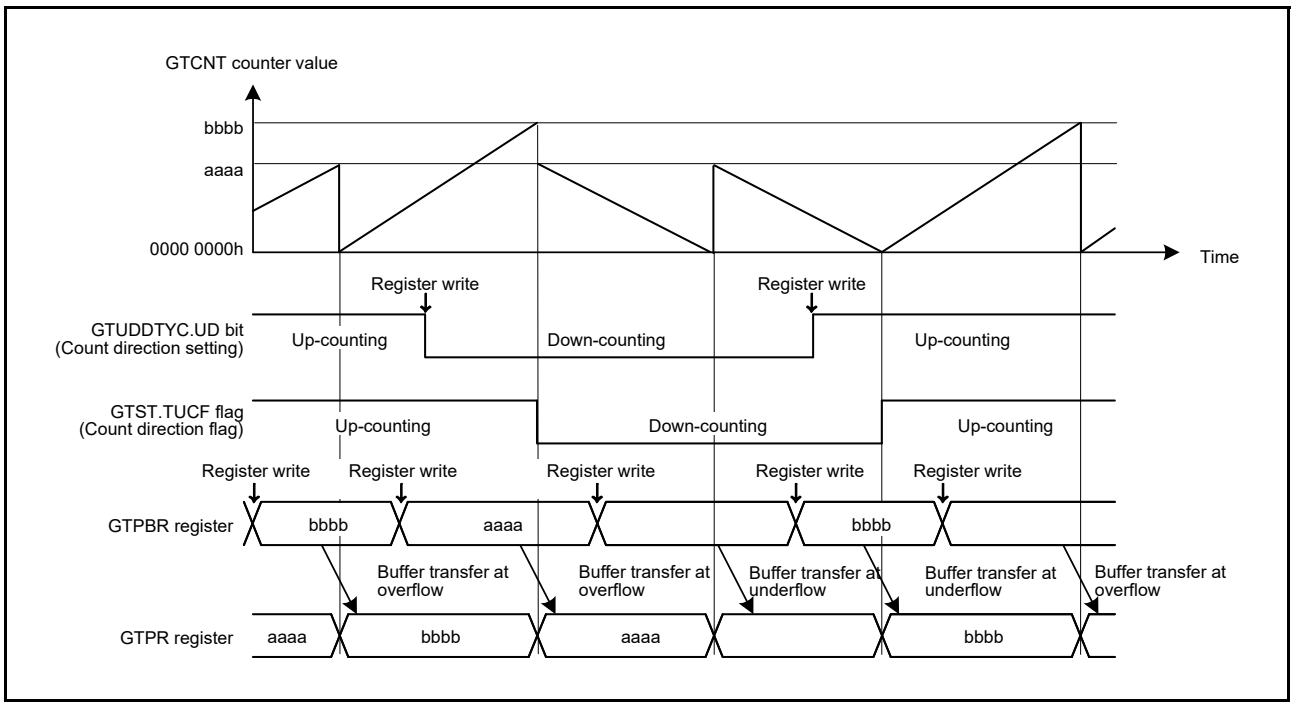


Figure 23.45 Example of a count direction changing function operation (during buffer operation)

23.3.6 Function of Output Duty 0% and 100%

The output duty of the GTIOCA pin and the GTIOCB pin are set to 0% or 100% by changing the GTUDDTYC.OADTY bit or GTUDDTYC.OBDTY bit.

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 stops 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.OADTYF or the GTUDDTYC.OBDTYF bit is set to 1 while the count operation stops, 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, the output duty setting is reflected 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 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%/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 GTIOCA pin at cycle end is decided by GTIOR.GTIOA[3:2] and GTUDDTYC.OADTYR. The output value of GTIOCB 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 23.6 shows the values of GTIOCA/GTIOCB pin output at cycle end.

Table 23.6 Output values after releasing 0%/100% duty setting (m = A, B)

GTIOR.GTIOm[3:2]	Compare match value at cycle end masked by 0%/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 23.46 shows an example of output duty 0% and 100% function operation.

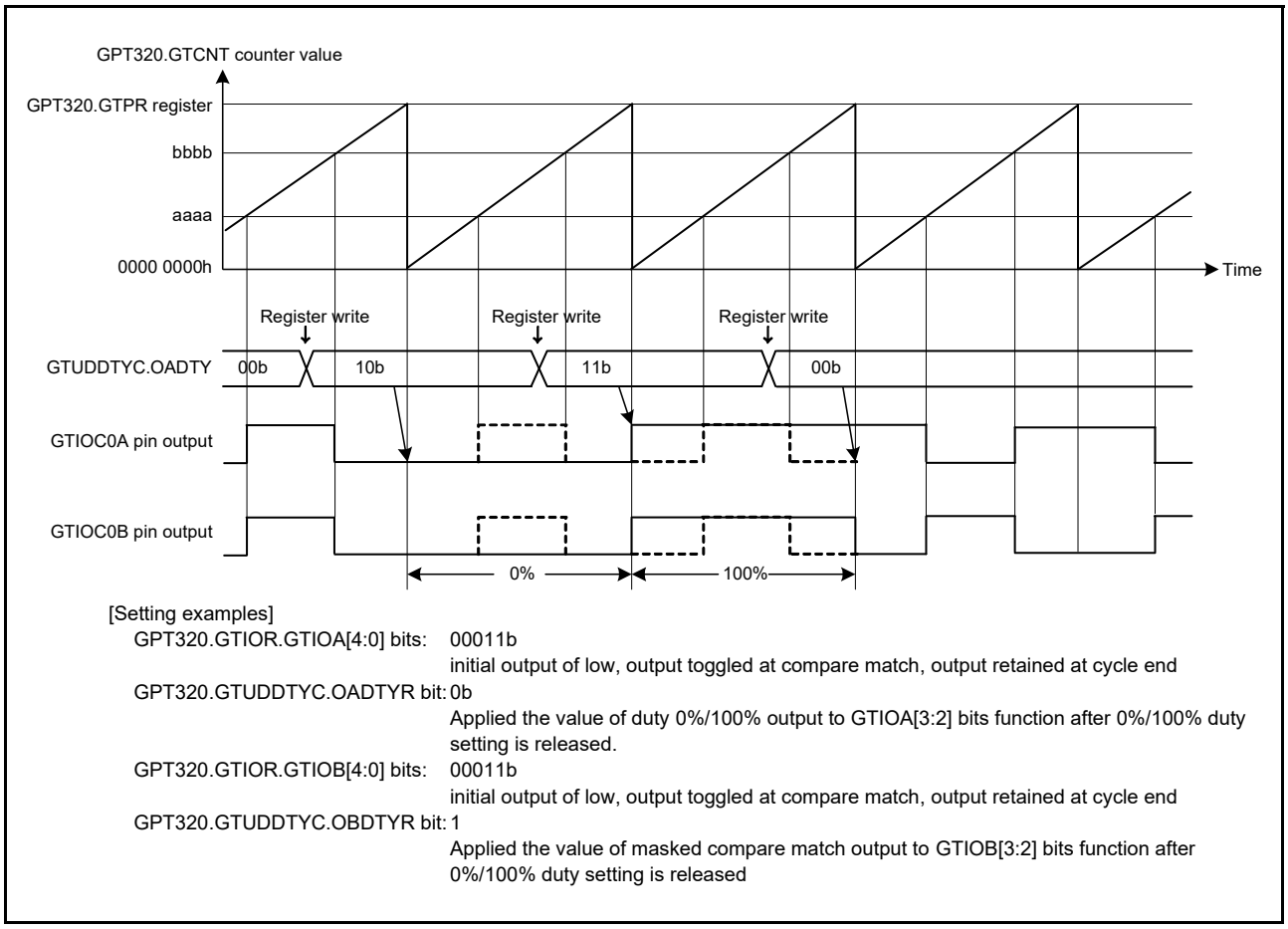


Figure 23.46 Example of output duty 0% and 100% function

23.3.7 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
- GTIOCA/GTIOCB pin input.

23.3.7.1 Hardware Start Operation

The GTCNT counter can be started by selecting a hardware source using GTSSR. Figure 23.47 shows an example of a count start operation by a hardware source. Figure 23.48 shows the setting example.

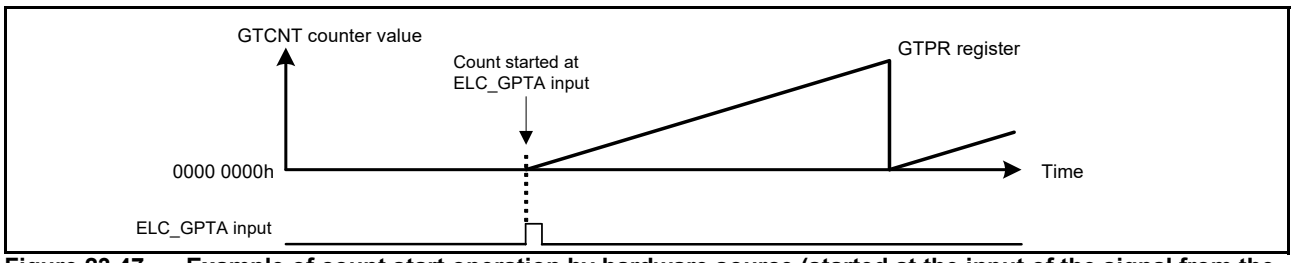


Figure 23.47 Example of count start operation by hardware source (started at the input of the signal from the ELC_GPTA)

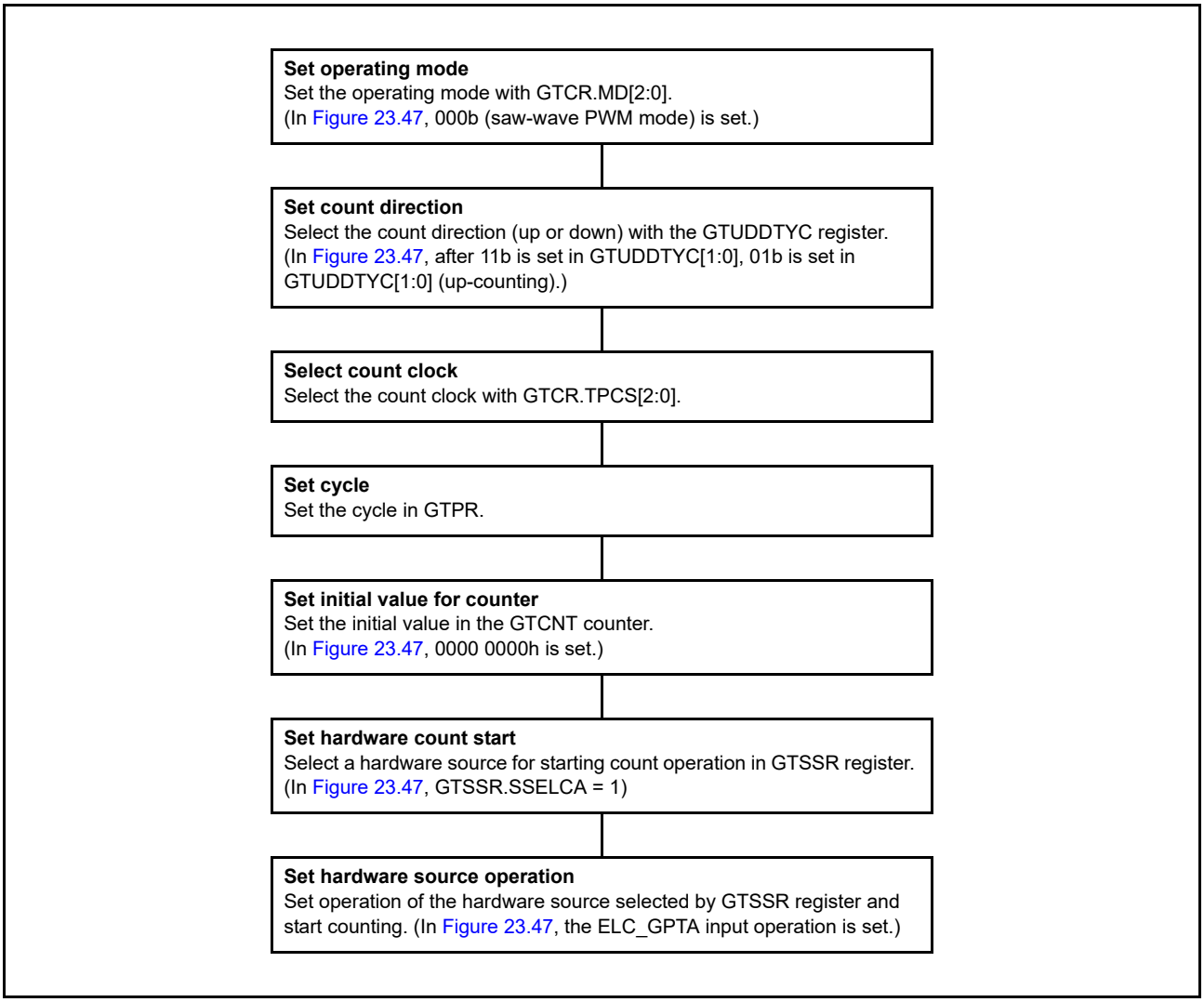


Figure 23.48 Example setting count start operation by hardware source

23.3.7.2 Hardware Stop Operation

The GTCNT counter can be stopped by selecting a hardware source using GTPSR. Figure 23.49 shows an example of a count stop operation by a hardware source. Figure 23.50 shows the setting example. In this example, the count operation stops at the edge of the ELC_GPTA input and restarts at the edges of the ELC_GPTB input.

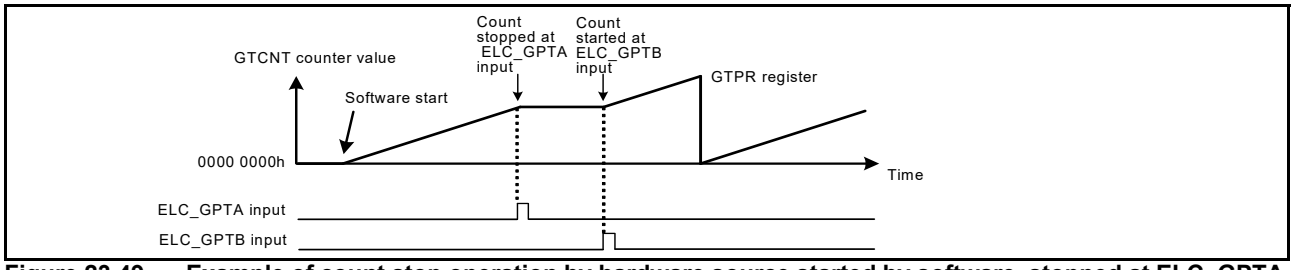


Figure 23.49 Example of count stop operation by hardware source started by software, stopped at ELC_GPTA input, restarted at ELC_GPTB input

Set operating mode
Set the operating mode with GTCR.MD[2:0].
(In Figure 23.49, 000b (saw-wave PWM mode) is set.)

Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
(In Figure 23.49, after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).)

Select count clock
Select the count clock with GTCR.TPCS[2:0].

Set cycle
Set the cycle in GTPR.

Set initial value for counter
Set the initial value in the GTCNT counter.
(In Figure 23.49, 0000 0000h is set.)

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 23.49, GTSSR.SSELCB = 1.)

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 23.49, GTPSR.PSELCA = 1.)

Set hardware source operation
Set operation of the hardware source selected in GTSSR register or GTPSR register, and start or stop counting. (In Figure 23.49, ELC_GPTA input operation and ELC_GPTB input operation are set.)

Figure 23.50 Example for setting count stop operation by hardware source

Figure 23.51 shows an example of a count start/stop operation by a hardware source. Figure 23.52 shows the setting example. In this example, the counter operates during the high-level periods of the external trigger input GTETRGA.

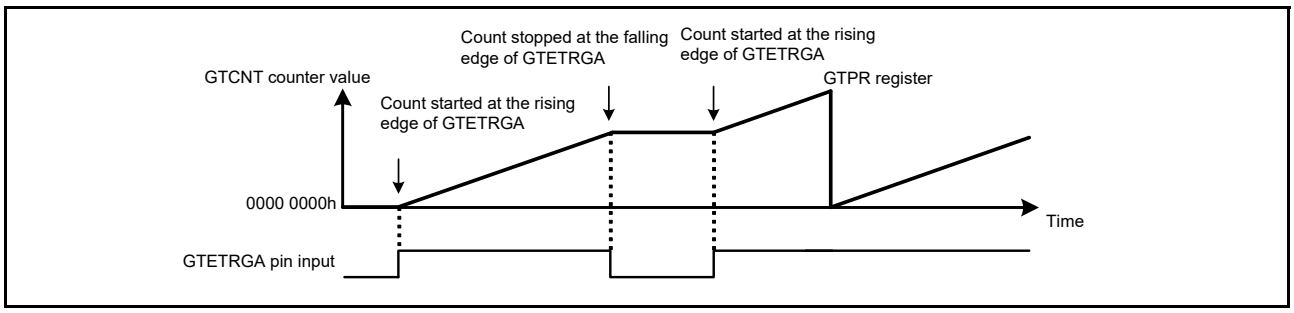


Figure 23.51 Example of count start/stop operation by hardware source (started at rising edge of GTETRGA pin input, stopped at falling edge of GTETRGA pin input)

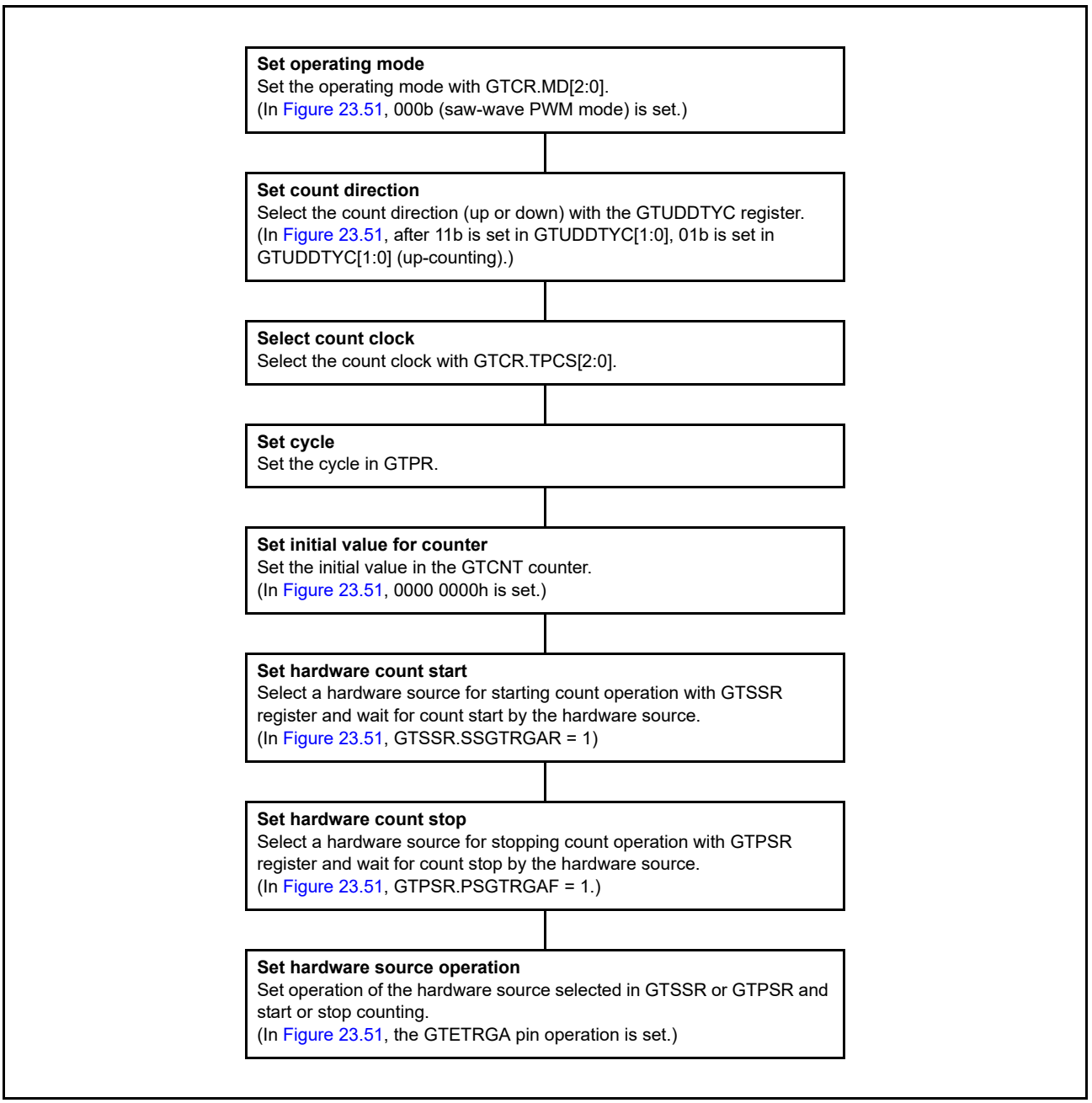


Figure 23.52 Example for setting count start/stop operation by hardware source

23.3.7.3 Hardware Clear Operation

The GTCNT counter can be cleared by selecting a hardware source using GTCSR.

Note that the GPTn_OVF/GPTn_UDF (n = 0 to 9) interrupt (overflow/underflow interrupt) is not generated when the GTCNT counter is cleared by a hardware source or by software.

Figure 23.53 and Figure 23.54 show examples of the GTCNT counter clearing operation by a hardware source. Figure 23.55 shows the setting example. In this example, the GTCNT counter starts at the edge of the ELC_GPTA input, and the counter stops and clears at the edge of the ELC_GPTB input.

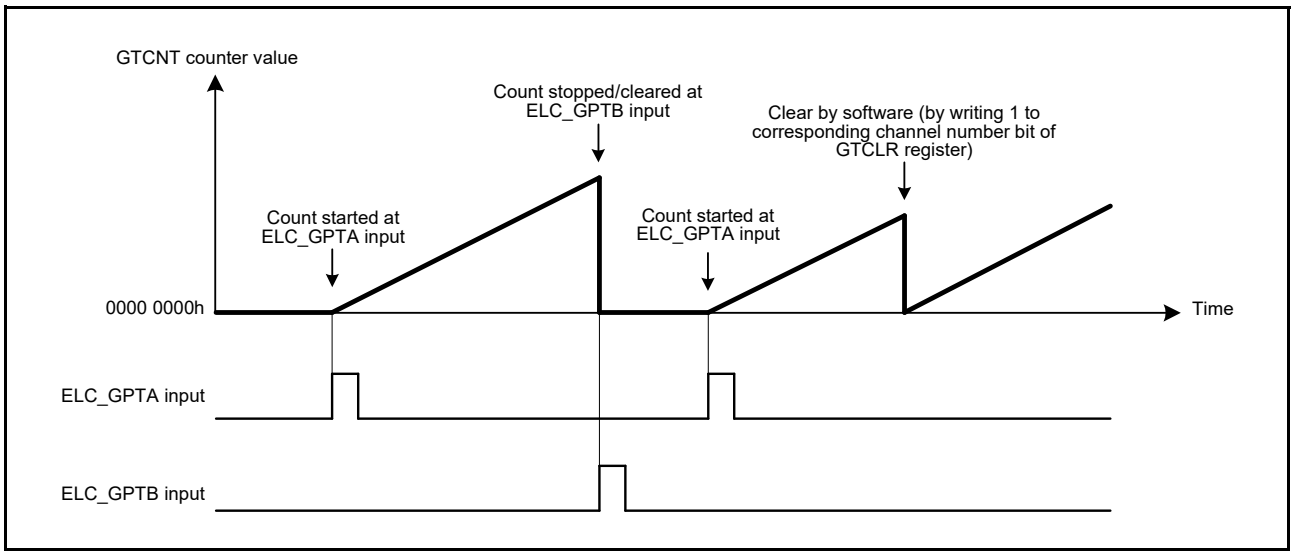


Figure 23.53 Examples of count clearing operation by hardware source with saw wave up-counting, started at ELC_GPTA input, stopped/cleared at ELC_GPTB input

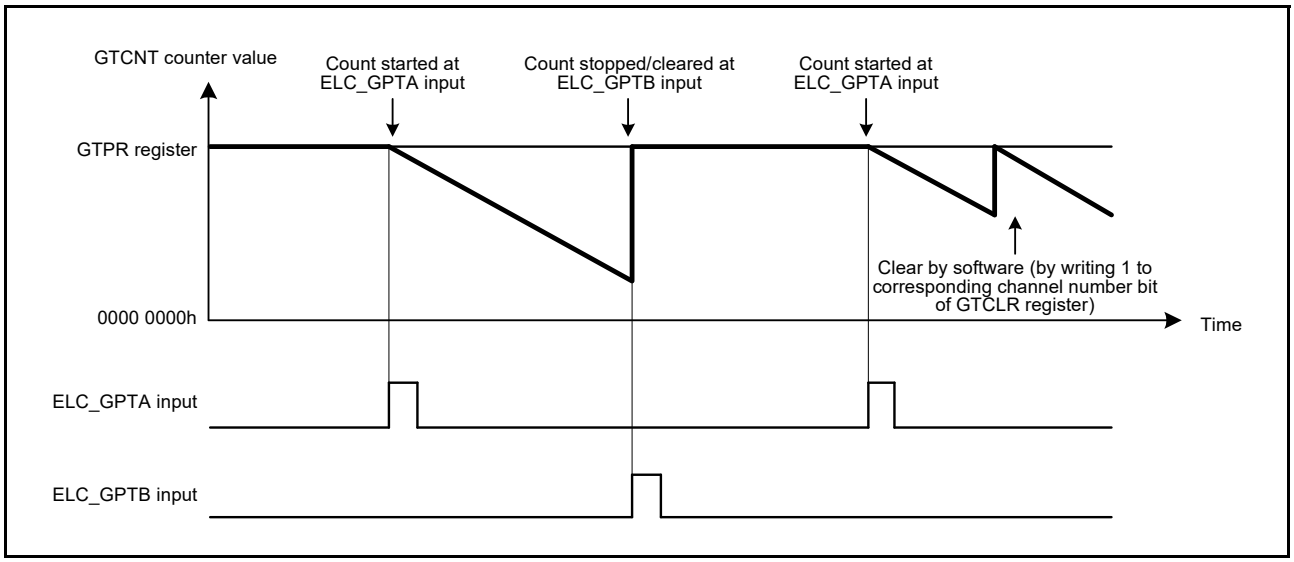


Figure 23.54 Examples of count clearing operation by hardware source (saw wave down-counting, started at ELC_GPTA input, stopped/cleared at ELC_GPTB input)

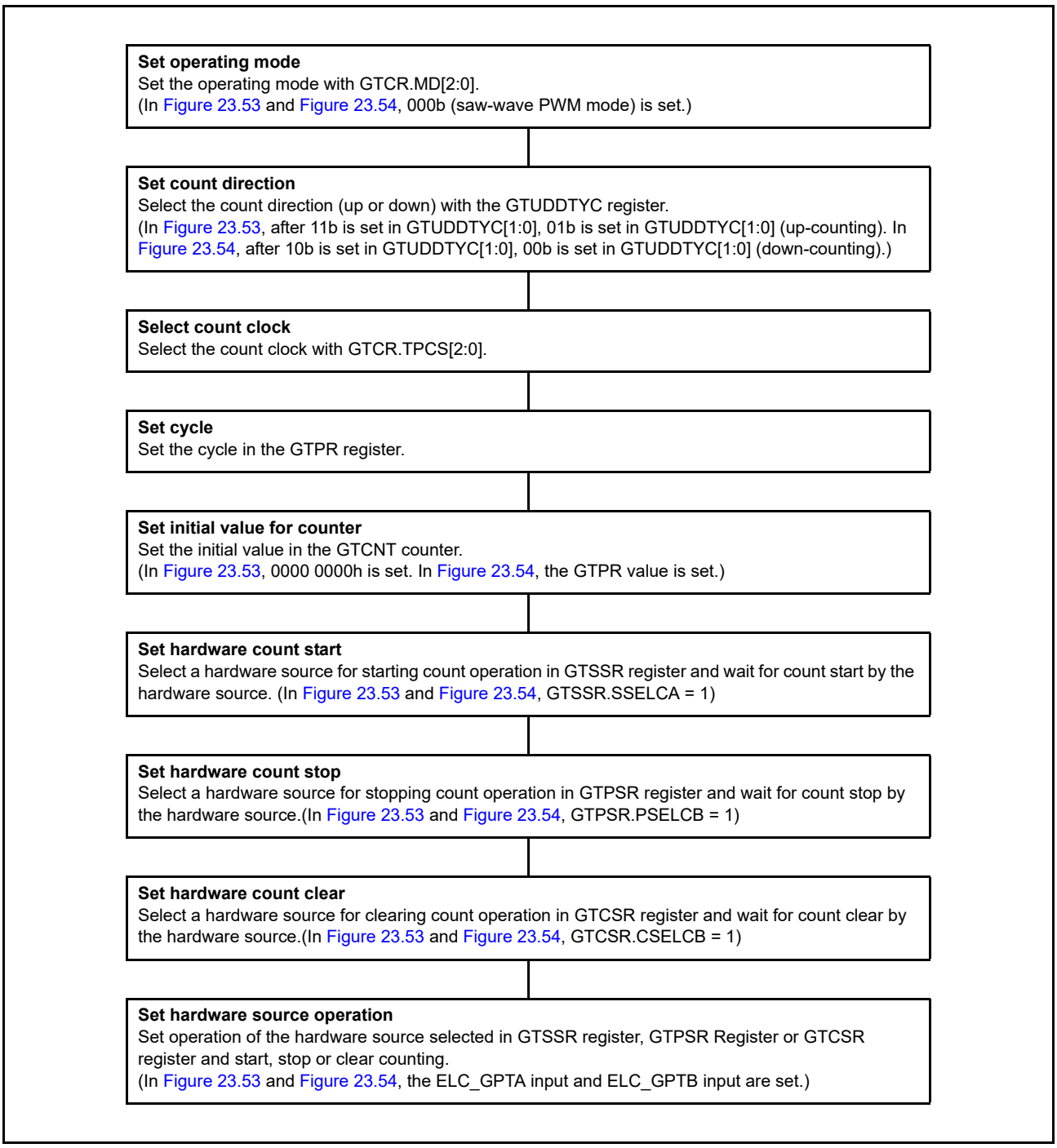


Figure 23.55 Example for setting count clearing operation by a hardware source

The GPTn_OVF/GPTn_UDF (n = 0 to 9) interrupt (overflow/underflow interrupt) is not generated when the counter is cleared by a hardware source or by software.

Figure 23.56 shows the relationship between the counter clearing by a hardware source and the GPTn_OVF (n = 0 to 9) interrupt.

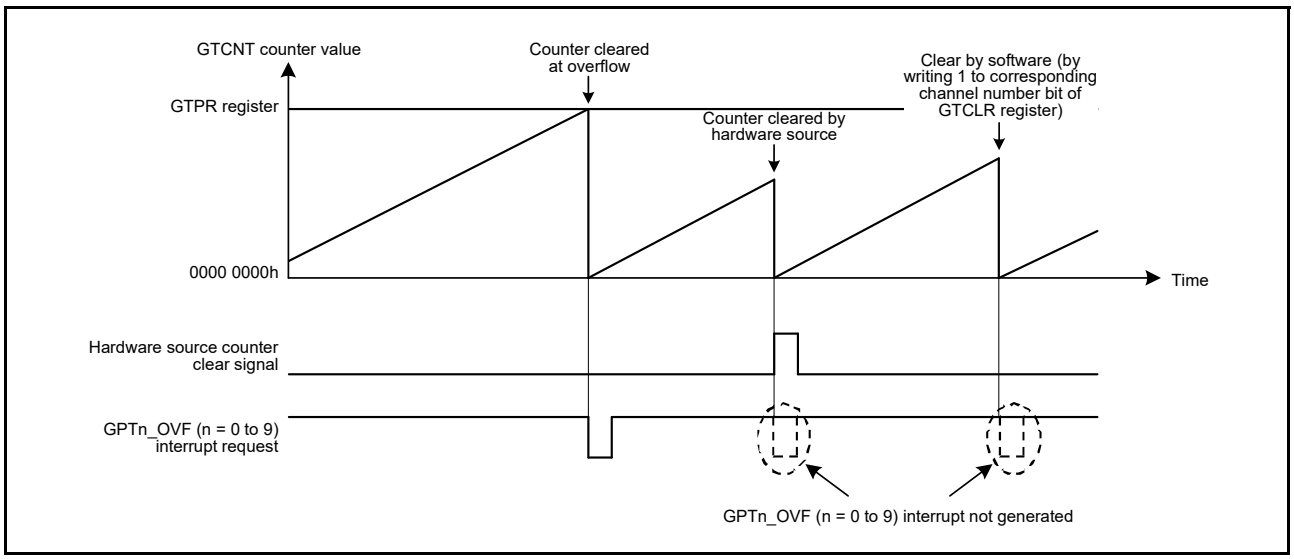


Figure 23.56 Relationship between counter clearing by hardware source and GPTn_OVF (n = 0 to 9) interrupt

23.3.8 Synchronized Operation

Synchronized operation on channels such as a synchronized start, stop and clear operation can be performed.

23.3.8.1 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.

Figure 23.57 shows an example of a simultaneous start, stop and clear by software. Figure 23.58 shows an example a of phase start operation by software.

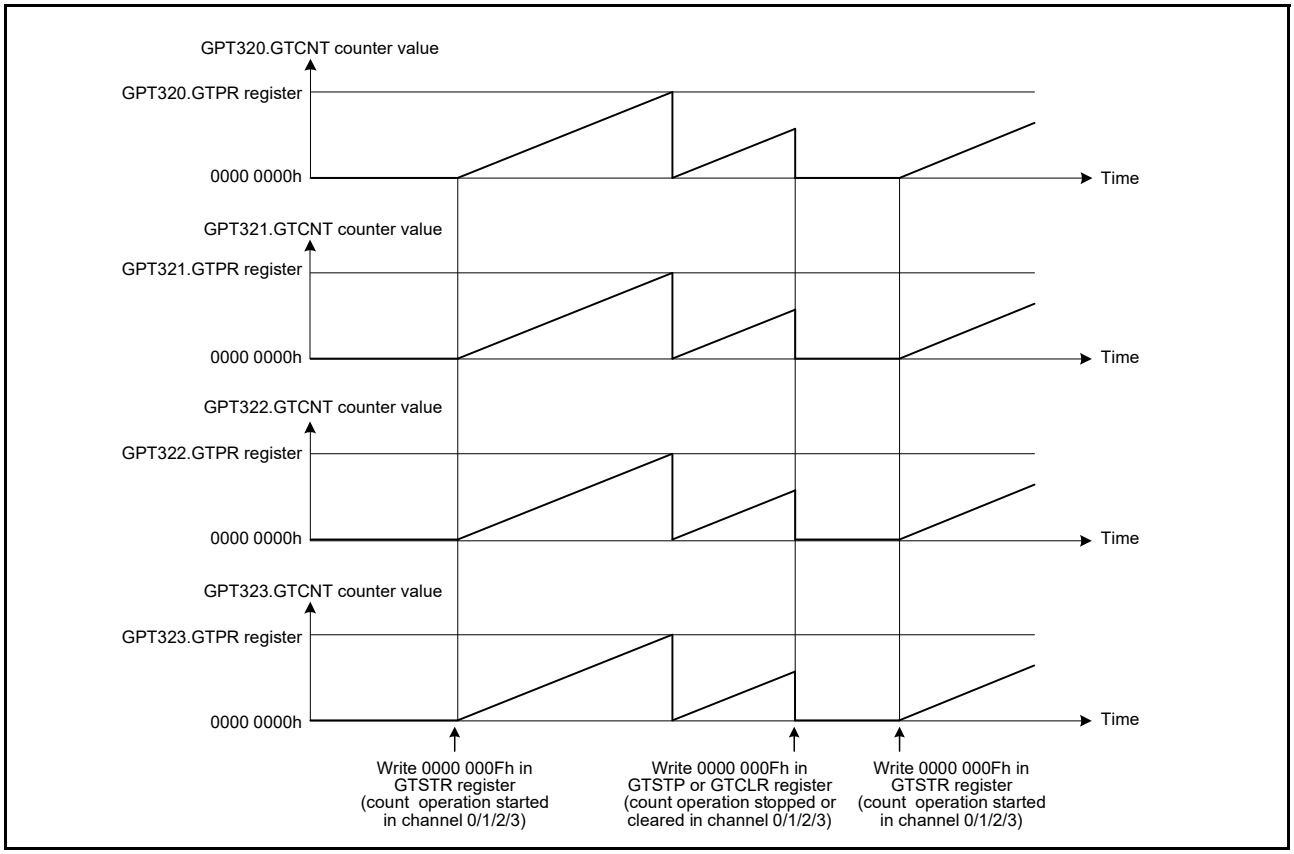


Figure 23.57 Example of a simultaneous start, stop, and clear by software with the same count cycle (GTPR register value)

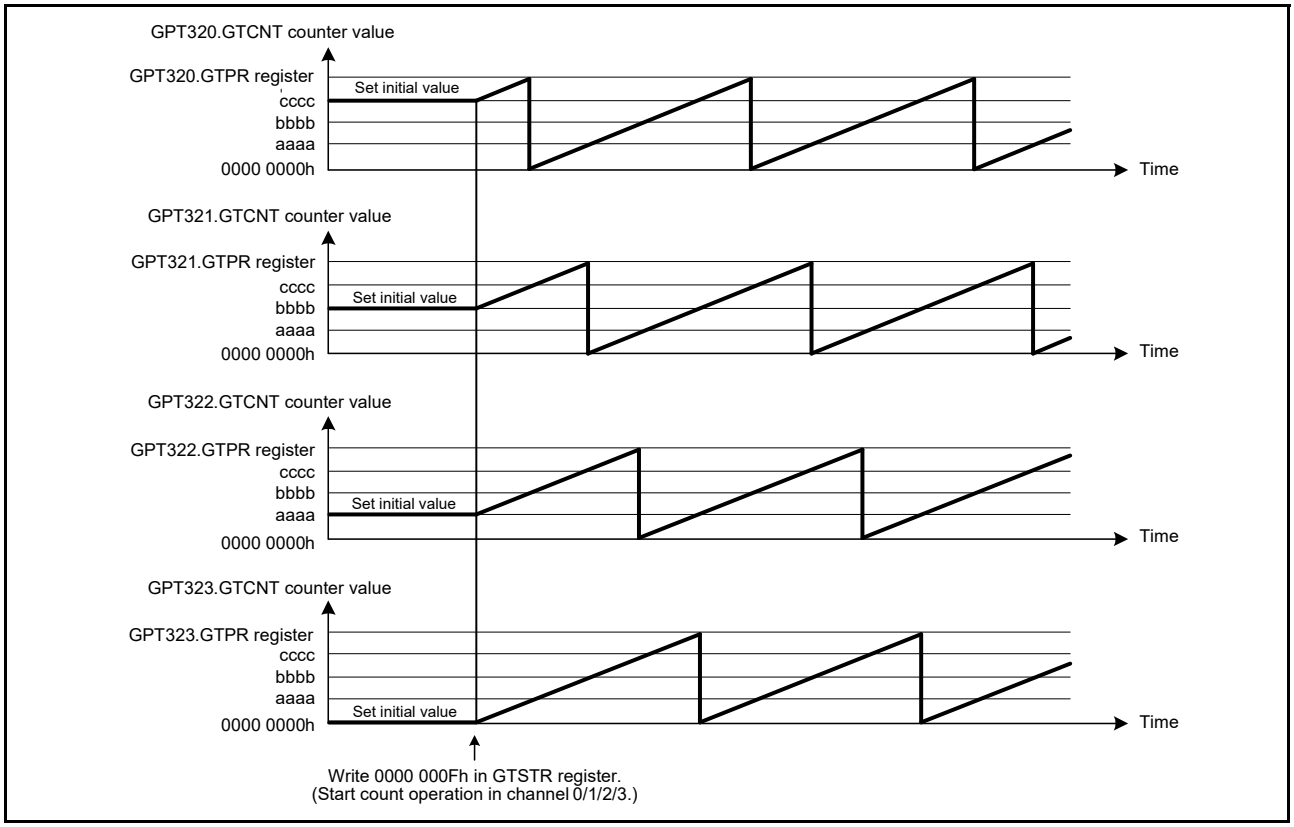


Figure 23.58 Example of software phase start with the same count cycle (GTPR register value)

23.3.8.2 Synchronized Operation by Hardware

The GTCNT counters can be started simultaneously by the following hardware sources:

- External trigger input
- ELC event input.

Figure 23.59 shows an example of a simultaneous start, stop and clear operation by a hardware source. Figure 23.60 shows the setting example.

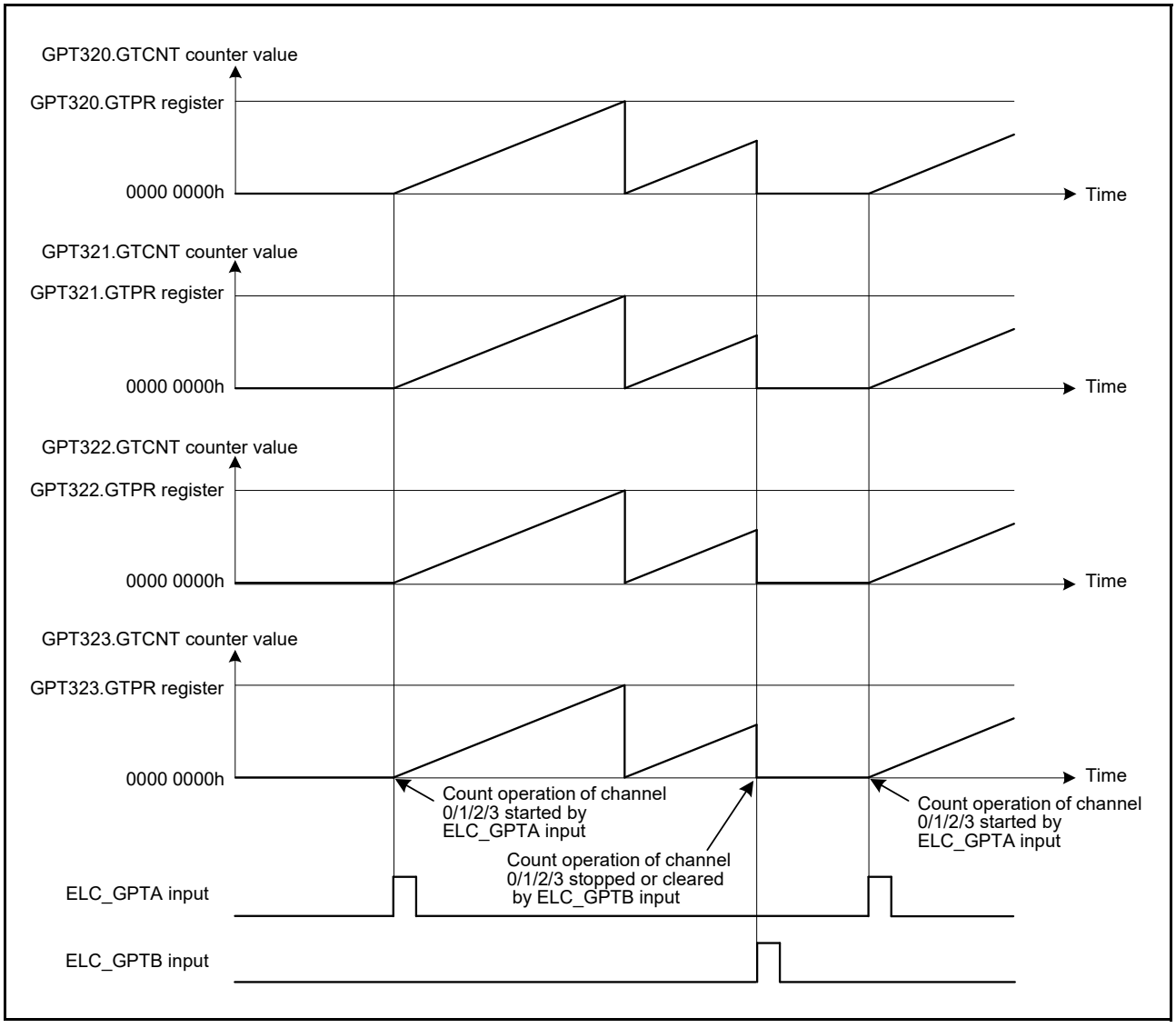


Figure 23.59 Example of a simultaneous start, stop and clear by a hardware source with the same count cycle (GTPR register value)

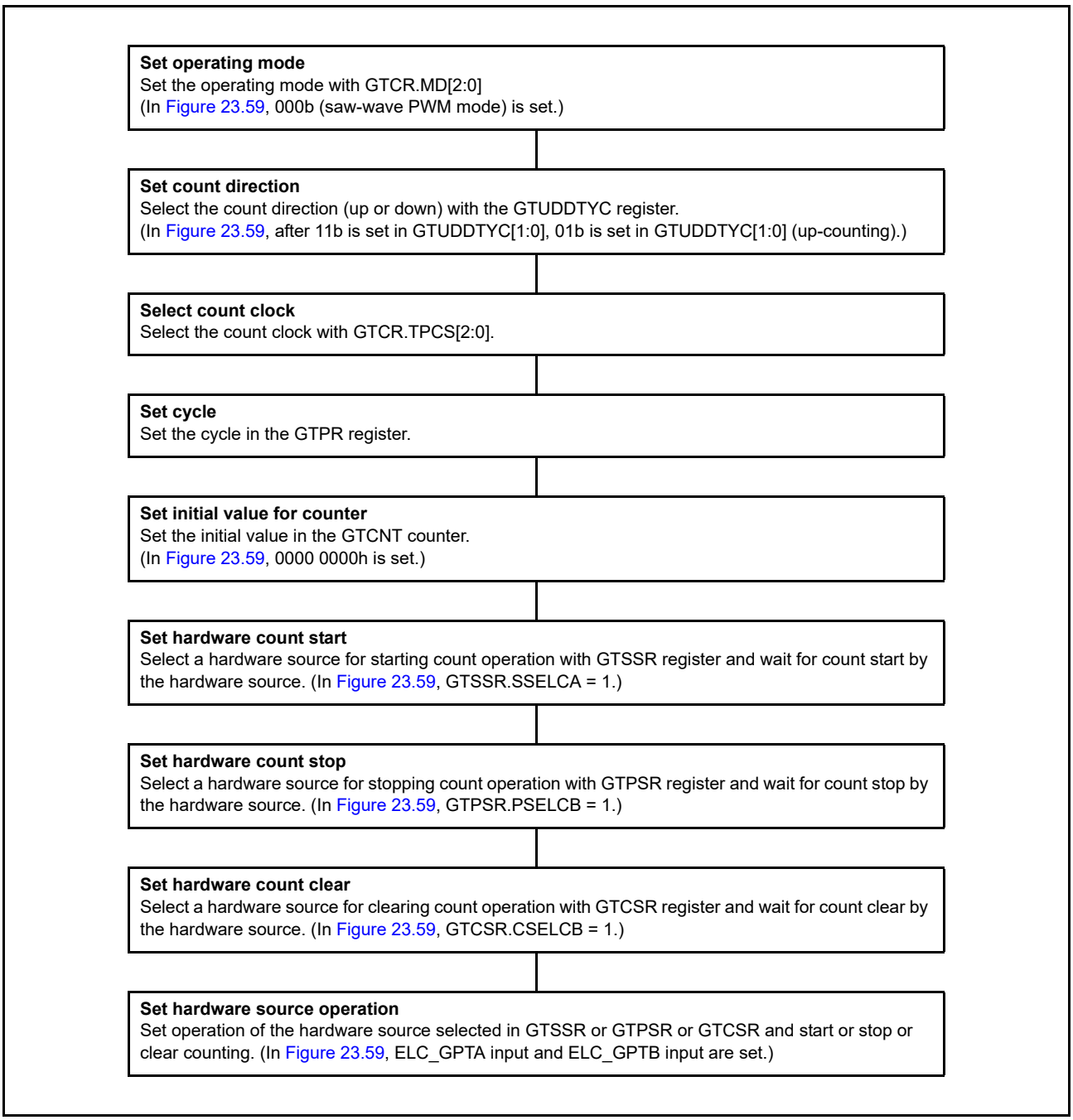


Figure 23.60 Example for setting simultaneous start by hardware source

23.3.9 PWM Output Operation Examples

(1) Synchronized PWM Output

The GPT output 20 phases of linked PWM waveforms for a maximum of 10 channels by multiple GPTs.

Figure 23.61 shows an example in which four channels perform synchronized operation in saw-wave PWM mode and eight phases of PWM waveforms are output. The GTIOCA is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCB is set so that it outputs low as the initial value, high at a GTCCRB compare match, and low at the cycle end.

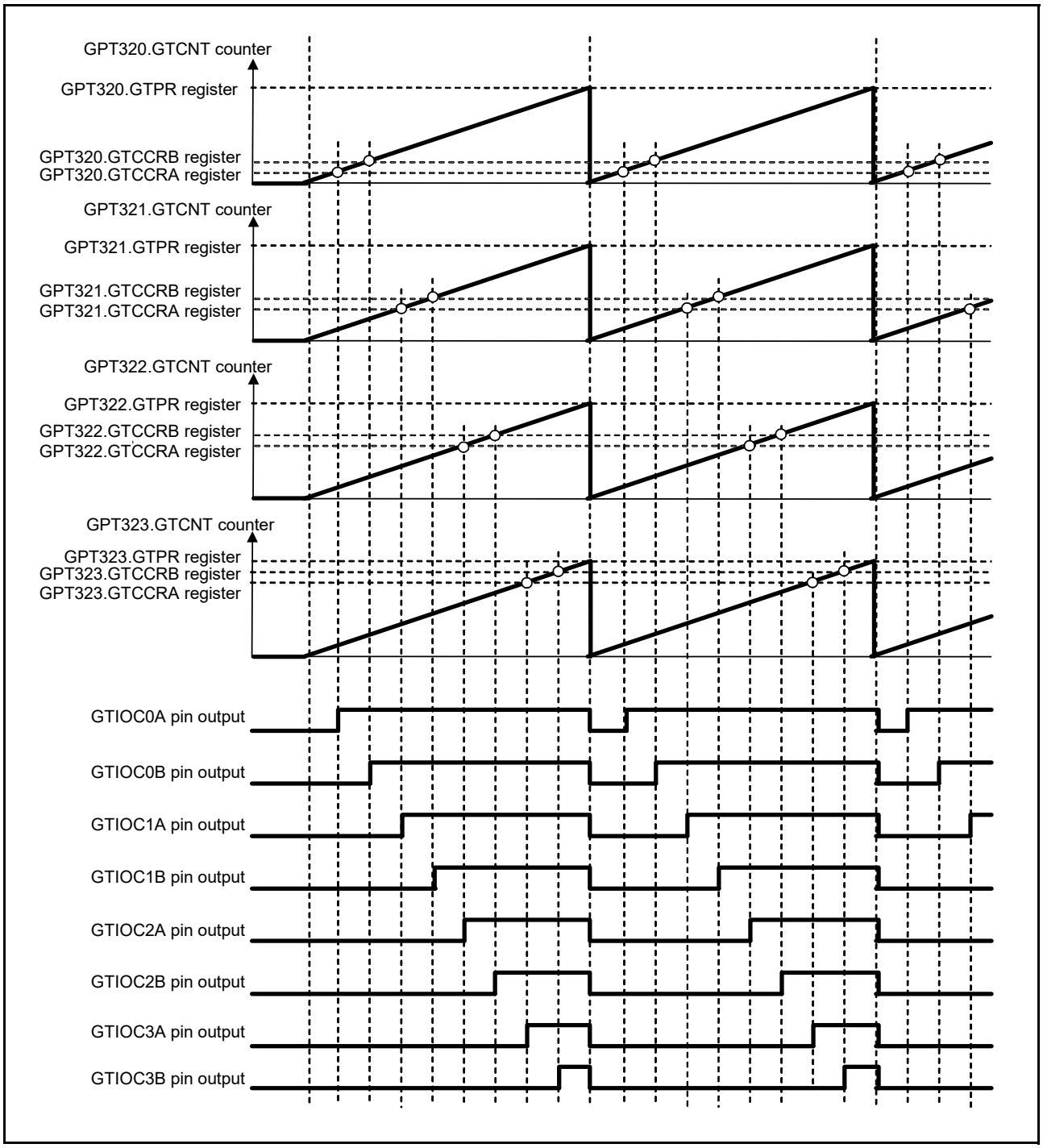


Figure 23.61 Example of synchronized PWM output

(2) Three-Phase Saw-Wave Complementary PWM Output

Figure 23.62 shows an example in which three channels perform synchronized operation in saw-wave PWM mode and 3-phase complementary PWM waveforms are output. The GTIOCA 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 GTIOCB pin is set so that it outputs high as the initial value, low at a GTCCRB compare match, and high at the cycle end.

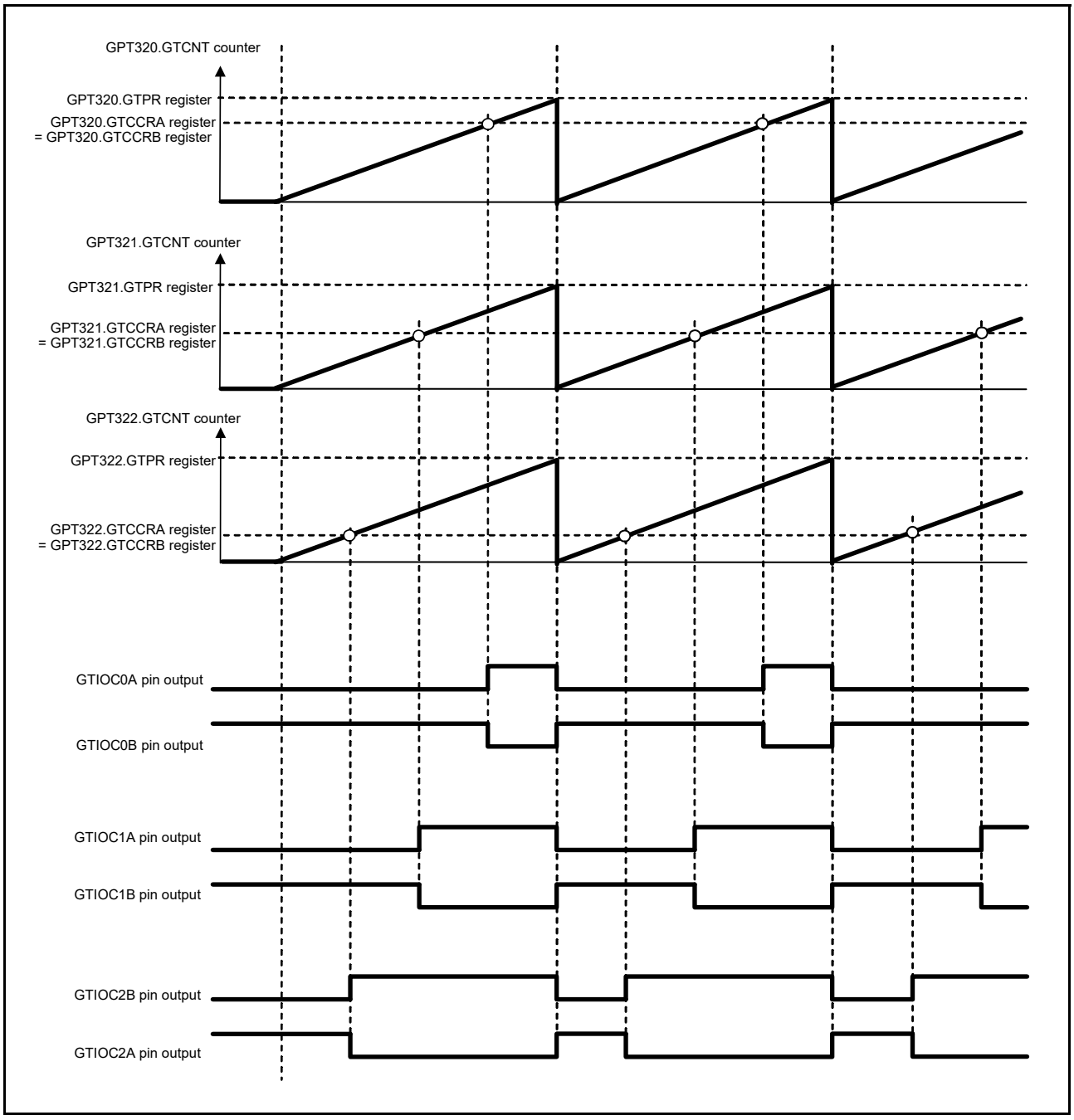


Figure 23.62 Example of 3-phase saw-wave complementary PWM output

(3) 3-Phase Saw-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 23.63 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 GTIOCA pin is set so that it outputs low as the initial value, toggle the output at a GTCCRA compare match, and retain the output at the cycle end. The GTIOCB 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.

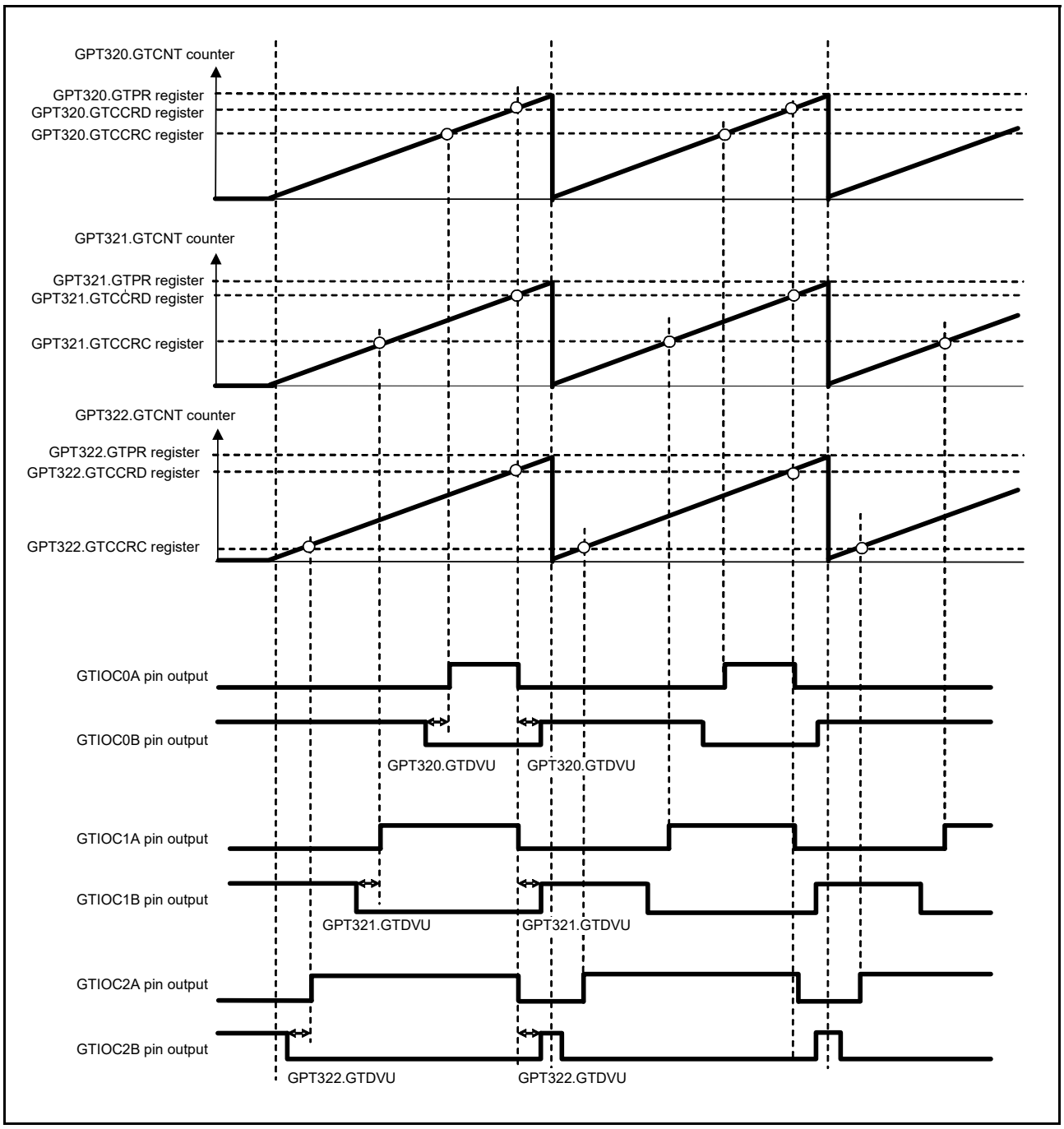


Figure 23.63 Example of 3-phase saw-wave complementary PWM output with automatic dead time setting

(4) 3-Phase Triangle-Wave Complementary PWM Output

Figure 23.64 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 GTIOCA 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 GTIOCB 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.

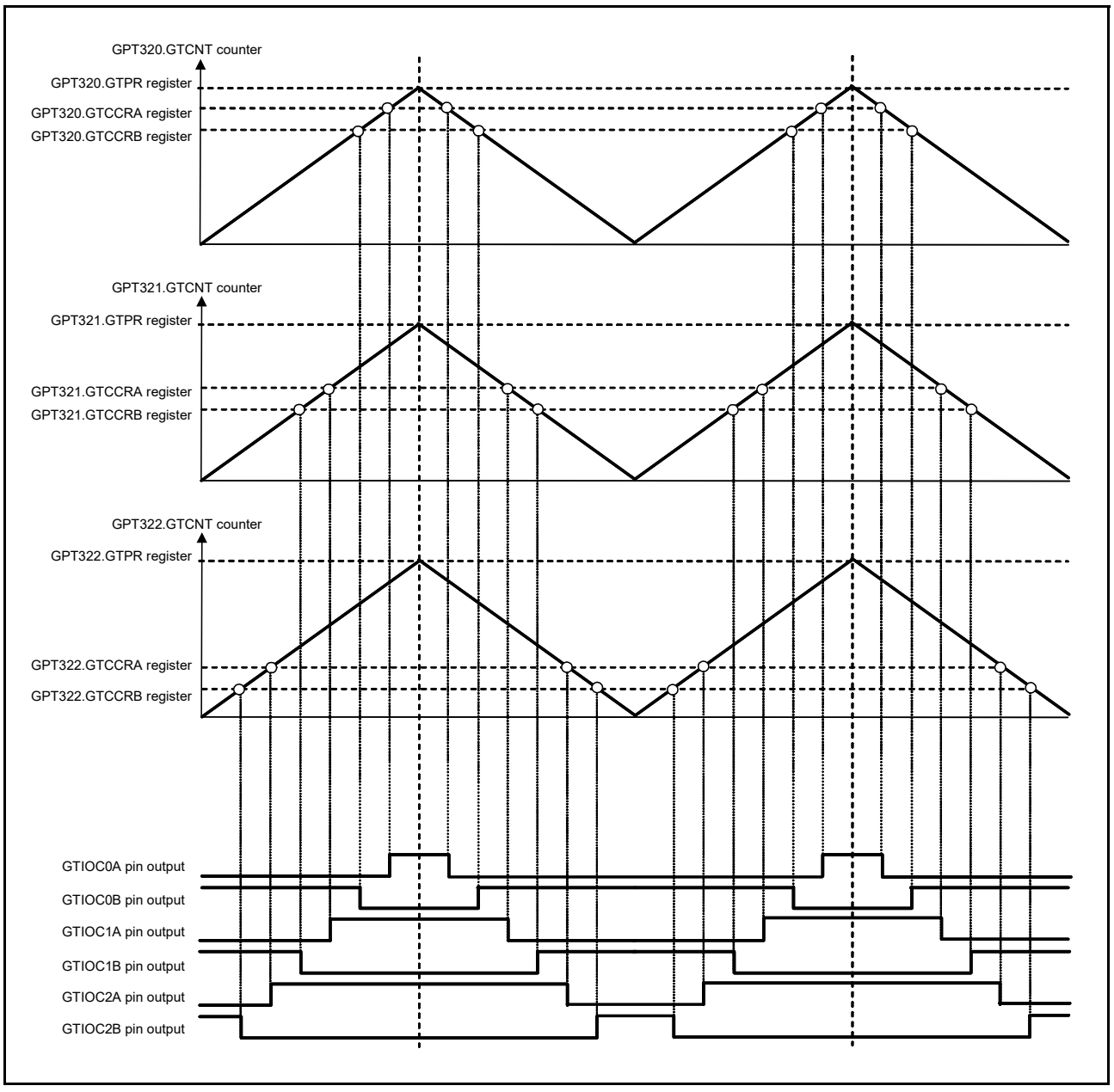


Figure 23.64 Example of 3-phase triangle-wave complementary PWM output

(5) 3-Phase Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 23.65 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 GTIOCA 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 GTIOCB 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.

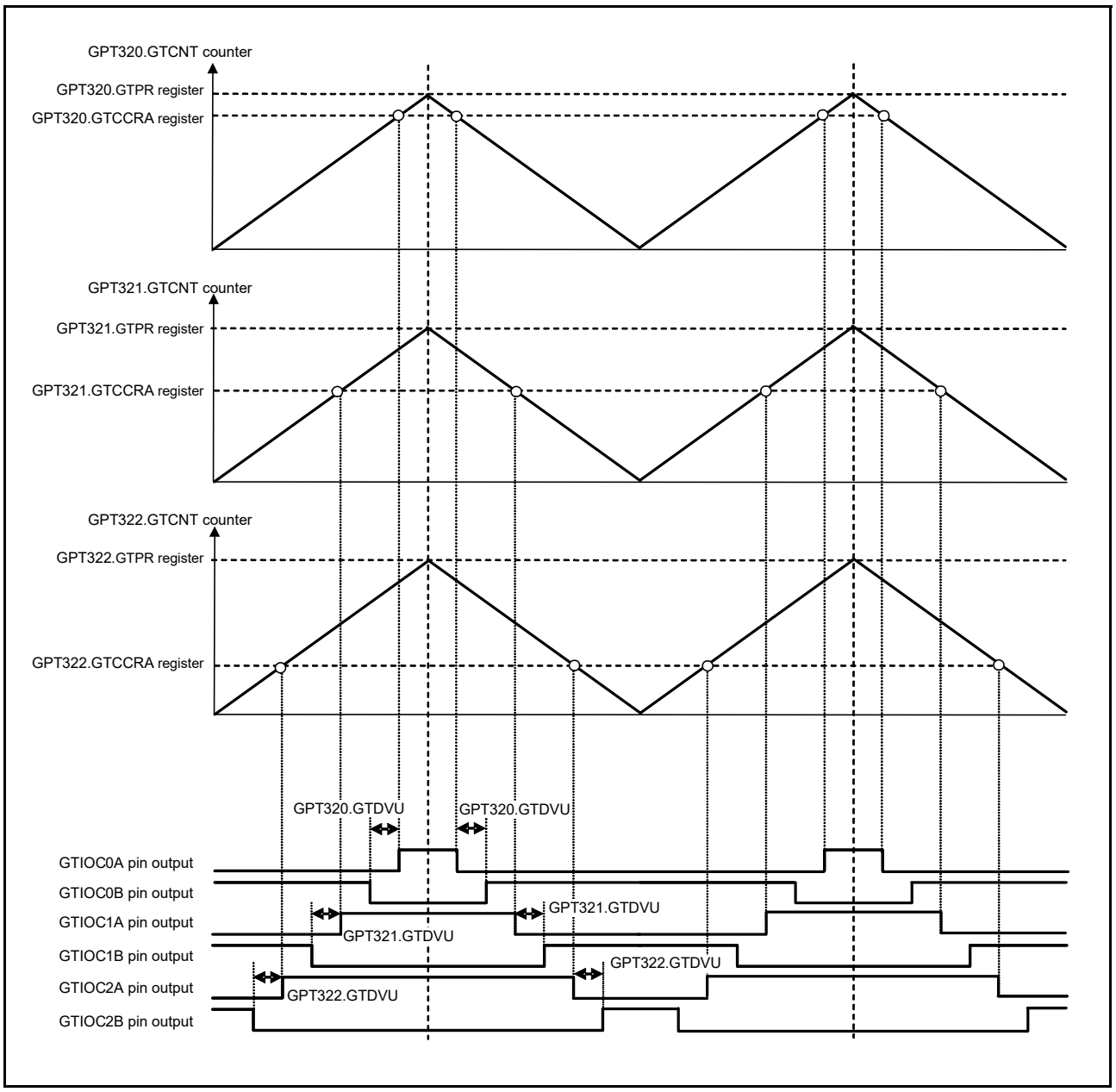


Figure 23.65 Example of 3-phase triangle-wave complementary PWM output with automatic dead time setting

(6) 3-Phase Asymmetric Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 23.66 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 GTIOCA 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 GTIOCB 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.

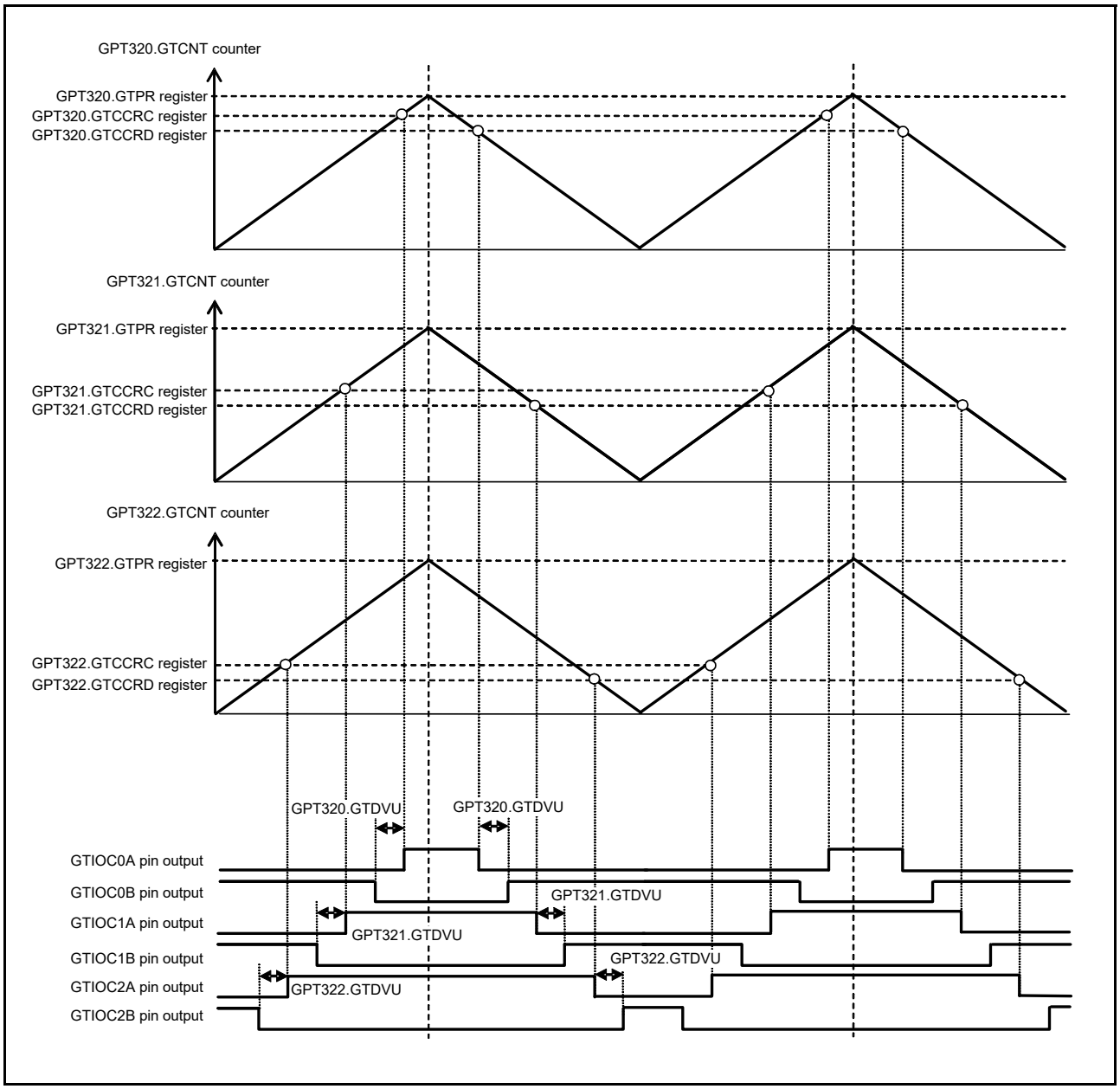


Figure 23.66 Example of 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting

23.3.10 Phase Counting Function

The phase difference between the GTIOCA and GTIOCB pin 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 GTIOCA and GTIOCB pin inputs being set in the GTUPSR and GTDNSR registers. For details on count operation, see section 23.3.1.1, Counter Operation.

Figure 23.67 to Figure 23.76 show phase counting modes 1 to 5. Table 23.7 to Table 23.16 show conditions of up-counting or down-counting and lists settings for the GTUPSR and GTDNSR registers.

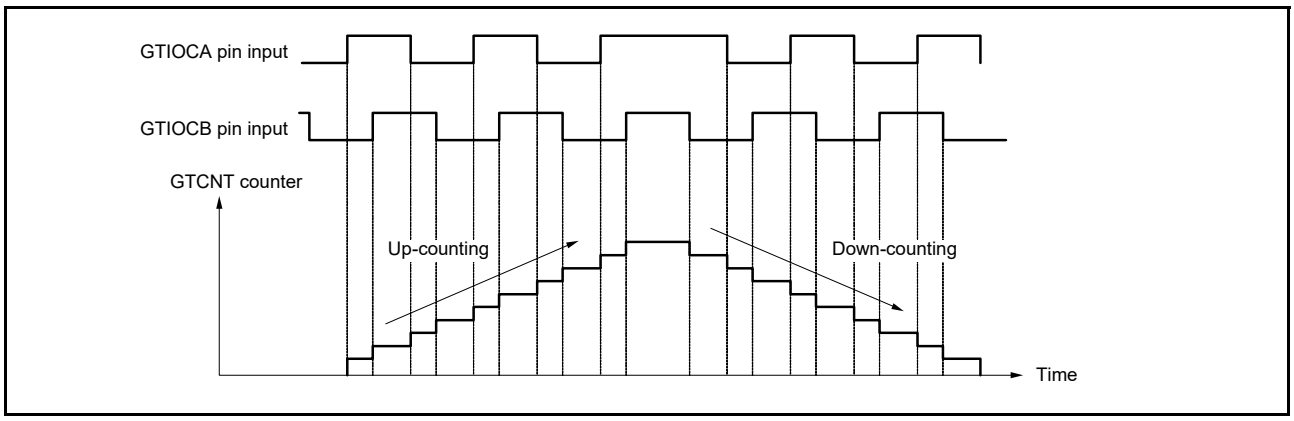


Figure 23.67 Example of phase counting mode 1

Table 23.7 Conditions of up-counting/down-counting in phase counting mode 1

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
High	↑	Up-counting	GTUPSR = 00006900h GTDNSR = 00009600h
Low	↓		
↑	Low		
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High		
↓	Low		

↑ : Rising edge

↓ : Falling edge

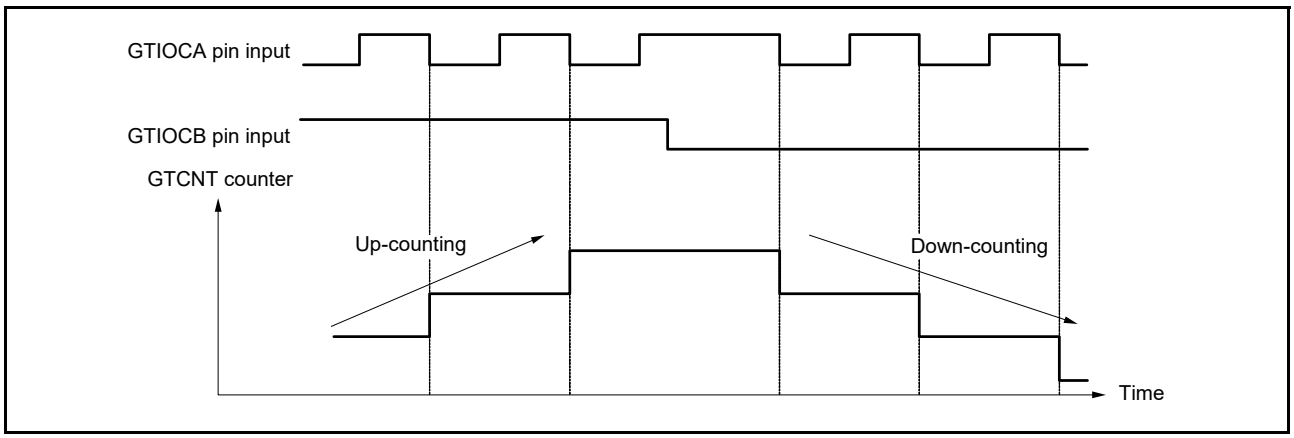


Figure 23.68 Example of phase counting mode 2 (A)

Table 23.8 Conditions of up-counting/down-counting in phase counting mode 2 (A)

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
High	↑	Don't care	GTUPSR = 00000800h GTDNSR = 00000400h
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High		
↓	Low	Down-counting	

↑ : Rising edge

↓ : Falling edge

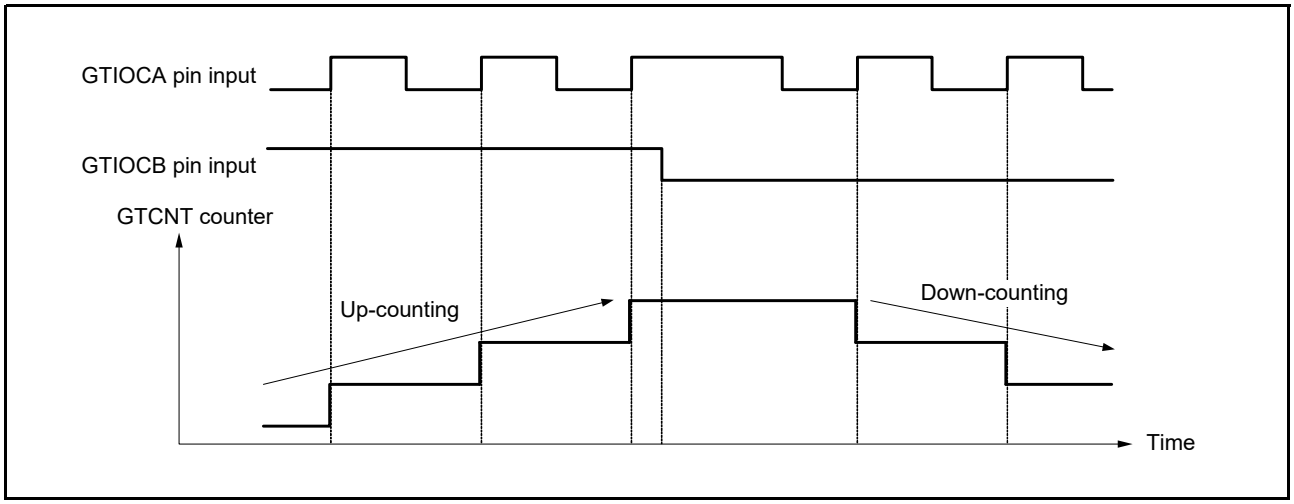


Figure 23.69 Example of phase counting mode 2 (B)

Table 23.9 Conditions of up-counting/down-counting in phase counting mode 2 (B)

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
High	↑	Don't care	GTUPSR = 00000200h GTDNSR = 00000100h
Low	↓		
↑	Low	Down-counting	
↓	High	Don't care	
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Don't care	

↑ : Rising edge

↓ : Falling edge

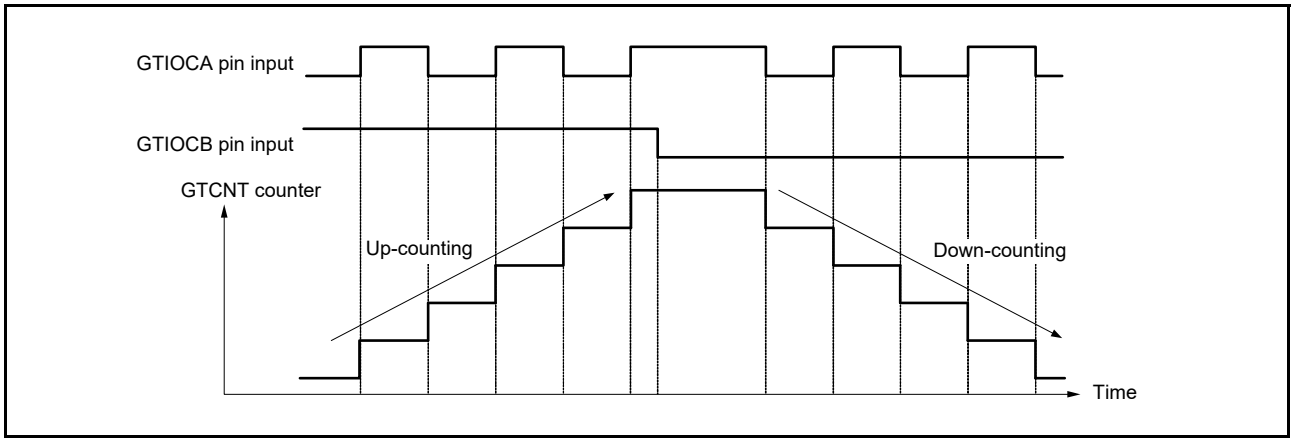


Figure 23.70 Example of phase counting mode 2 (C)

Table 23.10 Conditions of up-counting/down-counting in phase counting mode 2 (C)

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
High	↑	Don't care	GTUPSR = 00000A00h GTDNSR = 00000500h
Low	↓		
↑	Low	Down-counting	
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High	Up-counting	
↓	Low	Down-counting	

↑ : Rising edge

↓ : Falling edge

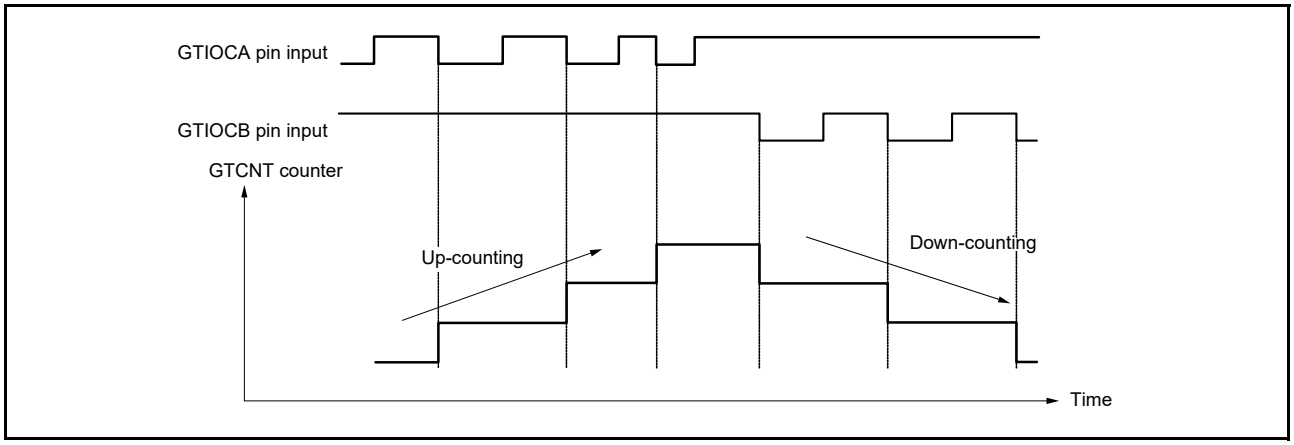


Figure 23.71 Example of phase counting mode 3 (A)

Table 23.11 Conditions of up-counting/down-counting in phase counting mode 3 (A)

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
High	↑	Don't care	GTUPSR = 00000800h GTDNSR = 00008000h
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Don't care	
↑	High		
↓	Low		

↑ : Rising edge

↓ : Falling edge

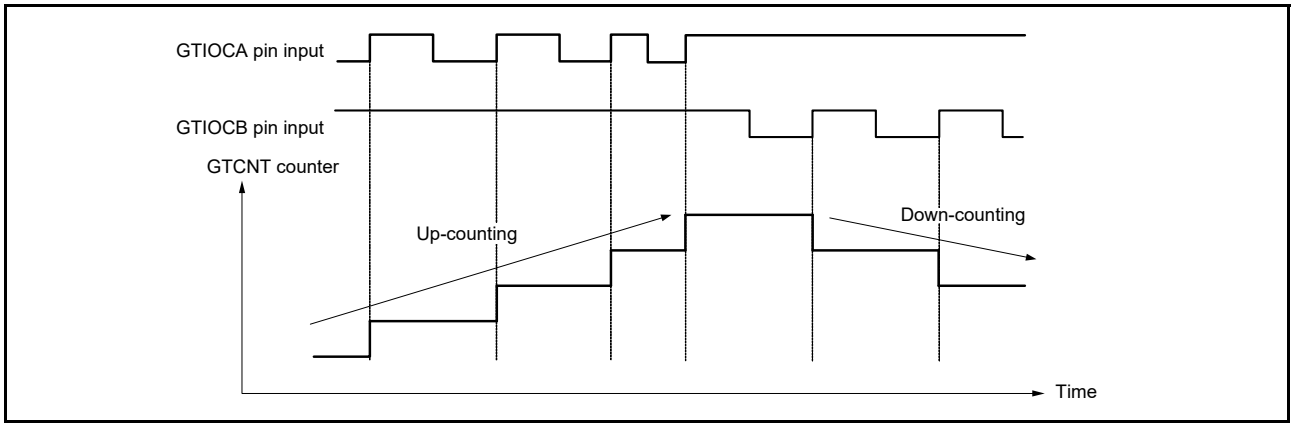


Figure 23.72 Example of phase counting mode 3 (B)

Table 23.12 Conditions of up-counting/down-counting in phase counting mode 3 (B)

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
High	↑	Down-counting	GTUPSR = 00000200h GTDNSR = 00002000h
Low	↓	Don't care	
↑	Low		
↓	High		
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Don't care	

↑ : Rising edge

↓ : Falling edge

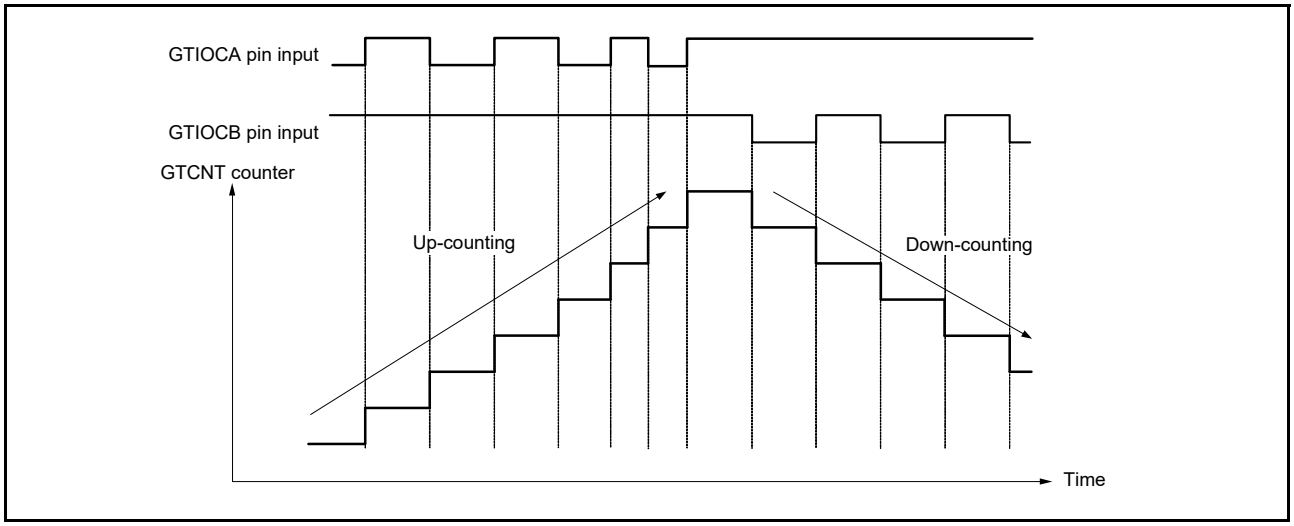


Figure 23.73 Example of phase counting mode 3 (C)

Table 23.13 Conditions of up-counting/down-counting in phase counting mode 3 (C)

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
High	↑	Down-counting	GTUPSR = 00000A00h GTDNSR = 0000A000h
Low	↓	Don't care	
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Don't care	
↑	High	Up-counting	
↓	Low	Don't care	

↑ : Rising edge

↓ : Falling edge

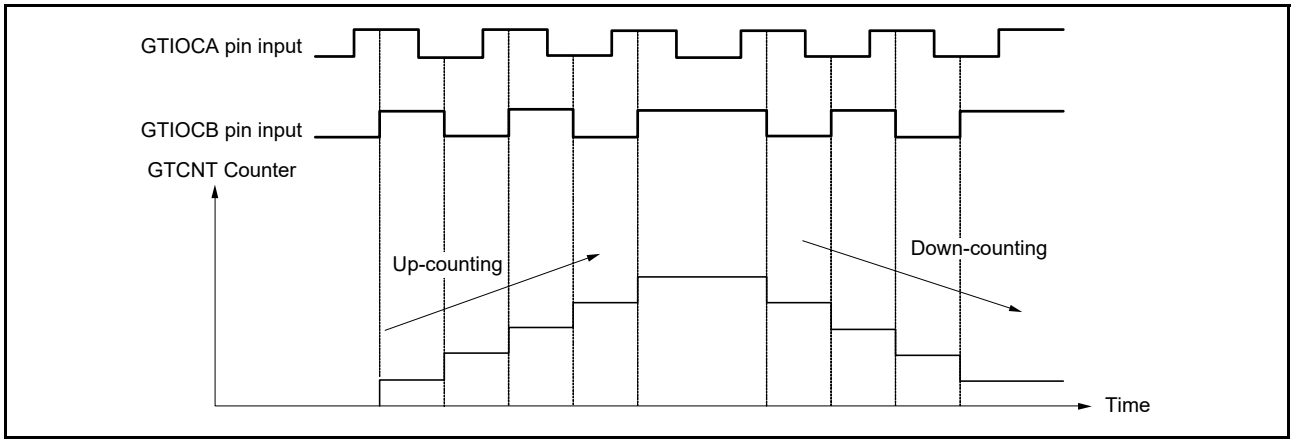


Figure 23.74 Example of phase counting mode 4

Table 23.14 Conditions of up-counting/down-counting in phase counting mode 4

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
High	↑	Up-counting	GTUPSR = 00006000h GTDNSR = 00009000h
Low	↓		
↑	Low	Don't care	
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High	Don't care	
↓	Low		

↑ : Rising edge

↓ : Falling edge

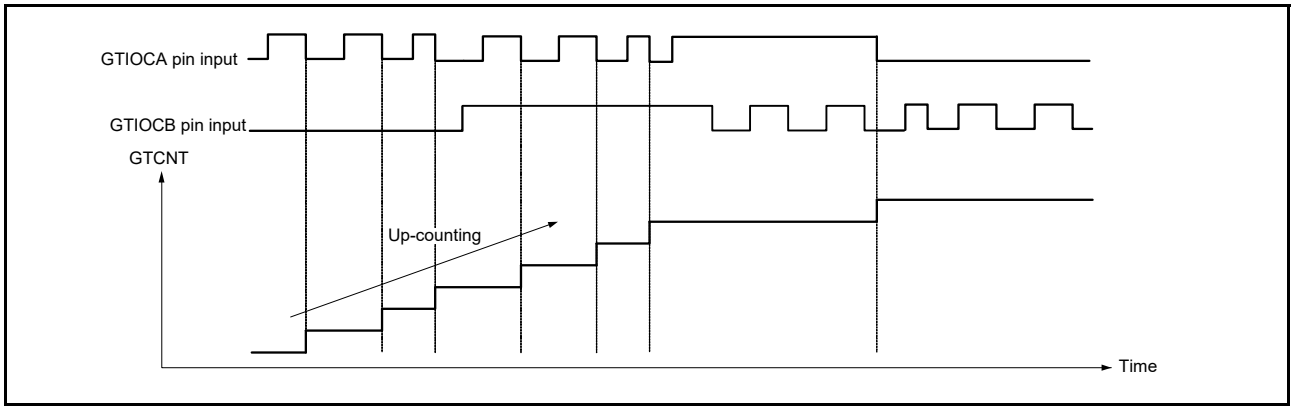


Figure 23.75 Example of phase counting mode 5 (A)

Table 23.15 Conditions of up-counting/down-counting in phase counting mode 5 (A)

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
High	↑	Don't care	GTUPSR = 00000C00h GTDNSR = 00000000h
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High		
↓	Low	Up-counting	

↑ : Rising edge

↓ : Falling edge

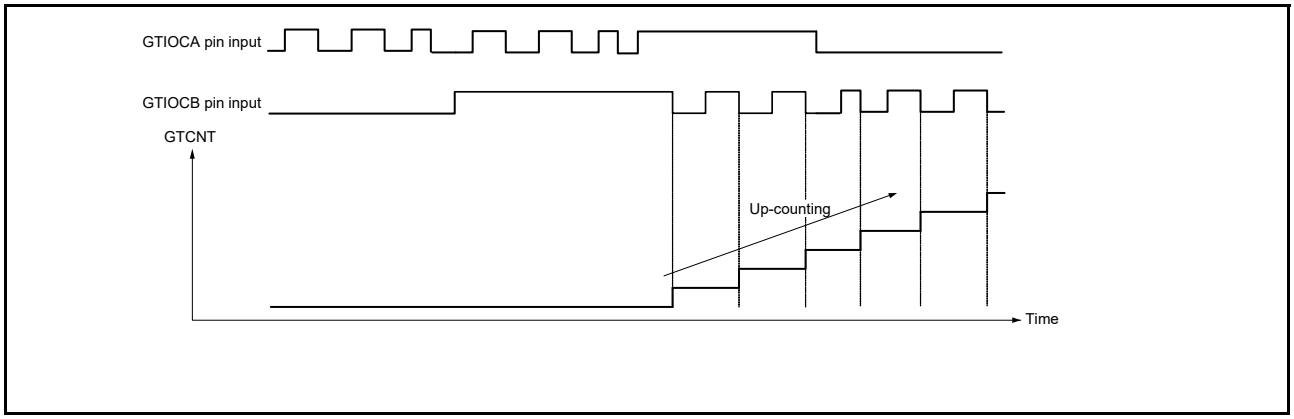


Figure 23.76 Example of phase counting mode 5 (B)

Table 23.16 Conditions of up-counting/down-counting in phase counting mode 5 (B)

GTIOCA pin input	GTIOCB pin input	Operation	Register setting
High	↑	Don't care	GTUPSR = 0000C000h GTDNSR = 00000000h
Low	↓	Up-counting	
↑	Low	Don't care	
↓	High		
High	↓	Up-counting	
Low	↑	Don't care	
↑	High		
↓	Low		

↑ : Rising edge

↓ : Falling edge

23.3.11 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 or external signals detected by the Hall element, a PWM waveform of GPT320.GTIOCA.

Figure 23.77 shows the GPT_OPS control flow conceptual diagram.

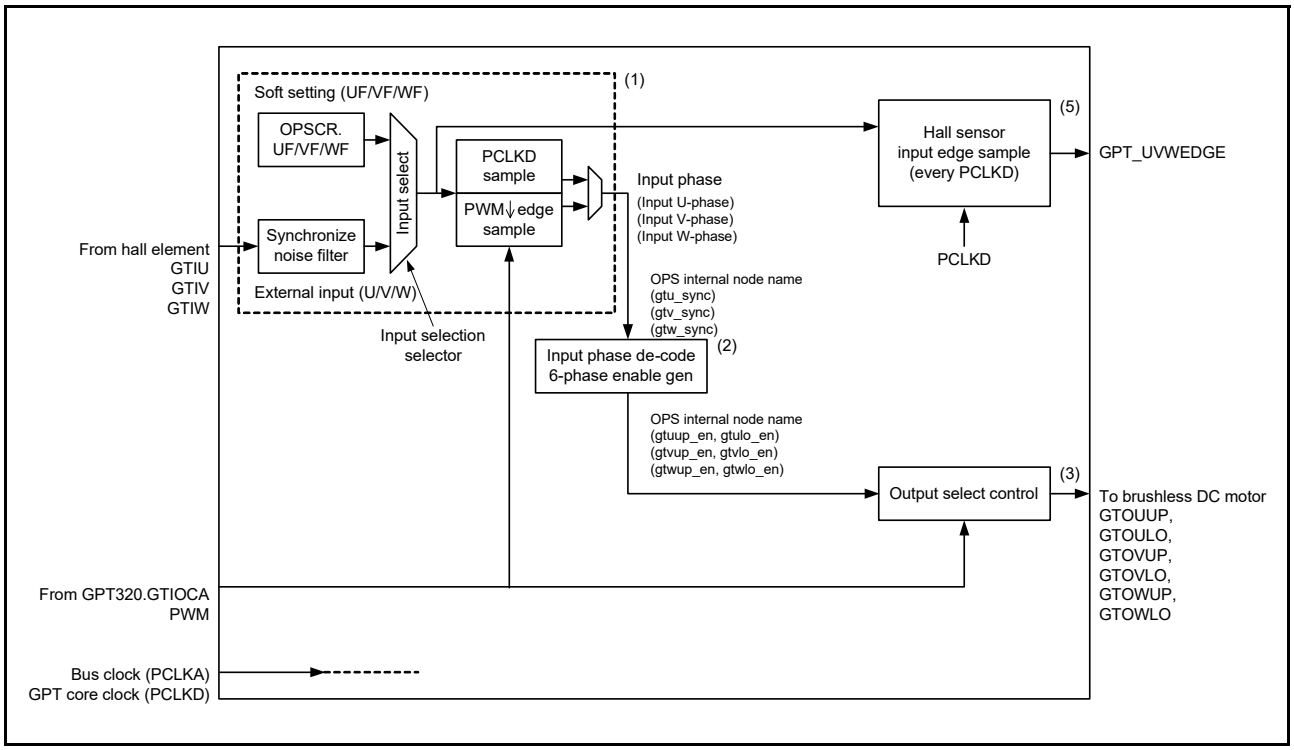


Figure 23.77 GPT_OPS control flow conceptual diagram

Figure 23.78 shows a 6-phase level signals output example of an GPT_OPS operation.

The GPT_UVWEDGE signal in Figure 23.78 is the Hall sensor input edge to ELC output.

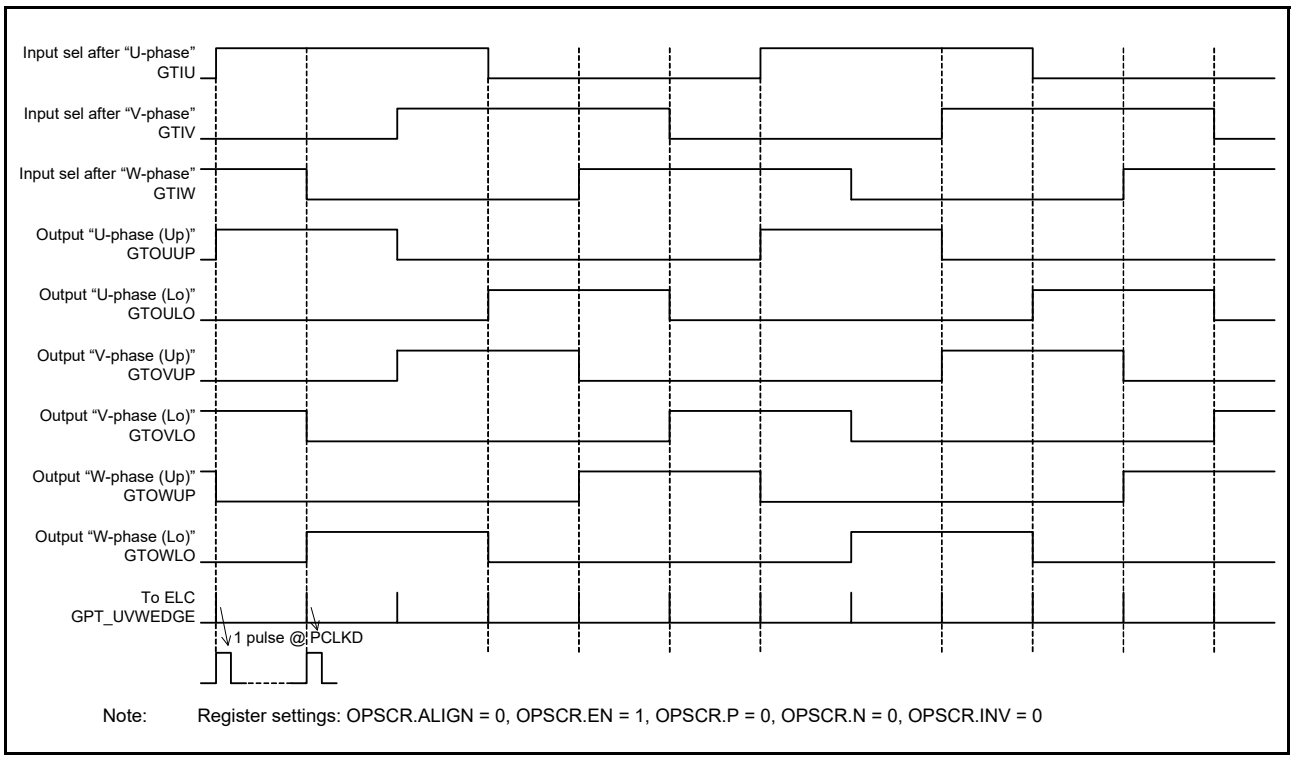


Figure 23.78 6-phase level output operation example

Figure 23.79 shows a 6-phase PWM output example of the GPT_OPS operation (chopper control).

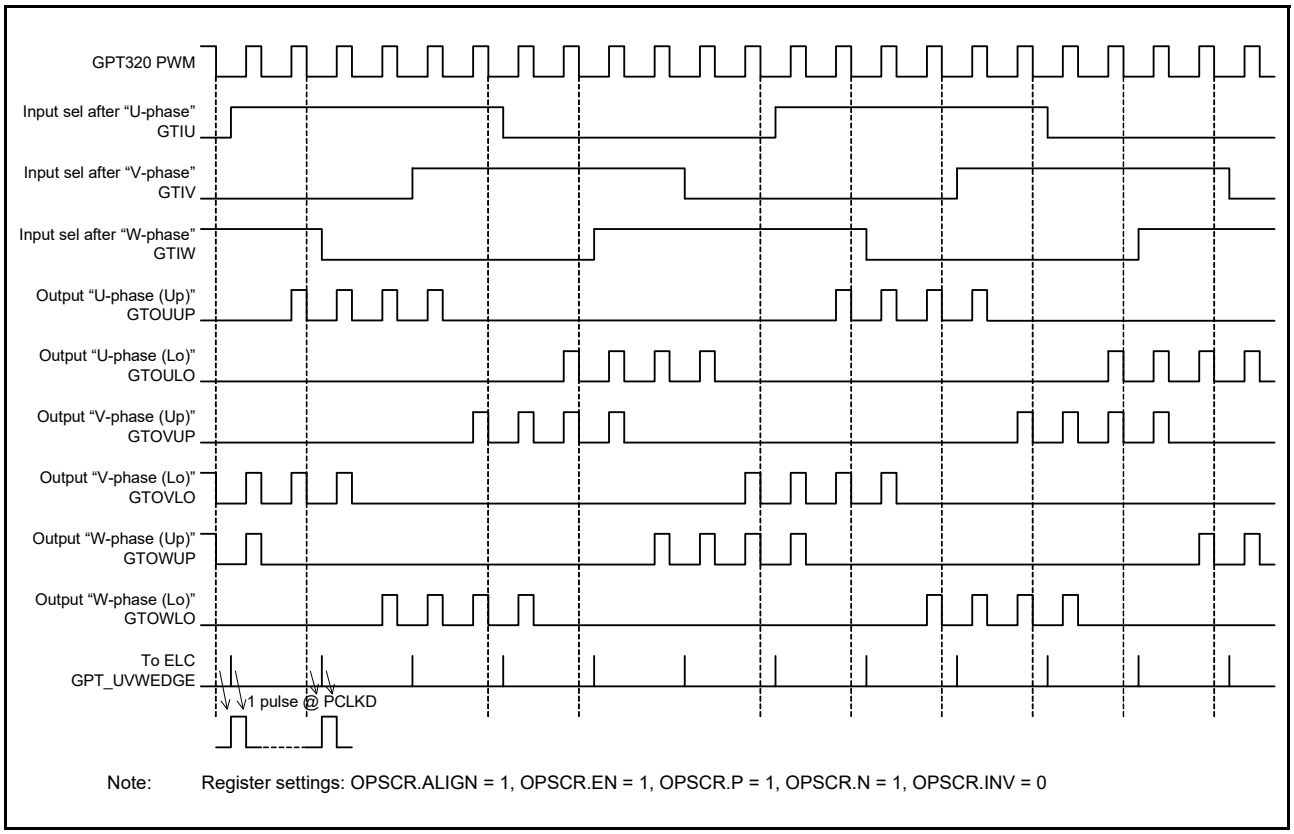


Figure 23.79 6-phase PWM output operation example (chopper control)

Figure 23.80 shows an example of output disable control (6-phase PWM output operation).

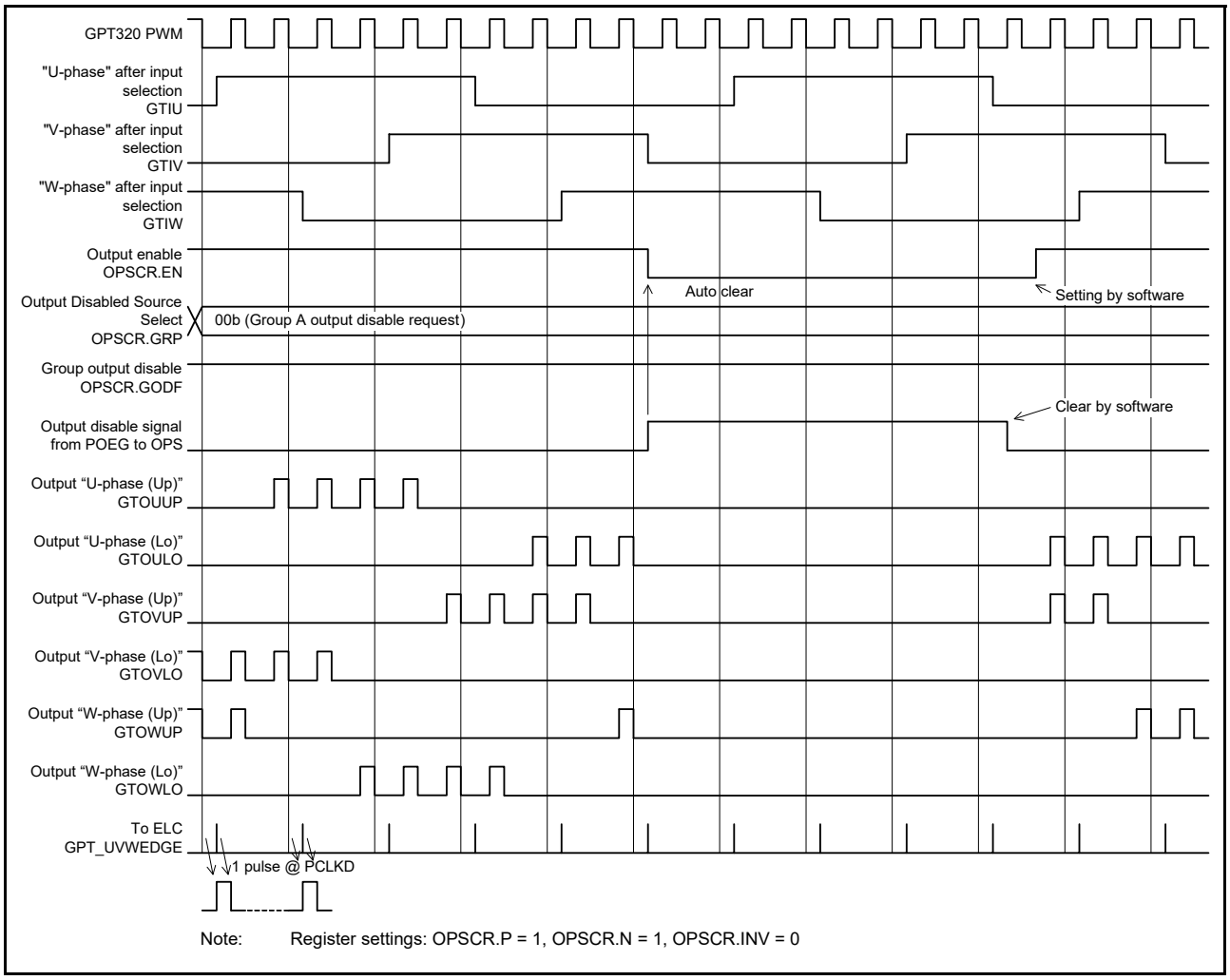


Figure 23.80 Group output disable control operation example

23.3.11.1 Input Selection and Synchronization of External Input Signal

In the GPT_OPS control flow conceptual diagram shown in Figure 23.77, (1) is a selection of input phase from the software settings and external input by the OPSCR.FB bit.

When OPSCR.FB bit = 0, select the external input. Enable the input signal after synchronization with the GPT clock (PCLKD). After carrying out noise filtering (optional), set the external input to the input phase of PWM (PWM of GPT320.GTIOCA) using falling edge sampling with OPSCR.ALIGN bit = 1.

When OPSCR.FB bit = 1, select the soft setting (OPSCR.UF, VF, WF) with the value of the input phase of PWM (PWM of GPT320.GTIOCA) using falling edge sampling with OPSCR.ALIGN bit = 1.

When OPSCR.ALIGN bit = 0, GPT_OPS operates with the input phase of PCLKD synchronization with either OPSCR.FB bit = 0 or OPSCR.FB bit = 1. 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 23.17 shows the input selection process and setting of associated OPSCR bits.

Table 23.17 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		
0	1	External Input at PWM Falling Edge Sampling (PCLKD Sync + Falling Edge Sample)	"Input Phase" Input U-Phase (gtu_sync) Input V-Phase (gtv_sync) Input W-Phase (gtw_sync)
	0	External Input at PCLKD Synchronization Output (PCLKD Sync + Through mode)	
1	1	Software Settings at PWM Falling Edge Sampling (OPSCR.UF, VF, WF of Falling Edge Sample)	
	0	Software Setting Value Selection (= OPSCR.UF/VF/WF value) (= PCLKD Synchronization)	

23.3.11.2 Input Sampling

The OPSCR.U, V, W register indicates the PCLKD sampling results of the input selected by the OPSCR.FB bit.

When OPSCR.FB bit = 0 and after synchronization with the GPT clock (PCLKD) and noise filtering (optional), OPSCR.U, V, W register indicates the sampling results of the external input. When OPSCR.FB bit = 1, OPSCR.U, V, W register is the value (OPSCR.UF, VF, WF) of the soft setting.

23.3.11.3 Input Phase Decode

In the GPT_OPS control flow conceptual diagram shown in Figure 23.77, (2) enables the 6-phase signals by decoding the input phase selected by the OPSCR.FB bit. The 6-phase enable signal is used for internal processing of GPT_OPS.

Table 23.18 shows the decode table of input phase.

Table 23.18 Decode table of input phase

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

23.3.11.4 Output Selection Control

In the GPT_OPS control flow conceptual diagram in Figure 23.77, (3) 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 choose from the level signal or PWM signal (chopper output) for the output phase
- The polarity of the output phase can be set to positive logic or negative logic by the OPSCR. INV bit.

Table 23.19 and Table 23.20 show the output selection control method using the OPSCR register bit.

Table 23.19 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)	
Register OPSCR.EN	Register OPSCR.P	Register 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_en) (gtvup_en) (gtwup_en)	Level Output Mode (Positive phase) (Positive logic)
1	0	1	Level signal (~gtuup_en) (~gtvup_en) (~gtwup_en)	Level Output Mode (Positive phase) (Negative logic)
1	1	0	PWM signal (PWM & gtuup_en) (PWM & gtvup_en) (PWM & gtwup_en)	PWM Output Mode (Positive phase) (Positive logic)
1	1	1	PWM signal (~(PWM & gtuup_en)) (~(PWM & gtvup_en)) (~(PWM & gtwup_en))	PWM Output Mode (Positive phase) (Negative logic)

Table 23.20 Output selection control method (negative phase)

Enable-phase output control	Negative-phase output (N) control	Invert-phase output control	Output port name (negative phase = Lo) (output selection internal node allocation)	
Register OPSCR.EN	Register OPSCR.N	Register 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_en) (gtvlo_en) (gtwlo_en)	Level Output Mode (Negative phase) (Positive logic)
1	0	1	Level signal (~gtulo_en) (~gtvlo_en) (~gtwlo_en)	Level Output Mode (Negative phase) (Negative logic)
1	1	0	PWM signal (PWM & gtulo_en) (PWM & gtvlo_en) (PWM & gtwlo_en)	PWM Output Mode (Negative phase) (Positive logic)
1	1	1	PWM signal (~(PWM & gtulo_en)) (~(PWM & gtvlo_en)) (~(PWM & gtwlo_en))	PWM Output Mode (Negative phase) (Negative logic)

23.3.11.5 Output Selection Control (Group Output Disable Function)

When OPSCR.GODF = 1 and the signal value selected by the OPSCR.GRP bit is high (output disable request), the GPT_OPS output pins are changed to Hi-Z asynchronously and the OPSCR.EN bit is set to 0 by the output disable request signal synchronized with PCLKD. For the return, after clearing the output disable request by software, set the OPSCR.EN = 1.

The timing of OPSCR.EN bit set to 0 is 3 PCLKD cycles after generating the output disable request. To perform output disable control reliably, allow at least 4 PCLKD cycles after generating the output disable request (by clearing the output disable request flag in POEG) until the output disable request is terminated. For an example of the operation of group output disable control, see Figure 23.80.

23.3.11.6 Event Link Controller (ELC) Output

In the GPT_OPS control flow conceptual diagram shown in Figure 23.77, (5) outputs the hall sensor input signal edge to the event link controller.

The hall sensor 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 input is short in duration, the Hall sensor edge input signal is not output at that time.

When OPSCR.FB bit = 0, the Hall sensor input edge signal is the logical OR of the edge signals of the external input phase sampled at PCLKD.

When OPSCR.FB bit = 1, the Hall sensor input edge signal is the logical OR of the edge of the soft setting (OPSCR.UF, VF, WF) sampled at PCLKD.

See Figure 23.78 to Figure 23.80 for examples of the output signal to the ELC.

23.3.11.7 GPT_OPS Start Operation Setting Flow

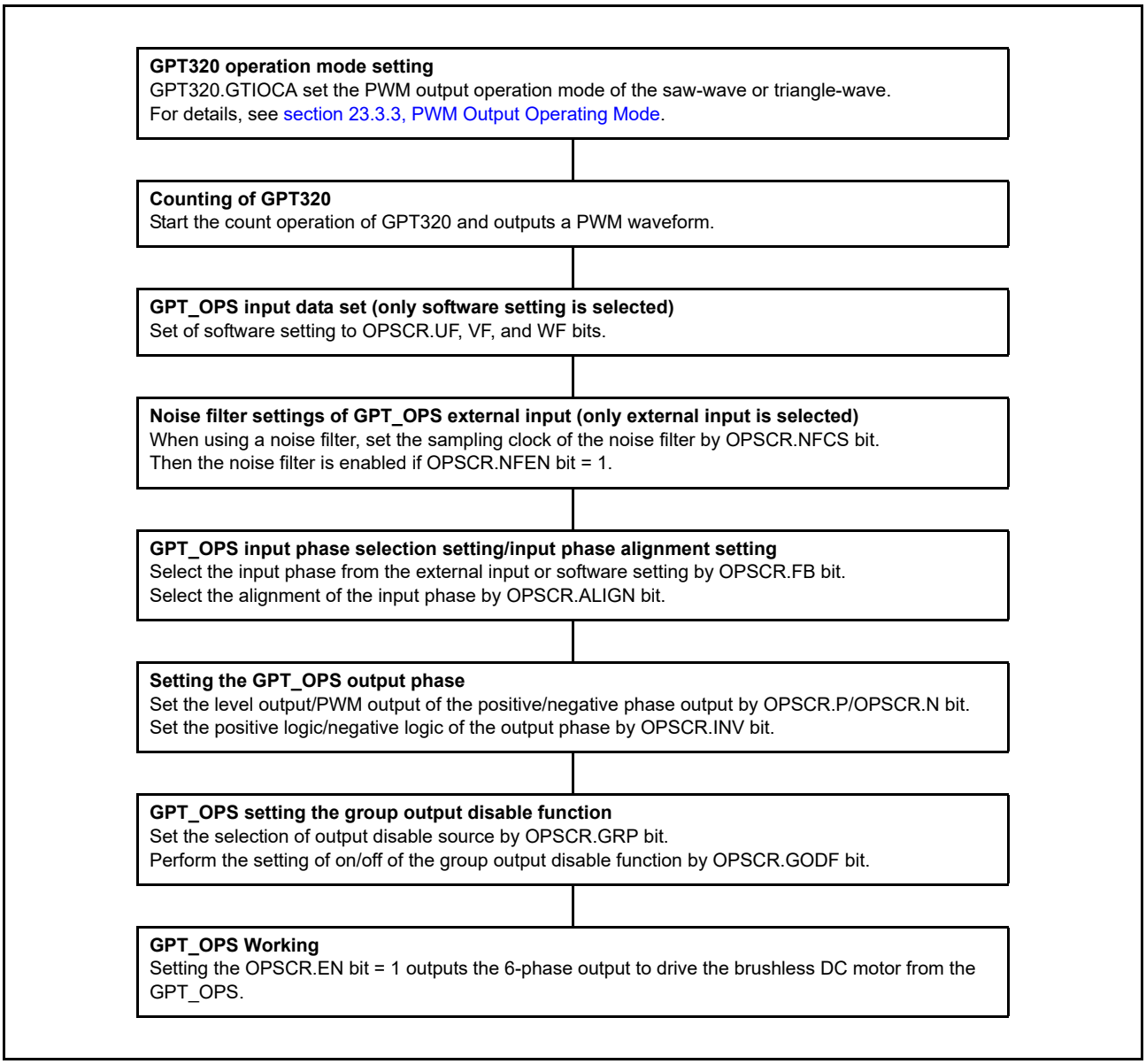


Figure 23.81 Example for setting of GPT_OPS start operation

23.4 Interrupt Sources

23.4.1 Interrupt Sources

The GPT provides the following interrupt sources:

- GTCCR input capture/compare match
- GTCNT counter overflow (GTPR compare match)/underflow.

Each interrupt source has its own status flag. When an interrupt source signal is generated, the associated status flag in GTST is set to 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.

Table 23.21 lists the GPT interrupt sources.

Table 23.21 Interrupt sources (1 of 3)

Channel	Name	Interrupt Source	Interrupt Flag	DMAC/DTC Activation
0	GPT0_CCMPA	GPT320.GTCCRA input capture/compare match	TCFA	Possible
	GPT0_CCMPB	GPT320.GTCCRB input capture/compare match	TCFB	Possible
	GPT0_CMPC	GPT320.GTCCRC compare match	TCFC	Possible
	GPT0_CMPD	GPT320.GTCCRD compare match	TCFD	Possible
	GPT0_CMPE	GPT320.GTCCRE compare match	TCFE	Possible
	GPT0_CMPF	GPT320.GTCCRF compare match	TCFF	Possible
	GPT0_OVF	GPT320.GTCNT overflow (GPT320.GTPR compare match)	TCFPO	Possible
	GPT0_UDF	GPT320.GTCNT underflow	TCFPU	Possible
1	GPT1_CCMPA	GPT321.GTCCRA input capture/compare match	TCFA	Possible
	GPT1_CCMPB	GPT321.GTCCRB input capture/compare match	TCFB	Possible
	GPT1_CMPC	GPT321.GTCCRC compare match	TCFC	Possible
	GPT1_CMPD	GPT321.GTCCRD compare match	TCFD	Possible
	GPT1_CMPE	GPT321.GTCCRE compare match	TCFE	Possible
	GPT1_CMPF	GPT321.GTCCRF compare match	TCFF	Possible
	GPT1_OVF	GPT321.GTCNT overflow (GPT321.GTPR compare match)	TCFPO	Possible
	GPT1_UDF	GPT321.GTCNT underflow	TCFPU	Possible
2	GPT2_CCMPA	GPT322.GTCCRA input capture/compare match	TCFA	Possible
	GPT2_CCMPB	GPT322.GTCCRB input capture/compare match	TCFB	Possible
	GPT2_CMPC	GPT322.GTCCRC compare match	TCFC	Possible
	GPT2_CMPD	GPT322.GTCCRD compare match	TCFD	Possible
	GPT2_CMPE	GPT322.GTCCRE compare match	TCFE	Possible
	GPT2_CMPF	GPT322.GTCCRF compare match	TCFF	Possible
	GPT2_OVF	GPT322.GTCNT overflow (GPT322.GTPR compare match)	TCFPO	Possible
	GPT2_UDF	GPT322.GTCNT underflow	TCFPU	Possible
3	GPT3_CCMPA	GPT323.GTCCRA input capture/compare match	TCFA	Possible
	GPT3_CCMPB	GPT323.GTCCRB input capture/compare match	TCFB	Possible
	GPT3_CMPC	GPT323.GTCCRC compare match	TCFC	Possible
	GPT3_CMPD	GPT323.GTCCRD compare match	TCFD	Possible
	GPT3_CMPE	GPT323.GTCCRE compare match	TCFE	Possible
	GPT3_CMPF	GPT323.GTCCRF compare match	TCFF	Possible
	GPT3_OVF	GPT323.GTCNT overflow (GPT323.GTPR compare match)	TCFPO	Possible
	GPT3_UDF	GPT323.GTCNT underflow	TCFPU	Possible

Table 23.21 Interrupt sources (2 of 3)

Channel	Name	Interrupt Source	Interrupt Flag	DMAC/DTC Activation
4	GPT4_CCMPA	GPT324.GTCCRA input capture/compare match	TCFA	Possible
	GPT4_CCMPB	GPT324.GTCCRB input capture/compare match	TCFB	Possible
	GPT4_CMPC	GPT324.GTCCRC compare match	TCFC	Possible
	GPT4_CMPD	GPT324.GTCCRD compare match	TCFD	Possible
	GPT4_CMPE	GPT324.GTCCRE compare match	TCFE	Possible
	GPT4_CMPF	GPT324.GTCCRF compare match	TCFF	Possible
	GPT4_OVF	GPT324.GTCNT overflow (GPT324.GTPR compare match)	TCFPO	Possible
	GPT4_UDF	GPT324.GTCNT underflow	TCFPU	Possible
5	GPT5_CCMPA	GPT325.GTCCRA input capture/compare match	TCFA	Possible
	GPT5_CCMPB	GPT325.GTCCRB input capture/compare match	TCFB	Possible
	GPT5_CMPC	GPT325.GTCCRC compare match	TCFC	Possible
	GPT5_CMPD	GPT325.GTCCRD compare match	TCFD	Possible
	GPT5_CMPE	GPT325.GTCCRE compare match	TCFE	Possible
	GPT5_CMPF	GPT325.GTCCRF compare match	TCFF	Possible
	GPT5_OVF	GPT325.GTCNT overflow (GPT325.GTPR compare match)	TCFPO	Possible
	GPT5_UDF	GPT325.GTCNT underflow	TCFPU	Possible
6	GPT6_CCMPA	GPT326.GTCCRA input capture/compare match	TCFA	Possible
	GPT6_CCMPB	GPT326.GTCCRB input capture/compare match	TCFB	Possible
	GPT6_CMPC	GPT326.GTCCRC compare match	TCFC	Possible
	GPT6_CMPD	GPT326.GTCCRD compare match	TCFD	Possible
	GPT6_CMPE	GPT326.GTCCRE compare match	TCFE	Possible
	GPT6_CMPF	GPT326.GTCCRF compare match	TCFF	Possible
	GPT6_OVF	GPT326.GTCNT overflow (GPT326.GTPR compare match)	TCFPO	Possible
	GPT6_UDF	GPT326.GTCNT underflow	TCFPU	Possible
7	GPT7_CCMPA	GPT327.GTCCRA input capture/compare match	TCFA	Possible
	GPT7_CCMPB	GPT327.GTCCRB input capture/compare match	TCFB	Possible
	GPT7_CMPC	GPT327.GTCCRC compare match	TCFC	Possible
	GPT7_CMPD	GPT327.GTCCRD compare match	TCFD	Possible
	GPT7_CMPE	GPT327.GTCCRE compare match	TCFE	Possible
	GPT7_CMPF	GPT327.GTCCRF compare match	TCFF	Possible
	GPT7_OVF	GPT327.GTCNT overflow (GPT327.GTPR compare match)	TCFPO	Possible
	GPT7_UDF	GPT327.GTCNT underflow	TCFPU	Possible
8	GPT8_CCMPA	GPT328.GTCCRA input capture/compare match	TCFA	Possible
	GPT8_CCMPB	GPT328.GTCCRB input capture/compare match	TCFB	Possible
	GPT8_CMPC	GPT328.GTCCRC compare match	TCFC	Possible
	GPT8_CMPD	GPT328.GTCCRD compare match	TCFD	Possible
	GPT8_CMPE	GPT328.GTCCRE compare match	TCFE	Possible
	GPT8_CMPF	GPT328.GTCCRF compare match	TCFF	Possible
	GPT8_OVF	GPT328.GTCNT overflow (GPT328.GTPR compare match)	TCFPO	Possible
	GPT8_UDF	GPT328.GTCNT underflow	TCFPU	Possible

Table 23.21 Interrupt sources (3 of 3)

Channel	Name	Interrupt Source	Interrupt Flag	DMAC/DTC Activation
9	GPT9_CCMPA	GPT329.GTCCRA input capture/compare match	TCFA	Possible
	GPT9_CCMPB	GPT329.GTCCRB input capture/compare match	TCFB	Possible
	GPT9_CMPC	GPT329.GTCCRC compare match	TCFC	Possible
	GPT9_CMPD	GPT329.GTCCRD compare match	TCFD	Possible
	GPT9_CMPE	GPT329.GTCCRE compare match	TCFE	Possible
	GPT9_CMPF	GPT329.GTCCRF compare match	TCFF	Possible
	GPT9_OVF	GPT329.GTCNT overflow (GPT329.GTPR compare match)	TCFPO	Possible
	GPT9_UDF	GPT329.GTCNT underflow	TCFPU	Possible

(1) GPTn_CCMPA interrupt (n = 0 to 9)

An interrupt request is generated under the following conditions:

- When the GTCCRA register functions as a compare match register, the GTCNT counter value 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.

(2) GPTn_CCMPB interrupt (n = 0 to 9)

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
- 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.

(3) GPTn_CMPC interrupt (n = 0 to 9)

An interrupt request is generated under the following condition:

- When the GTCCRC register functions as a compare match register, the GTCNT counter value matches with the GTCCRC register.

A compare match is not performed and thus interrupt is not requested under the following conditions:

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (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 9)

An interrupt request is generated under the following condition.:

- When the GTCCRD register functions as a compare match register, the GTCNT counter value matches with the GTCCRD register.

A compare match is not performed and thus interrupt is not requested under the following conditions:

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 10b, 11b (buffer operation with the GTCCRD register).

(5) GPTn_CMPE interrupt (n = 0 to 9)

An interrupt request is generated under the following condition:

- When the GTCCRE register functions as a compare match register, the GTCNT counter value matches with the GTCCRE register.

A compare match is not performed and therefore interrupt is not requested under the following conditions.:

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (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 9)

An interrupt request is generated under the following condition:

- When the GTCCRF register functions as a compare match register, the GTCNT counter value matches with the GTCCRF register.

A compare match is not performed and therefore interrupt is not requested under the following conditions:

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 10b, 11b (buffer operation with the GTCCRF register).

(7) GPTn_OVF interrupt (n = 0 to 9)

An interrupt request is generated under the following conditions:

- In saw-wave mode, interrupt requests are enabled at overflows (when the GTCNT counter value changes from GTPR to 0 during up-counting)
- In triangle-wave mode, interrupt requests are enabled at crests (GTCNT changes from GTPR to GTPR-1)
- In counting by hardware sources, overflow (GTCNT changes from GTPR to 0 in up count) has occurred.

(8) GPTn_UDF interrupt (n = 0 to 9)

An interrupt request is generated under the following conditions:

- In saw-wave 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 (GTCNT changes from 0 to 1)
- In counting by hardware sources, underflow (GTCNT changes from 0 to GTPR in down count) has occurred.

Table 23.22 Interrupt signals and interrupt status flags

Interrupt signal	Interrupt status flag
GPTn_UDF	GTST[7] (TCFPU)
GPTn_OVF	GTST[6] (TCFPO)
GPTn_CMPF	GTST[5] (TCFF)
GPTn_CMPE	GTST[4] (TCFE)
GPTn_CMPD	GTST[3] (TCFD)
GPTn_CMPC	GTST[2] (TCFC)
GPTn_CCMPB	GTST[1] (TCFB)
GPTn_CCMPA	GTST[0] (TCFA)

Note: n = 0 to 9

23.4.2 DMAC/DTC Activation

The DMAC and DTC can be activated by the interrupt in each channel. For details, see section 14, Interrupt Controller Unit (ICU), section 17, DMA Controller (DMAC), and section 18, Data Transfer Controller (DTC).

23.5 Operations Linked by ELC

23.5.1 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 GPT has the following ELC event signals:

- Generating of compare match A interrupt (GPTn_CCMPA)
- Generating of compare match B interrupt (GPTn_CCMPB)
- Generating of compare match C interrupt (GPTn_CMPC)
- Generating of compare match D interrupt (GPTn_CMPD)
- Generating of compare match E interrupt (GPTn_CMPE)
- Generating of compare match F interrupt (GPTn_CMPF)
- Generating of overflow interrupt (GPTn_OVF)
- Generating of underflow interrupt (GPTn_UDF).

Note: n = 0 to 9

23.5.2 Event Signal Inputs from ELC

The GPT can perform the following operations in response to a maximum of eight events from the ELC:

- Start counting, stop counting, clear counting
- Up-counting, down counting
- Input capture.

See section 23.3, Operation for detail on hardware resources.

23.6 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 23.82 shows the timing of noise filtering.

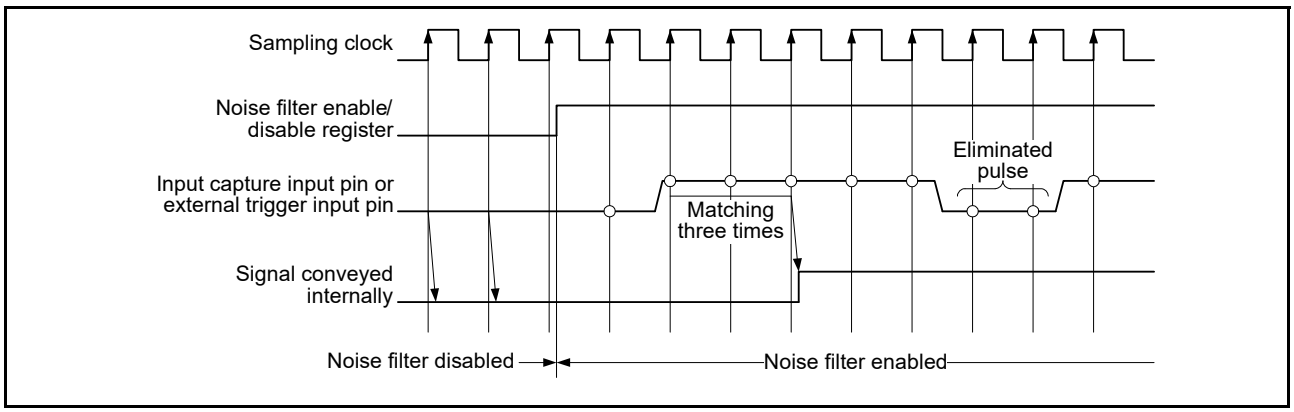


Figure 23.82 Timing of noise filtering

If noise filtering is enabled, the input capture operation or external trigger operation performs on the edges of the noise filtered signal after a delay of a minimum sampling interval × 2 + PCLKD. This is due to the noise filtering for the input capture input or external trigger operation.

23.7 Protection Function

23.7.1 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, GTIBCSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTDTCR, GTDVU.

23.7.2 Disabling of Buffer Operation

If the timing of buffer register write is delayed in relative to the timing for the buffer transfer, buffer operation can be suspended with the GTBER.BD setting. Buffer transfer can be temporarily disabled even though a buffer transfer condition is generated during a buffer register write. This can be done by setting the associated GTBER.BD bit to 1 (buffer operation disabled) before the buffer register write and by clearing the bit to 0 (buffer operation enabled) after completion of writing to all buffer registers.

Figure 23.83 shows an example of operation for disabling buffer operation.

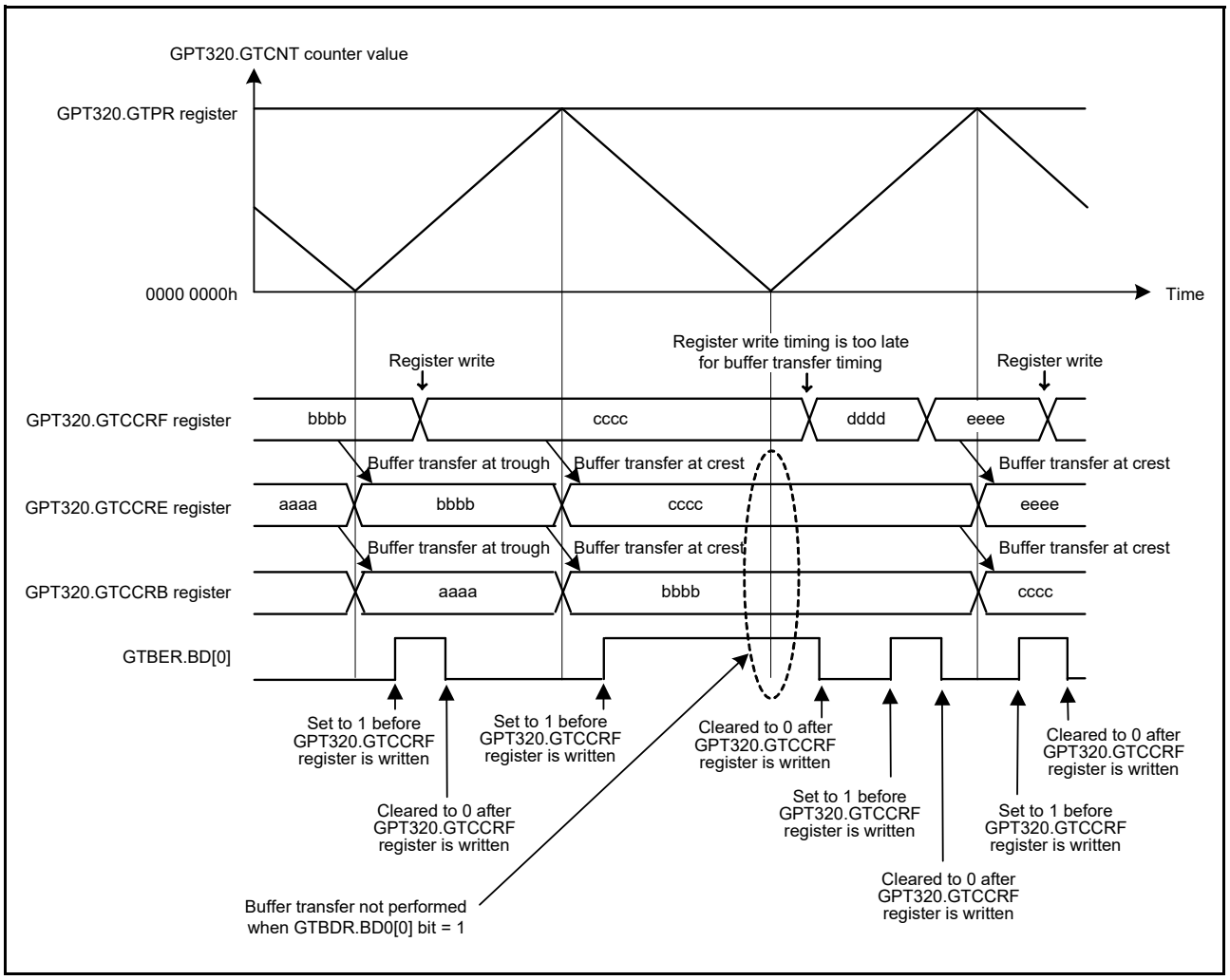


Figure 23.83 Example of operation for disabling buffer operation (triangle waves, double buffer operation, buffer transfer at both troughs and crests)

23.7.3 GTIOC Pin Output Negate Control

For protection from system failure, the output disable control that changes the GTIOC pin output value forcibly is provided for GTIOC pin output by the request of output disable from POEG.

When the GTIOCA pin output value is the same as the GTIOCB pin output value, output protection is required. GPT detects such a case and generates output disable requests to POEG according to the setting of output disable request permission bits, such as GTINTAD.GRPABH, GTINTAD.GRPABL. When the POEG receives output disable requests from each channel and calculates external input using an OR operation, the POEG generates output disable requests to GPT.

One output disable signal (representing the common output disable request signal of the GTIOCA pin and the GTIOCB pin) out of four 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 in association with GTIOR.OADF[1:0] bits for GTIOCA pin and GTIOR.OBDF[1:0] bits for GTIOCB 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. The timing of release of the output disable state is a minimum of 3 PCLKD cycles after terminating the output disable request. To perform output disable control reliably, allow at least 4 PCLKD cycles after generating the output disable request (by clearing the output disable request flag in POEG) until the output disable request is terminated.

When event count is performed or when the output disable state is to be released immediately without waiting for an end of cycle, GTIOR.OADF[1:0] should be set to 00b (for GTIOCA pin) or GTIOR.OBDF[1:0] should be set to 00b (for GTIOCB pin).

Figure 23.84 shows an example of the GTIOC pin output disable control operation.

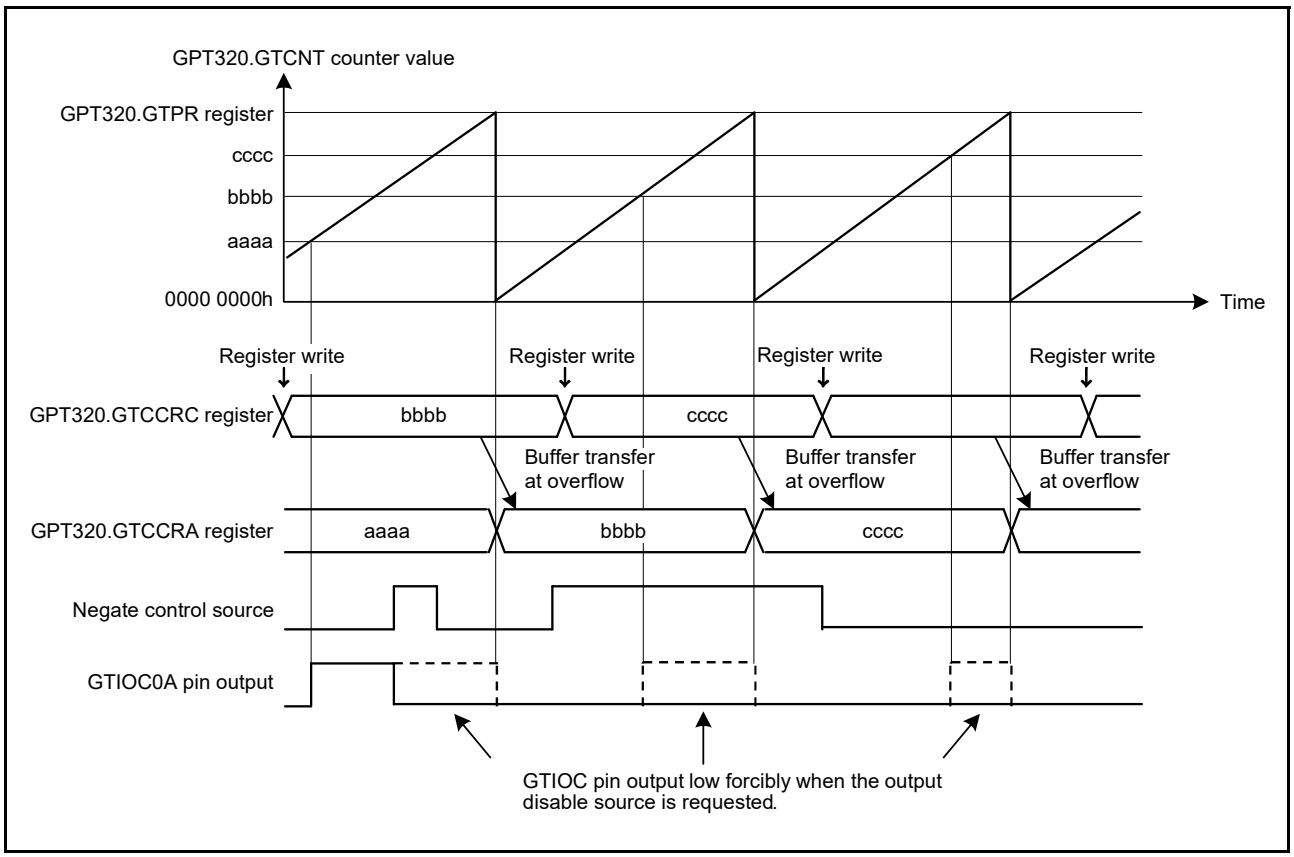


Figure 23.84 Example of GTIOC pin output disable control operation (saw-wave up-counting, buffer operation, active level 1 (high output at GTCCRA compare match, low output at cycle end, low output at output disable)

23.8 Initialization Method of Output Pins

23.8.1 Pin Settings after Reset

The GPT registers are initialized at reset. Start counting after selecting the port pin function by PmnPFS register, setting GTIOR.OAE and GTIOR.OBE bits, and outputting the GPT function to external pins.

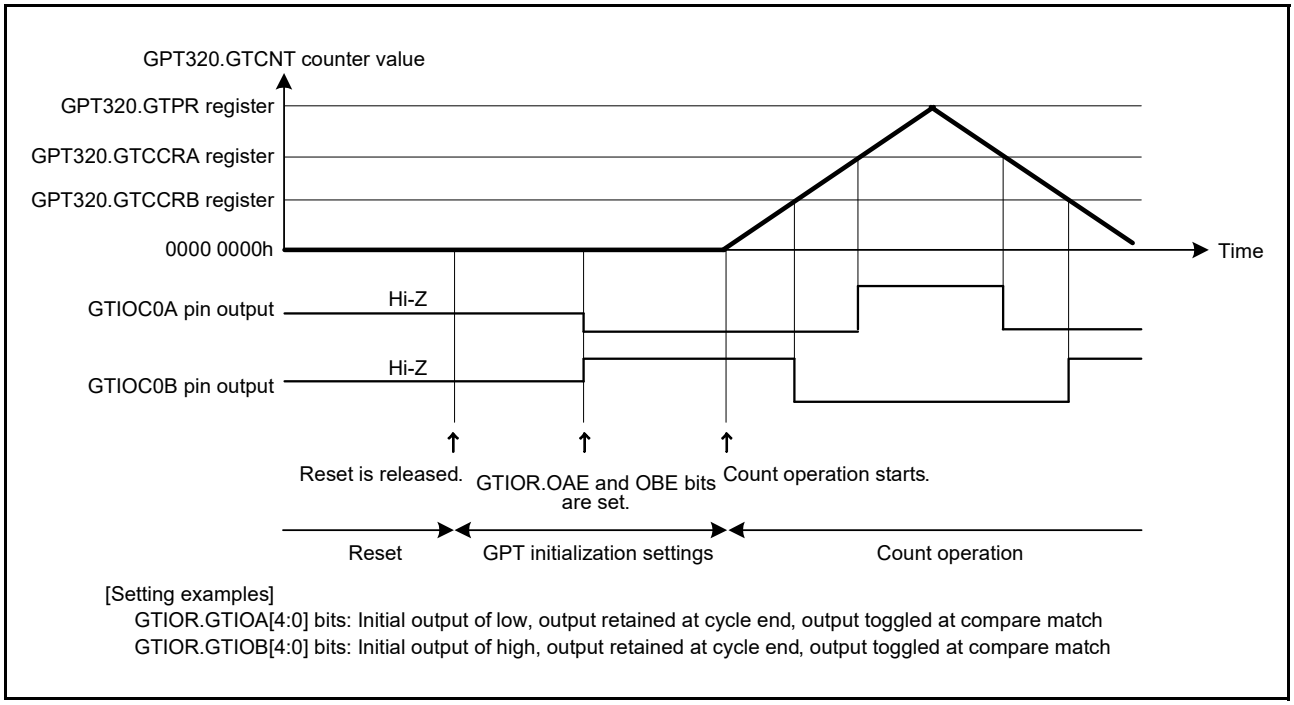


Figure 23.85 Example of pin settings after reset

23.8.2 Pin Initialization Due to Error during Operation

If an error occurs during GPT operation, the following four types of pin processing can be performed before pin initialization:

- Set OAHLD and OBHLD bits in GTIOR to 1 and retain the outputs at count stop
- Set 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, and PmnPFS registers 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 PmnPFS.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.

When 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 resumes, operation continues from where it stops. If counting stops, registers should be initialized before counting starts.

23.9 Usage Notes

23.9.1 Module Stop Function Setting

Operation of the GPT can be disabled or enabled by the module stop control register. The initial setting is for operation of the GPT to be stopped. Register access is enabled by clearing the module-stop state. For details, see section 11, Low Power Modes.

23.9.2 Settings of GTCCRn 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 the following conditions: GTDVU < GTCCRA and 0 < GTCCRA < GTPR.

(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 should 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 = 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, correct output waveforms with secured dead time may not be obtained.

- In up-counting: GTCCRC < GTCCRD, GTCCRC > GTDVU, GTCCRD < GTPR - GTDVU
- In down-counting: GTCCRC > GTCCRD, GTCCRC < GTPR – GTDVU, GTCCRD > GTDVU

(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

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.

23.9.3 Setting the Range of GTCNT Counter

The GTCNT counter register must be set with the range of 0 ≤ GTCNT ≤ GTPR.

23.9.4 GTCNT Counter Start/Stop

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[2:0]. When GTCR.CST is updated, the GTCNT counter starts/stops after a count clock that is selected in GTCR.TPCS[2:0]. Therefore, an event generated before the GTCNT counter actually starts is ignored, resulting in cases where an event is accepted or an interrupt occurs after GTCR.CST is set to 0.

23.9.5 Priority Order of each Event

(1) GTCNT register

Table 23.23 shows a priority order of events updating GTCNT register.

Table 23.23 Priority order of sources updating GTCNT

Source of updating GTCNT	Priority order
Writing by CPU (Writing to GTCNT/GTCLR)	High ↑ Low
Clear by hardware sources set in GTCSR	
Count up or down by hardware sources set in GTUPSR/GTDNSR	
Count operation	

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 a priority over starting/stopping by hardware sources.

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, pre-update data is read.

(3) GTCCRm registers (m = A to F)

When there is a conflict between input capture/buffer transfer operation and the writing to GTCCRm registers, the writing to GTCCRm registers has a 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 the writing to the GTPR register, the writing to GTPR register has a priority over buffer transfer operation. When there is a conflict between updating GTPR register and reading by the CPU, pre-update data is read.

24. Asynchronous General Purpose Timer (AGT)

24.1 Overview

The Asynchronous General Purpose Timer (AGT) is a 16-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events.

This 16-bit timer consists of a reload register and a down counter. The reload register and the down counter are allocated in the same address, and they can be accessed with the AGT register.

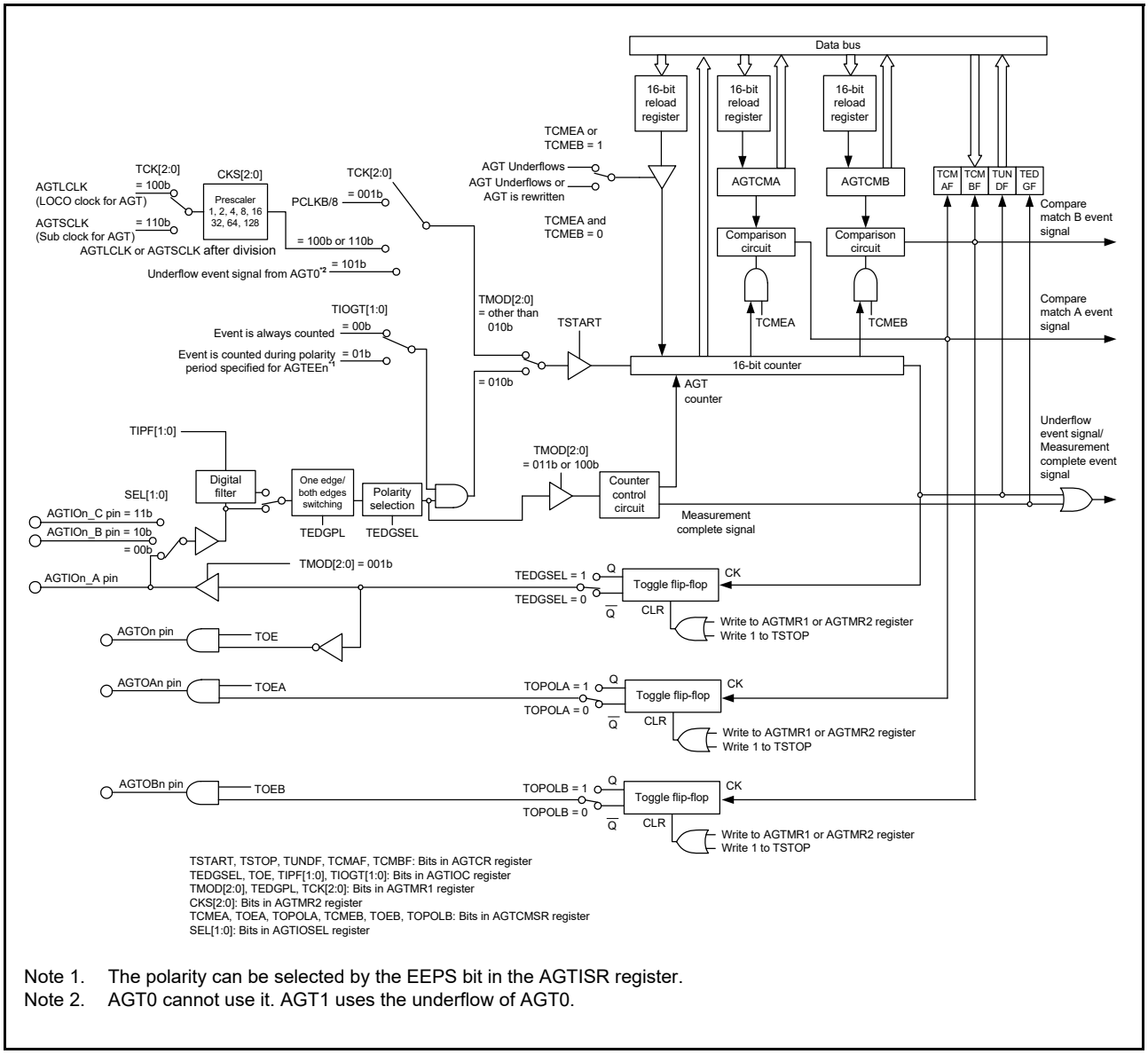
Table 24.1 lists the AGT specifications, Figure 24.1 shows the AGT block diagram, and Table 24.2 lists the AGT pin configuration.

Note: Regardless of the use of the VBATT function, set the VBTCR1.BPWSWSTP bit to 1 before accessing the AGT registers after a cold start. For details, see Figure 12.3 Setting Flowchart of the VBTCR1.BPWSWSTP bit, in section 12, Battery Backup Function.

Table 24.1 AGT 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
Count source (Operating clock)*2		PCLKB/8, AGTLCLK/d, AGTSCLK/d, or underflow signal of AGT0*1 selectable. (d = 1, 2, 4, 8, 16, 32, 64, or 128)
Interrupt/Event link function (Output)		• Unterflow event signal or measurement complete event signal – When the counter underflows – When the measurement of the active width of the external input (AGTIOn) is complete in pulse width measurement mode – When the set edge of the external input (AGTIOn) is input in pulse period measurement mode. • Compare match A event signal – When the values of AGT and AGTCMA matched (Compare match A function enabled) • Compare match B event signal – When the values of AGT and AGTCMB matched (Compare match B function enabled) • Recovery from Software Standby mode can be perfomed with AGT1_AGTI, AGT1_AGTCMAI, or AGT1_AGTCMBI.
Selectable functions		• Compare match function One or two of the Compare Match A Register and Compare Match B Register is selectable.

Note 1. AGT0 cannot use it. AGT1 connects directly with the underflow event signal from the AGT0 timer.
Note 2. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source clock.



Note 1. The polarity can be selected by the EEPS bit in the AGTISR register.
Note 2. AGT0 cannot use it. AGT1 uses the underflow of AGT0.

Figure 24.1 AGT block diagram

Table 24.2 AGT I/O pins

Pin Name	I/O	Function
AGTEEn	Input	External event input for AGT
AGTIOn*1	Input*1/output	External event input and pulse output for AGT
AGTOn	Output	Pulse output for AGT
AGTOAn	Output	Output compare match A output for AGT
AGTOBn	Output	Output compare match B output for AGT

Note: Channel number (n = 0, 1)
Note 1. When AGTIOn are assigned P402 and P403, AGTIOn can only be used as inputs. In this case, AGTIOn can also be used in VBATT operation.

24.2 Register Descriptions

24.2.1 AGT Counter Register (AGT)

Address(es): AGT0.AGT 4008 4000h, AGT1.AGT 4008 4100h

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

Bit	Description	Setting Range	R/W
b15 to b0	16-bit counter and reload register *1, *2	0000h to FFFFh	R/W

Note 1. When 1 is written to the TSTOP bit in the AGTCR register, the 16-bit counter is forcibly stopped and set to FFFFh.
Note 2. When the TCK[2:0] bit setting in the AGTMR1 register is a value other than 001b (PCLKB/8), if the AGT register is set to 0000h, a request signal to the ICU, the DTC and the ELC is generated once immediately after the count starts. The AGTOn and AGTIOn output is toggled.
When the AGT register is set to 0000h 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 output toggles even during a period other than the specified count period. When the AGT register is set to 0001h or more, a request signal is generated each time AGT underflows.

AGT is a 16-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 24.3.1, Reload Register and Counter Rewrite Operation. The AGT register can be set by a 16-bit memory manipulation instruction.

24.2.2 AGT Compare Match A Register (AGTCMA)

Address(es): AGT0.AGTCMA 4008 4002h, AGT1.AGTCMA 4008 4102h

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

Bit	Description	Setting range	R/W
b15 to b0	16-bit compare match A data is stored.*1	0000h to FFFFh	R/W

Note 1. Set the AGTCMA register to FFFFh when Compare match A is not to be 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 the compare register A change according to the TSTART bit in the AGTCR register. For details, see section 24.3.2, Reload Register and Compare Register A/B Rewrite Operation. The AGTCMA register can be set by a 16-bit memory manipulation instruction.

24.2.3 AGT Compare Match B Register (AGTCMB)

Address(es): AGT0.AGTCMB 4008 4004h, AGT1.AGTCMB 4008 4104h

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

Bit	Description	Setting range	R/W
b15 to b0	16-bit compare match B data is stored.*1	0000h to FFFFh	R/W

Note 1. Set the AGTCMB register to FFFFh 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 the compare register B change according to the TSTART bit in the AGTCR register. For details, see section 24.3.2, Reload Register and Compare Register A/B Rewrite Operation. The AGTCMB register can be set by a 16-bit memory manipulation instruction.

24.2.4 AGT Control Register (AGTCR)

Address(es): AGT0.AGTCR 4008 4008h, AGT1.AGTCR 4008 4108h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCMBF	TCMAF	TUNDF	TEDGF	—	TSTOP	TCSTF	TSTART
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TSTART	AGT count start*2	0: Count stops 1: Count starts.	R/W
b1	TCSTF	AGT count status flag*2	0: Count stops 1: Count in progress.	R
b2	TSTOP	AGT count forced stop*1	0: Writing is invalid 1: The count is forcibly stopped.	W
b3	—	Reserved	The read value is 0. The write value should be 0.	R/W
b4	TEDGF	Active edge judgment flag	0: No active edge received 1: Active edge received.	R/(W)*3
b5	TUNDF	Underflow flag	0: No underflow 1: Underflow.	R/(W)*3
b6	TCMAF	Compare match A flag	0: No match 1: Match.	R/(W)*3
b7	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 and TCSTF bits are initialized at the same time. The pulse output level is also initialized. The read value is 0.
Note 2. For information on using TSTART and TCSTF bits, see section 24.4.1, 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 this bit is set to 1, the TCSTF bit 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 24.4.1, Count Operation Start and Stop Control.

TCSTF flag (AGT count status flag)

[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 this bit, the count is forcibly stopped. The read value is 0.

TEDGF flag (Active edge judgment flag)

[Setting condition]

- When the measurement of the active width of the external input (AGTIOn) is complete in pulse width measurement mode
- When the set edge of the external input (AGTIOn) is input in pulse period measurement mode.

[Clearing condition]

- When 0 is written to this flag by software.

TUNDF flag (Underflow flag)

[Setting condition]

- When the counter underflows.

[Clearing condition]

- When 0 is written to this flag by software.

TCMAF flag (Compare match A flag)

[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)

[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.

24.2.5 AGT Mode Register 1 (AGTMR1)

Address(es): AGT0.AGTMR1 4008 4009h, AGT1.AGTMR1 4008 4109h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	TCK[2:0]			TEDGPL	TMOD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	TMOD[2:0]	Operating mode*3	b2 b0 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. Other settings are prohibited.	R/W
b3	TEDGPL	Edge polarity*4	0: Single-edge 1: Both-edge.	R/W
b6 to b4	TCK[2:0]	Count source*1, *2, *5	b6 b4 0 0 0: Reserved 0 0 1: PCLKB/8 1 0 0: Divided clock AGTLCLK specified by CKS[2:0] bits in the AGTMR2 register 1 0 1: Underflow event signal from AGT0*6 1 1 0: Divided clock AGTSCLK specified by CKS[2:0] bits in the AGTMR2 register. Other settings are prohibited.	R/W
b7	—	Reserved	The read value is 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 of the AGT (n = 0, 1). For details on the output level at initialization, see the description of section 24.2.7, AGT I/O Control Register (AGTIOC).

Note 1. When event counter mode is selected, the external input (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. Count sources should be switched when both the TSTART and TCSTF bits 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 and TCSTF bits in the AGTCR register are set to 0 (count stops). 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, select AGTLCLK or AGTSCLK.

Note 6. AGT0 cannot use it (setting prohibited). AGT1 uses the AGT0 underflow.

24.2.6 AGT Mode Register 2 (AGTMR2)

Address(es): AGT0.AGTMR2 4008 400Ah, AGT1.AGTMR2 4008 410Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	LPM	—	—	—	—	CKS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CKS[2:0]	AGTLCLK/AGTSCLK count source clock frequency division ratio *1, *2, *3	b2 b0 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
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	LPM	Low Power Mode	0: Normal mode 1: Low power mode.	R/W

Note 1. Do not rewrite the CKS[2:0] bit during count operation. The CKS[2:0] bit should be rewritten when both the TSTART and TCSTF bits in the AGTCR register are set to 0 (count stops).
Note 2. When the count source is AGTLCLK or AGTSCLK, CKS[2:0] switch is valid.
Note 3. Do not switch the TCK[2:0] bits in the AGTMR1 register when CKS[2:0] are not 000b. Switch the TCK[2:0] bits in the AGTMR1 register after CKS[2:0] are set to 000b, and wait for 1 cycle of the count source.

LPM bit (Low Power Mode)

This 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 set to 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:

- AGT — Read AGT register twice. Only the second reading of data is valid
- AGT, AGTCMA, AGTCMB, and AGTCR — Allow at least 2 cycles of the count source clock when writing to the register.

24.2.7 AGT I/O Control Register (AGTIOC)

Address(es): AGT0.AGTIOC 4008 400Ch, AGT1.AGTIOC 4008 410Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	TIOGT[1:0]		TIPF[1:0]		—	TOE	—	TEDGSEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TEDGSEL	I/O polarity switch	Function varies depending on the operating mode (see Table 24.3 and Table 24.4). The TEDGSEL bit switches the AGTOn output polarity and the AGTIOn input/output edge and polarity. In pulse output mode, it only controls the polarity of the AGTOn output and AGTIOn output. AGTOn output and AGTIOn output are initialized when the AGTMR1 register is written and the TSTOP bit in the AGTCR register is written with 1.	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	TOE	AGTOn output enable	0: AGTOn output disabled. 1: AGTOn output enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	TIPF[1:0]	Input filter *3	b5 b4 0 0: No filter 0 1: Filter sampled at PCLKB 1 0: Filter sampled at PCLKB/8 1 1: Filter sampled at PCLKB/32. These bits specify 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.	R/W
b7, b6	TIOGT[1:0]	Count control *1, *2, *4	b7 b6 0 0: Event is always counted. 0 1: Event is counted during polarity period specified for AGTEEn. Other settings are 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, and battery backup function, the digital filter function cannot be used.
Note 4. When using in VBATT operation, set TIOGT[1:0] = 00b (event is always counted).

Table 24.3 AGTIOn 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) 1: Output is started at low (Initialization level: Low).
Event counter mode	0: Count at rising edge 1: Count at 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 24.4 AGTOn output polarity switching

Operating mode	Function
All modes	0: Output is started at low (Initialization level: Low) 1: Output is started at high (Initialization level: High).

24.2.8 AGT Event Pin Select Register (AGTISR)

Address(es): AGT0.AGTISR 4008 400Dh, AGT1.AGTISR 4008 410Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	EEPS	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	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
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

24.2.9 AGT Compare Match Function Select Register (AGTCMSR)

Address(es): AGT0.AGTCMSR 4008 400Eh, AGT1.AGTCMSR 4008 410Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	TOPOLB	TOEB	TCMEB	—	TOPOLA	TOEA	TCMEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TCMEA	Compare match A register enable *1, *2	0: Disable compare match A register 1: Enable compare match A register.	R/W
b1	TOEA	AGTOAn output enable *1, *2	0: AGTOAn output disabled 1: AGTOAn output enabled.	R/W
b2	TOPOLA	AGTOAn polarity select *1, *2	0: AGTOAn Output is started at low 1: AGTOAn Output is started at high.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	TCMEB	Compare match B register enable *1, *2	0: Disable compare match B register 1: Enable compare match B register.	R/W
b5	TOEB	AGTOBn output enable *1, *2	0: AGTOBn output disabled 1: AGTOBn output enabled.	R/W
b6	TOPOLB	AGTOBn polarity select *1, *2	0: AGTOBn Output is started at low 1: AGTOBn Output is started at high.	R/W
b7	—	Reserved	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. The AGTCMSR register should be rewritten when both the TSTART and TCSTF bits 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.

24.2.10 AGT Pin Select Register (AGTIOSEL)

Address(es): AGT0.AGTIOSEL 4008 400Fh, AGT1.AGTIOSEL 4008 410Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TIES	—	—	SEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	SEL[1:0]	AGTIOn Pin Select	b1 b0 0 0: Select the AGTIOn of AGTIOn_A 0 1: Setting prohibited 1 0: Select the AGTIOn of AGTIOn_B AGTIOn_B is input only. It is not possible to output. 1 1: Select the AGTIOn of AGTIOn_C AGTIOn_C is input only. It is not possible to output.	R/W*1
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	TIES	AGTIOn Input Enable	0: External event input is disabled during Software Standby mode 1: External event input is enabled during Software Standby mode.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. During AGT operation using the voltage from the VBATT pin, AGTIOn_B and AGTIOn_C within the backup power supply area can only be used as external event input pins for the AGT. AGTIOn_A cannot be used. AGTIOn_B and AGTIOn_C are input only.
Note 2. When AGTIOn_A is selected, you must set the Pin Function Select Register. See section 20, I/O Ports.
Note 3. When AGTIOn_B, AGTIOn_C is selected, you must set the VBTICTLR register. See section 12, Battery Backup Function.

The AGTIOSEL register sets the AGTIOn pin when using the AGTIOn in Software Standby mode. The AGTIOSEL register can be set with an 8-bit memory manipulation instruction.

SEL[1:0] bits (AGTIOn Pin Select)

These bits select the AGTIOn pin function.

TIES bit (AGTIOn Input Enable)

Enables or disables an external event input.

24.3 Operation

24.3.1 Reload Register and Counter Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and the counter changes 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 (Compare Match A/B Registers 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 TCMEB bit is 1 (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 24.2 and Figure 24.3 show the timing of rewrite operation with TSTART bit value and TCMEA or TCMEB bit value.

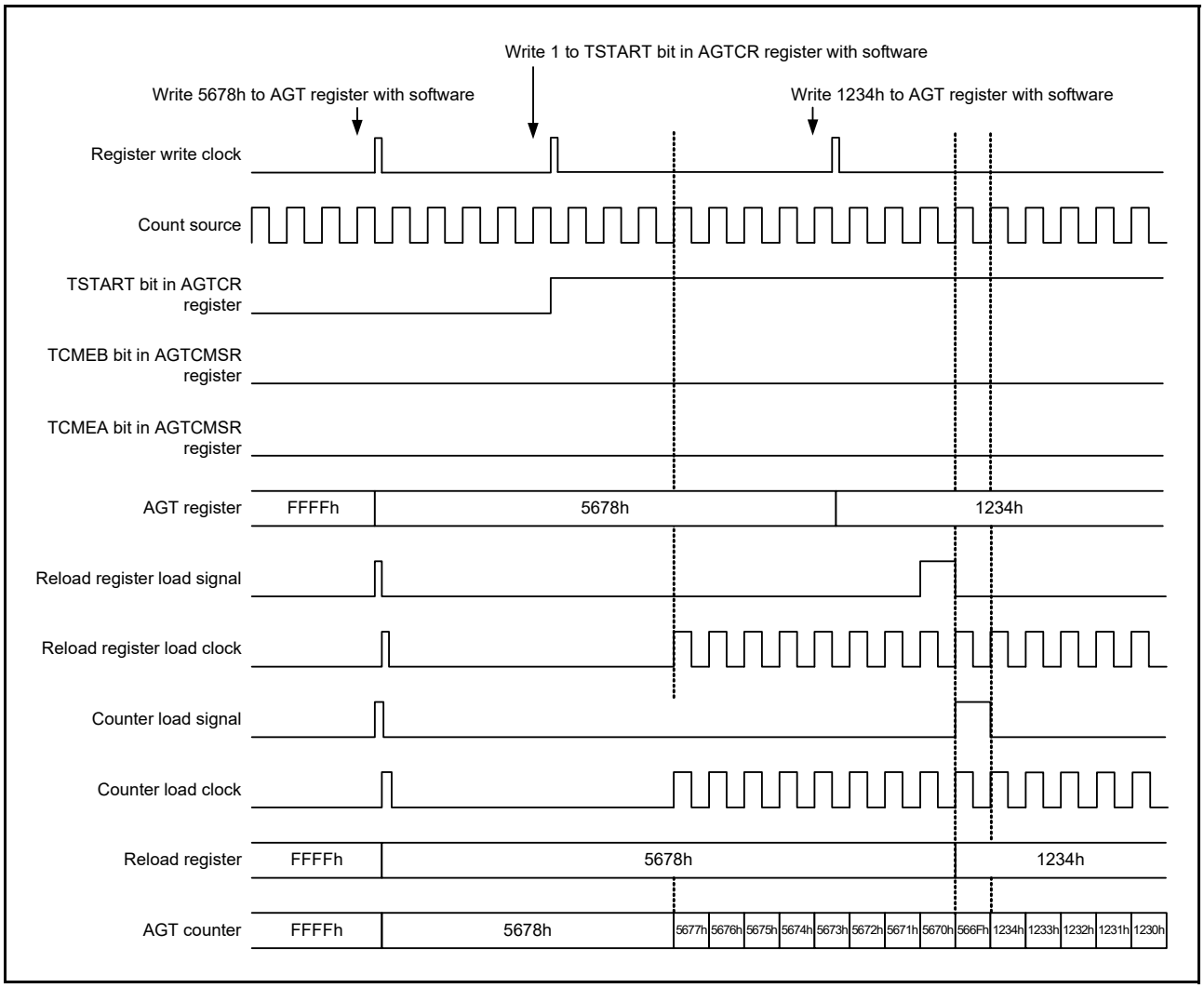


Figure 24.2 Timing of rewrite operation with TSTART, TCMEA, or TCMEB bit values when Compare Match A Register or Compare Match B Register is invalid

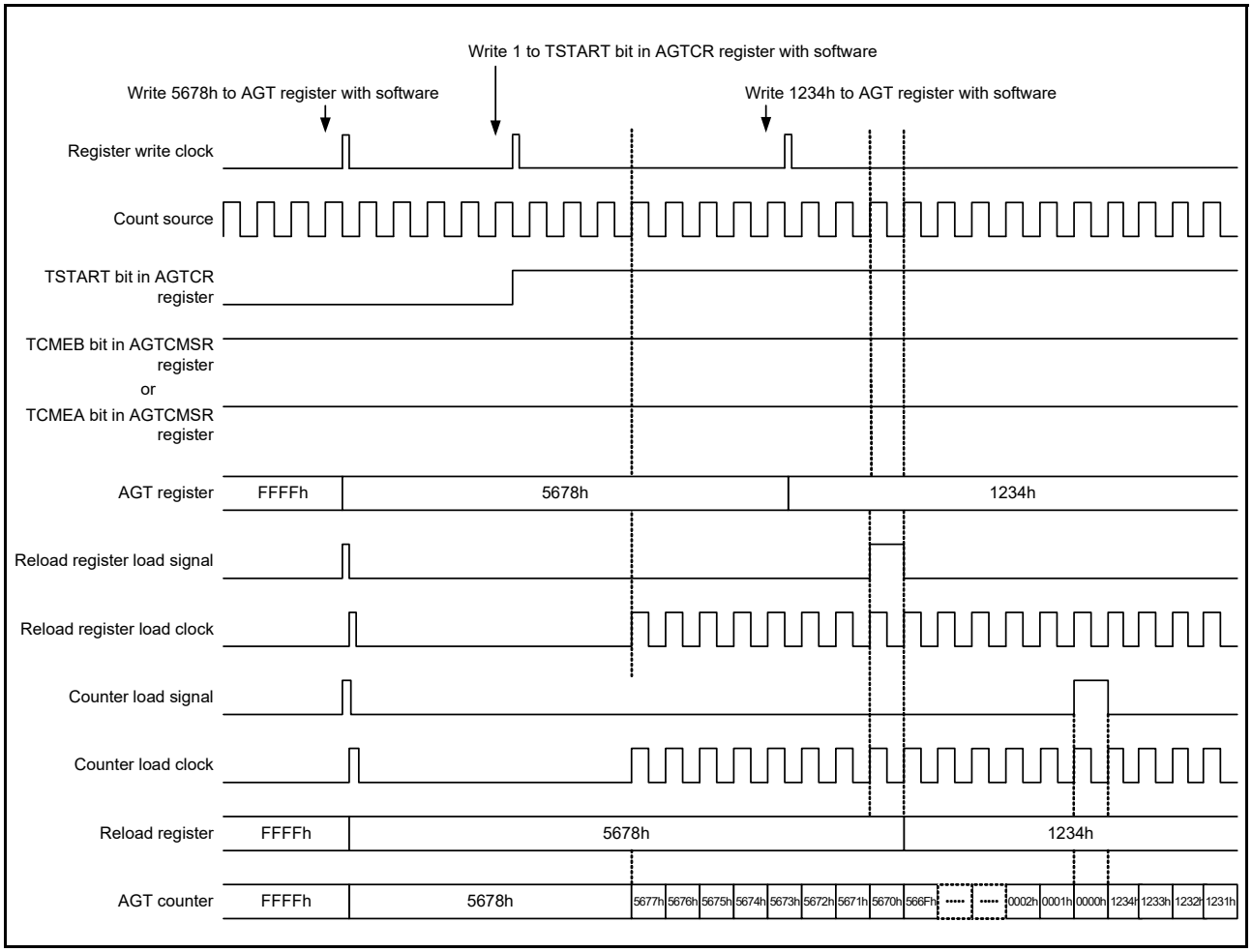


Figure 24.3 Timing of rewrite operation with TSTART, TCMEA, or TCMEB bit values when Compare Match A Register or Compare Match B Register is valid

24.3.2 Reload Register and Compare Register A/B Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to compare register A/B depends on the value in 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 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 24.4 shows the timing of rewrite operation with TSTART bit value for compare register A. Compare register B is of the same timing as compare register A.

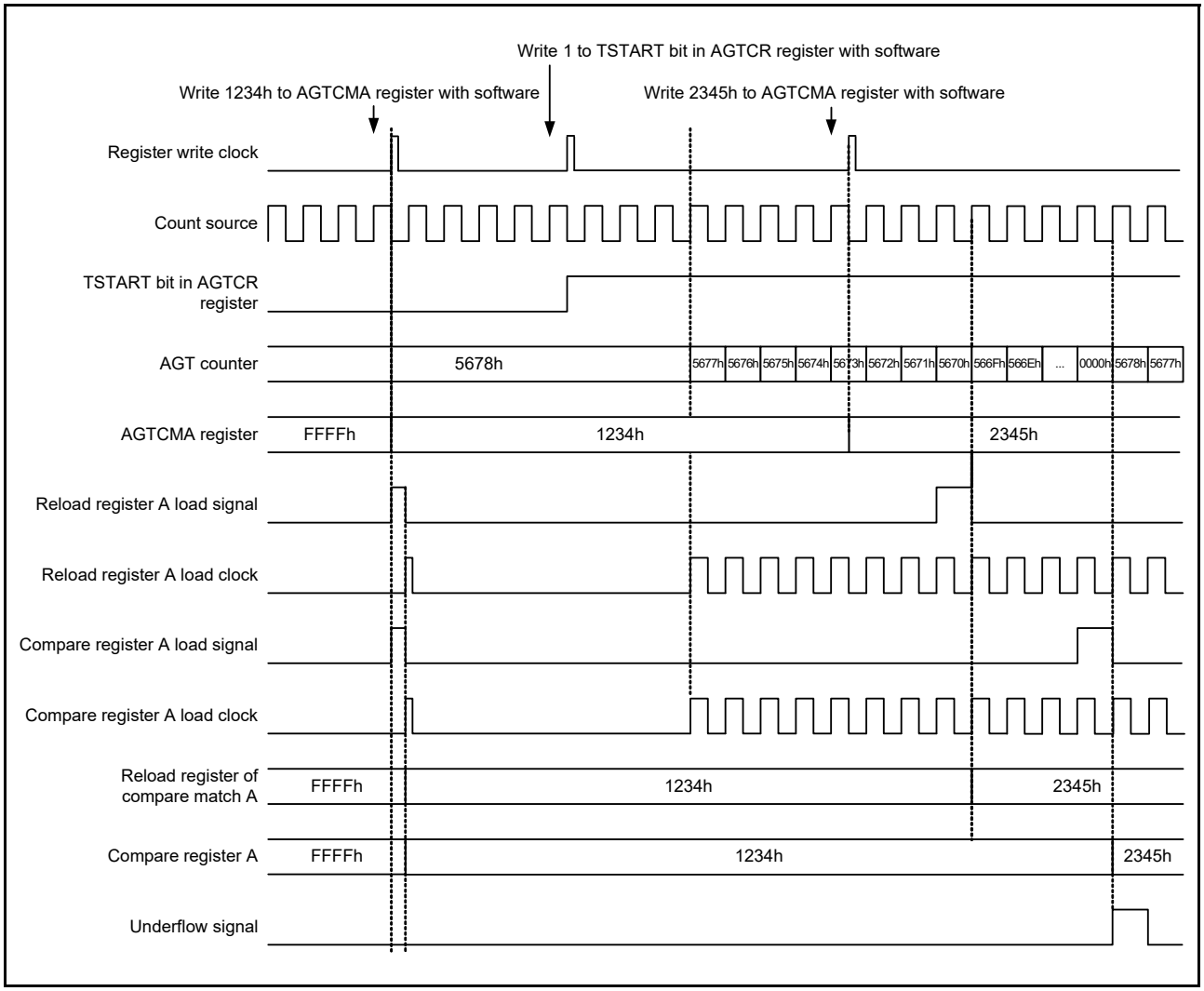


Figure 24.4 Timing of rewrite operation with the TSTART bit value (for compare register A)

24.3.3 Timer Mode

In this mode, the AGT counter is decremented by the count source selected by 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 0000h and the next count source is input, an underflow occurs and an interrupt request is generated.

Figure 24.5 shows the operation example in timer mode.

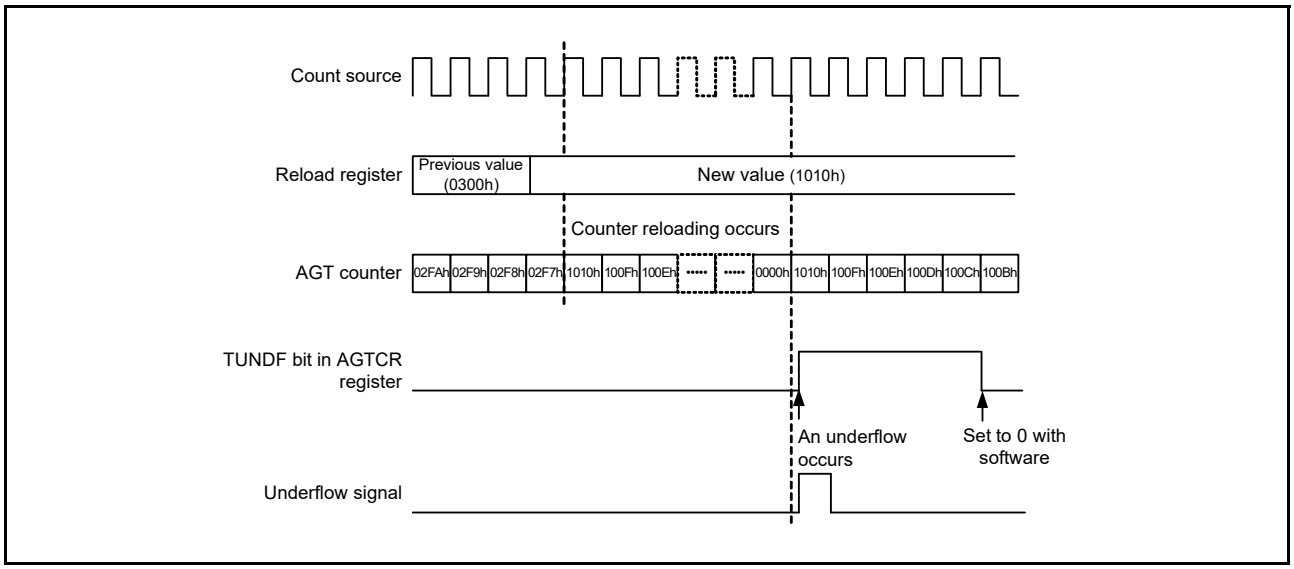


Figure 24.5 Operation example in timer mode

24.3.4 Pulse Output Mode

In this 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 0000h 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 24.6 shows the operation example in pulse output mode.

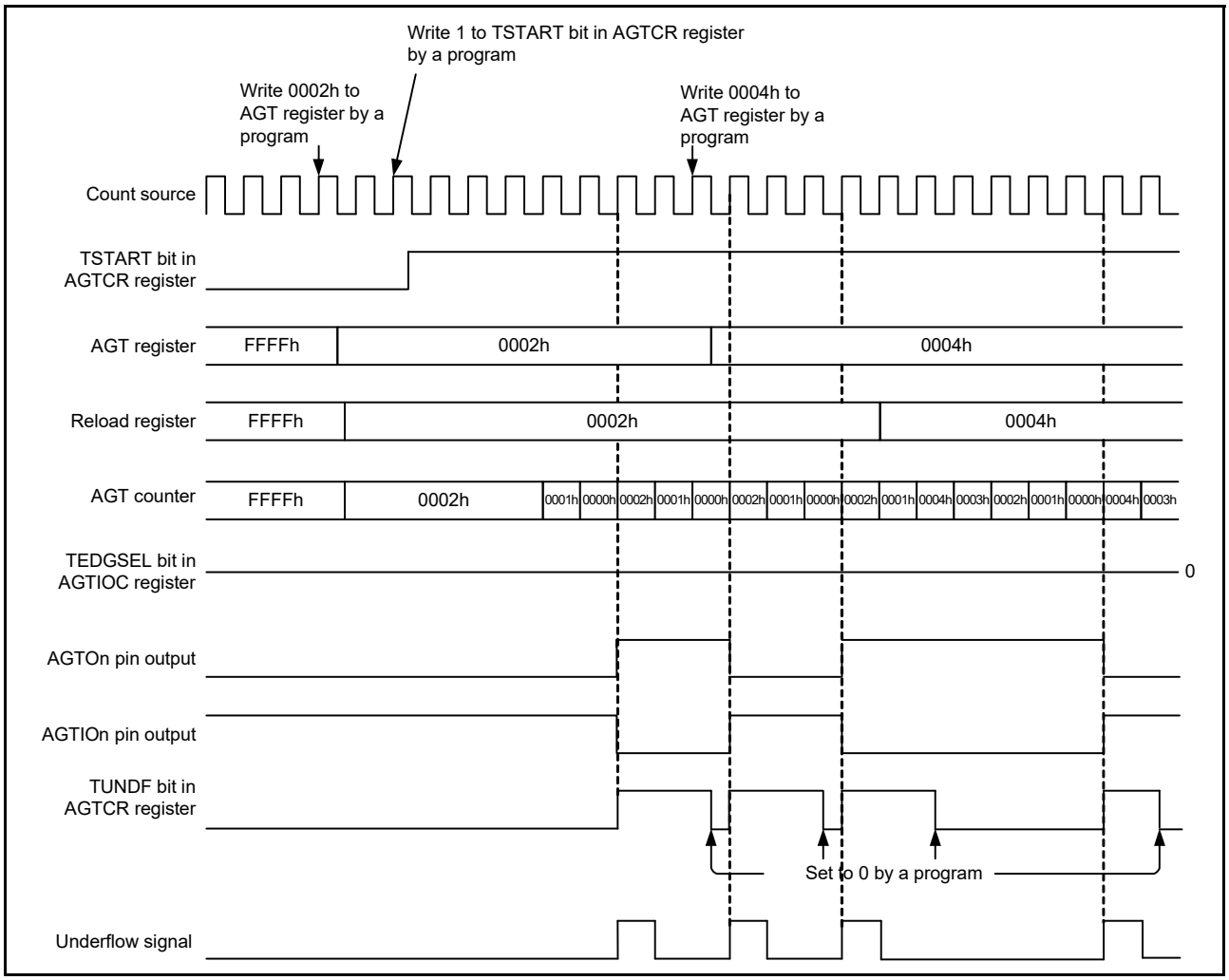


Figure 24.6 Operation example in pulse output mode

24.3.5 Event Counter Mode

In this mode, the counter is decremented by an external event signal input to the AGTIOn pin. Various periods for counting events can be set with the TIOGT[1:0] bits in the AGTIOC and AGTISR registers. In addition, the filter function for the AGTIOn input can be specified with the TIPF[1:0] bits in the AGTIOC register. The output from the AGTOn pin can be toggled even in event counter mode.

Figure 24.7 shows the operation example in event counter mode.

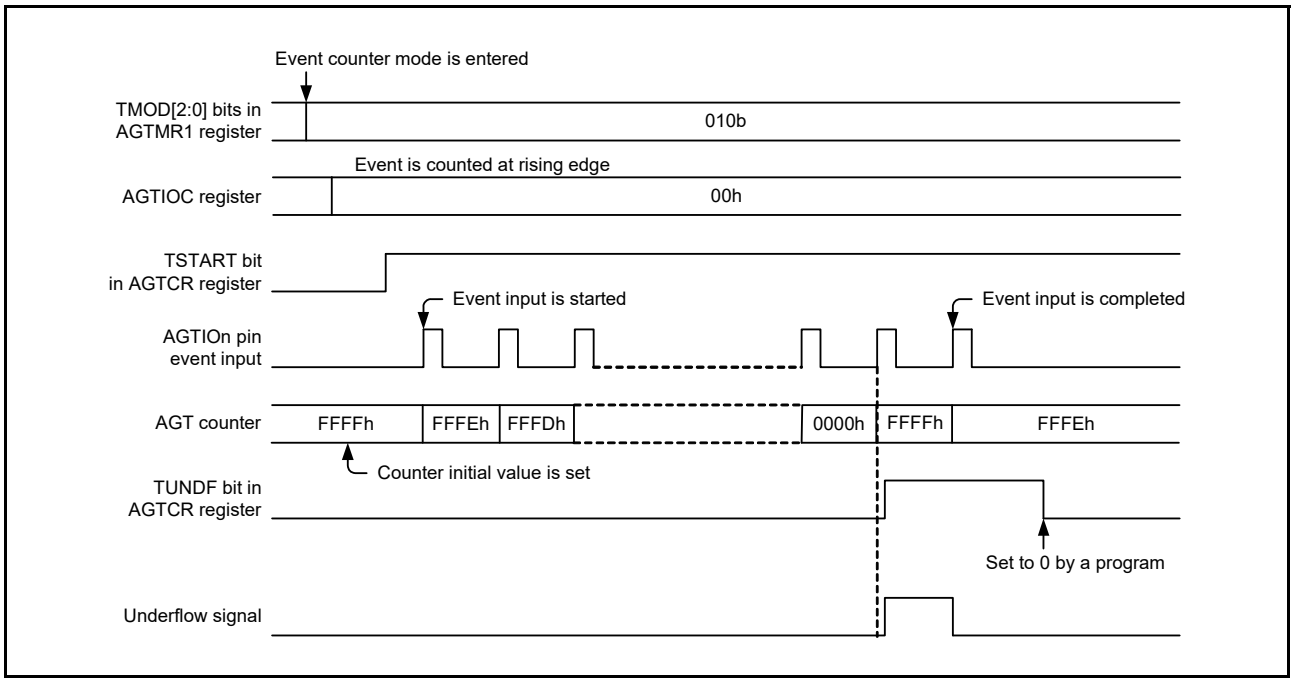
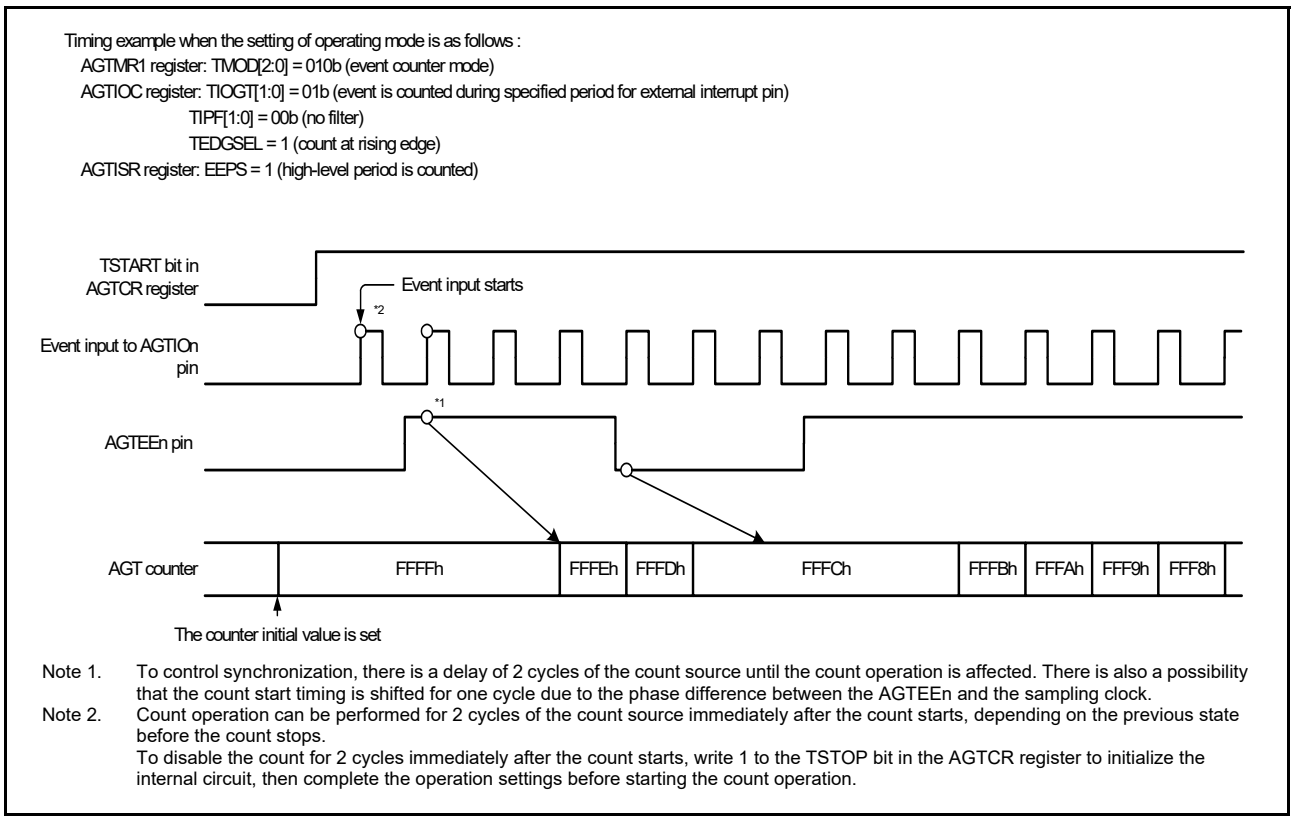


Figure 24.7 Operation example 1 in event counter mode

Figure 24.8 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. There is also a possibility that the count start timing is shifted for one cycle due to the phase difference between the AGTEEn 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, then complete the operation settings before starting the count operation.

Figure 24.8 Operation example 2 in event counter mode

24.3.6 Pulse Width Measurement Mode

In this 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 by TCK[2:0] bits in the AGTMR1 register. When the specified level on the AGTIOn pin ends, the counter is stopped, the TEDGF bit 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 bit in the AGTCR register is set to 1 and an interrupt request is generated.

Figure 24.9 shows the operation example in pulse width measurement mode.

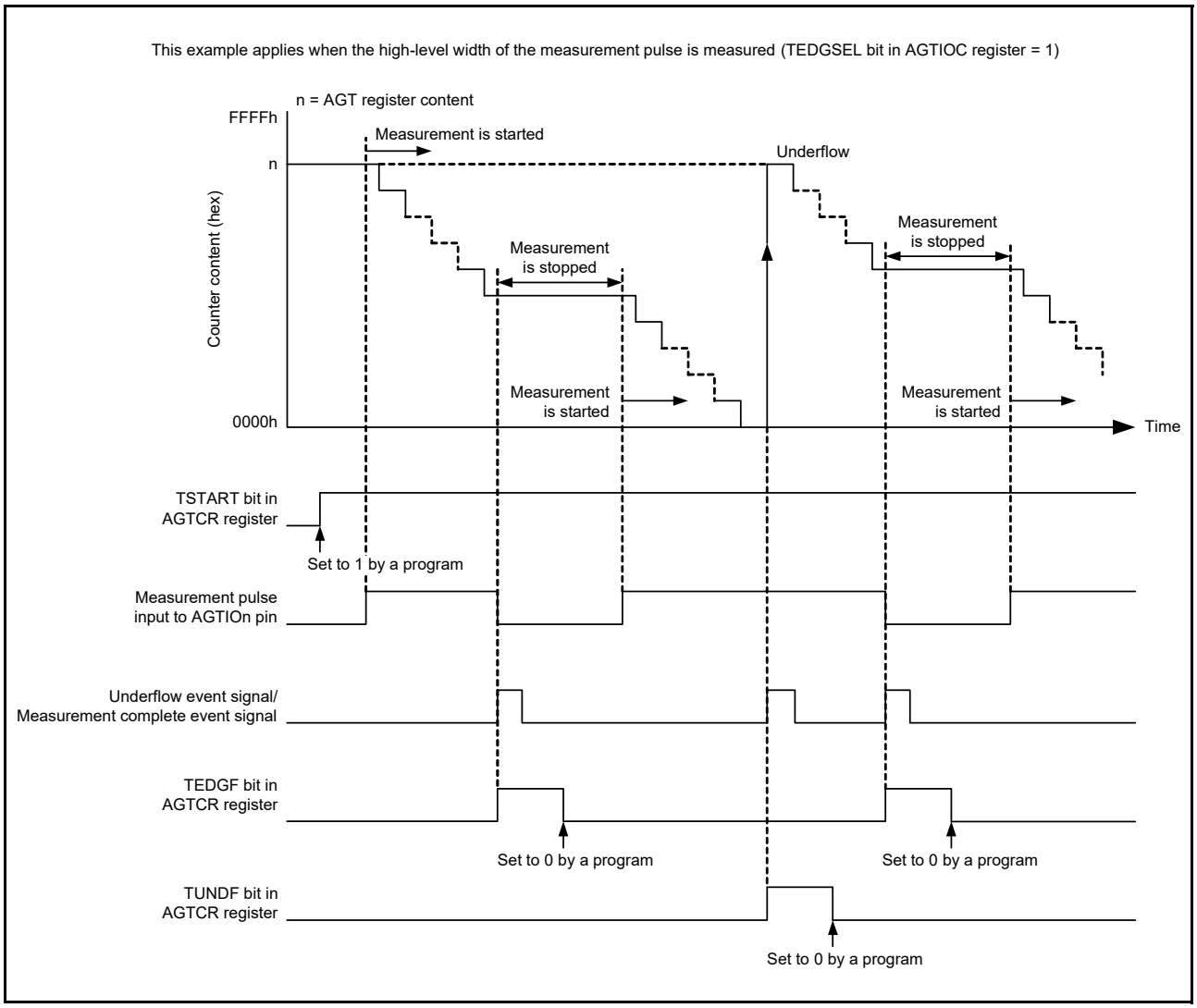


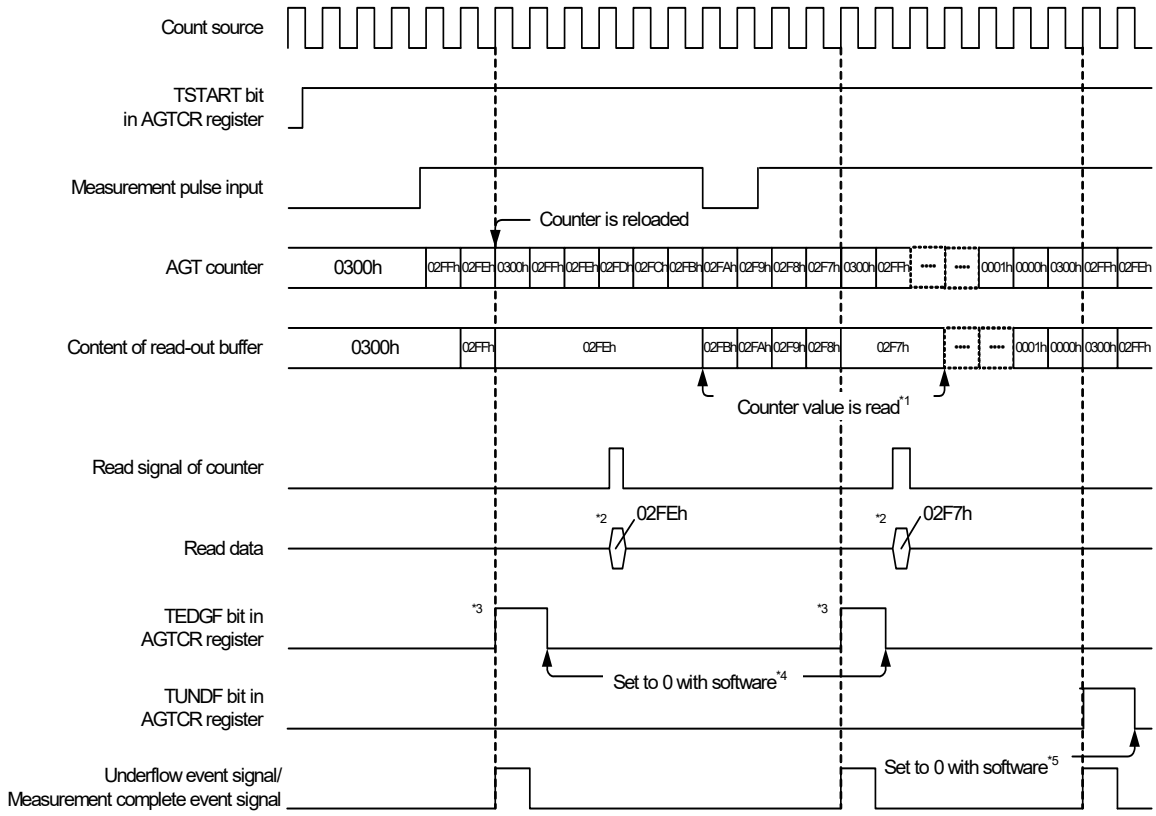
Figure 24.9 Operation example in pulse width measurement mode

24.3.7 Pulse Period Measurement Mode

In this mode, the pulse period of an external signal input to the AGTIOn pin is measured. The counter is decremented by the count source selected by 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 bit 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 24.4.5, 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 bit in the AGTCR register is set to 1 and an interrupt request is generated.

Figure 24.10 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.



Note 1. This example applies when the initial value of the AGT register is set to 0300h, 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 2. Reading from the AGT register must be performed during the period from when the TEDGF bit 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 3. When the AGT register is read in pulse period measurement mode, the content of the read-out buffer is read.
Note 4. When the active edge of the measurement pulse and the set edge of an external pulse are input, the TEDGF bit in the AGTCR register is set to 1 (active edge received).
Note 5. To set to 0 with software, write 0 to the TEDGF bit in the AGTCR register using an 8-bit memory manipulation instruction.
Note 6. To set to 0 with software, write 0 to the TUNDF bit in the AGTCR register using an 8-bit memory manipulation instruction.

Figure 24.10 Operation example in pulse period measurement mode

24.3.8 Compare Match function

This 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 bit or the 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 bit in the AGTCR register is set to 1 (match), and an interrupt request is generated.

When compare match function is enabled, the timing of the rewrite operation to the reload register and the counter differs. See section 24.3.1, 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 24.11 shows the operation example in compare match mode.

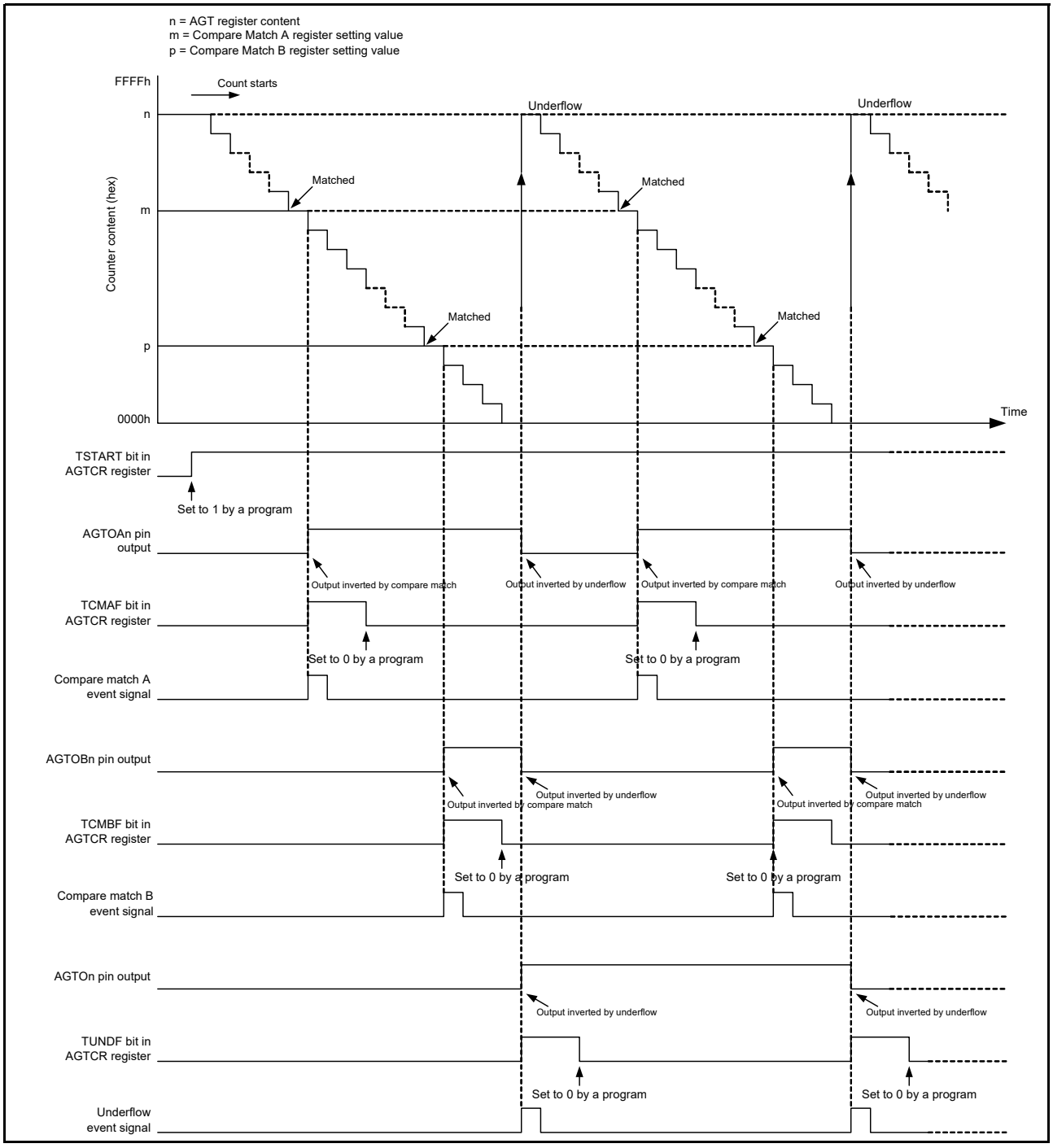


Figure 24.11 Operation example in compare match mode (TOPOLA = 0, TOPOLB = 0)

24.3.9 Output Settings for each Mode

Table 24.5 to Table 24.8 list the states of pins AGTOn, AGTIOn, AGTOAn, and AGTOBn in each mode.

Table 24.5 AGTOn pin setting

	AGTIOC Register		
Operating mode	**TOE bit**	**TEDGSEL bit**	**AGTOn pin output**
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

Table 24.6 AGTIOn pin setting

	AGTIOC Register	
Operating mode	**TEDGSEL bit**	**AGTIOn pin I/O**
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 24.7 AGTOAn pin setting

	AGTCMSR Register		
Operating mode	**TOEA bit**	**TOPOLA bit**	**AGTOAn pin output**
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 24.8 AGTOBn pin setting (1 of 2)

	AGTCMSR Register		
Operating mode	**TOEB bit**	**TOPOLB bit**	**AGTOBn pin output**
Timer mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (Not used)

Table 24.8 AGTOBn pin setting (2 of 2)

	AGTCMSR Register		
Operating mode	**TOEB bit**	**TOPOLB bit**	**AGTOBn pin output**
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			

24.3.10 Standby Mode

The AGT can operate in Software Standby mode. Set it to Software Standby mode with count operation start (TSTART = 1, TCSTF = 1).

Table 24.9 and Table 24.10 show the setting that can be used in Software Standby mode.

Table 24.9 Usable setting in Software Standby mode (AGT0)

Operating mode	**TCK[2:0] bits of AGTMR1 Register**	**Operating clock**	**Resurgence factor of CPU**
Timer mode	100b or 110b	AGTLCLK or AGTSCLK	-
Pulse output mode	100b or 110b	AGTLCLK or AGTSCLK	-
Event counter mode	- (invalid)	AGTIOn	-
Pulse width measurement mode	100b or 110b	AGTLCLK or AGTSCLK	-
Pulse period measurement mode	100b or 110b	AGTLCLK or AGTSCLK	-

Table 24.10 Usable setting in Software Standby mode (AGT1)

Operating mode	**TCK[2:0] bits of AGTMR1 Register**	**Operating clock**	**Resurgence factor of CPU**
Timer mode	100b or 110b or 101b *1	AGTLCLK or AGTSCLK or AGT0 underflow	• Underflow • Compare Match A/B
Pulse output mode	100b or 110b or 101b *1	AGTLCLK or AGTSCLK or AGT0 underflow	• Underflow • Compare Match A/B
Event counter mode	- (invalid)	AGTIOn	• Underflow • Compare Match A/B
Pulse width measurement mode	100b or 110b or 101b *1	AGTLCLK or AGTSCLK or AGT0 underflow	• Underflow • Active edge
Pulse period measurement mode	100b or 110b or 101b *1	AGTLCLK or AGTSCLK or AGT0 underflow	• Underflow • Active edge

Note: Release of Software Standby mode is only AGT1.
Note 1. Only when AGT0 operates in Table 24.9.

24.3.11 Interrupt Sources

The AGT has three interrupt sources described in Table 24.11.

Table 24.11 AGT interrupt sources

Name	Interrupt source	DMAC/DTC activation
AGTn_AGTI	• When the counter underflows • When measurement of the active width of the external input (AGTIOn) is complete in pulse width measurement mode • When the set edge of the external input (AGTIOn) is input in pulse period measurement mode.	Possible
AGTn_AGTCMAI	When the values of AGT and AGTCMA match	Possible
AGTn_AGTCMBI	When the values of AGT and AGTCMB match	Possible

Note: Channel number (n = 0, 1)

24.3.12 Event Signal Output to ELC

The AGT 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 AGT outputs Compare Match A, Compare Match B, and underflow/measurement complete signals as event signals. For details, see section 19, Event Link Controller (ELC).

24.4 Usage Notes

24.4.1 Count Operation Start and Stop Control

- When the operating mode (see Table 24.1) is set to other than the event counter mode, or the count source is set to other than AGT0 underflow (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 bit in the AGTCR register remains 0 (count stops) for 3 cycles of the count source. Do not access the registers associated with AGT*1 other than the TCSTF bit until this bit is set to 1 (count in progress).
  - After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF bit remains 1 for 3 cycles of the count source. When the TCSTF bit is set to 0, the count is stopped. Do not access the registers associated with AGT*1 other than the TCSTF bit until this bit is set to 0.
  - Clear the interrupt register before changing the TSTART bit from 0 to 1. See section 14, Interrupt Controller Unit (ICU) for details.

Note 1. Registers associated with AGT: AGT, AGTCMA, AGTCMB, AGTCR, AGTMR1, AGTMR2, AGTIOC, AGTISR and AGTCMSR

- When the operating mode (see Table 24.1) is set to event counter mode, or the count source is set to AGT0 underflow (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 bit in the AGTCR register remains 0 (count stops) for 2 cycles of the PCLKB. Do not access the registers associated with AGT*1 other than the TCSTF bit until this bit is set to 1 (count in progress).
  - After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF bit remains 1 for 2 cycles of the PCLKB. When the TCSTF bit is set to 0, the count is stopped. Do not access the registers associated with AGT*1 other than the TCSTF bit until this bit is set to 0.
  - Clear the interrupt register before changing the TSTART bit from 0 to 1. See section 14, Interrupt Controller Unit (ICU) for details.

Note 1. Registers associated with AGT: AGT, AGTCMA, AGTCMB, AGTCR, AGTMR1, AGTMR2, AGTIOC, AGTISR and AGTCMSR

24.4.2 Access to Counter Register

When bits TSTART and TCSTF 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.

24.4.3 When Changing Mode

The registers associated with AGT operating mode (AGTMR1, AGTMR2, AGTIOC, AGTISR, AGTCMSR, and AGTIOC) can be changed only when the count is stopped with both the TSTART and TCSTF bits 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 bits are undefined. Before starting the count, write 0 to the following bits:

- TEDGF (no active edge received)
- TUNDF (no underflow)
- TCMAF (no match)
- TCMBF (no match).

24.4.4 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.

24.4.5 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

24.4.6 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 one cycle of the count source:

- AGT
- AGTCMA
- AGTCMB
- AGTCR
- AGTMR1
- AGTMR2.

24.4.7 When Selecting AGT0 Underflow as the Count Source

Operate the AGT according to the procedures described in this section when selecting the underflow signal of AGT as the count source.

(1) Procedure for starting operation

1. Set AGT0 and AGT1.
2. Start the count operation of AGT1.
3. Start the count operation of AGT0.

(2) Procedure for stopping operation

1. Stop the count operation of AGT0.
2. Stop the count operation of AGT1.
3. Stop the count source clock of AGT1 (write 000b in AGT1.AGTMR1.TCK[2:0] bits).

24.4.8 Reset of I/O Register

The I/O register of the AGT is not initialized by different types of resets. For details, see section 6, Resets.

24.4.9 When Selecting PCLKB/8 as the Count Source

When a reset is generated, the operation of AGT cannot be guaranteed. Set the registers associated with AGT again.

24.4.10 When Selecting AGTLCLK or AGTSCLK as the Count Source

The MSTPD2 in MSTPCRD register must be set to 1 except when accessing the AGT1 registers. The MSTPD3 in MSTPCRD register must be set to 1 except when accessing the AGT0 registers. When a reset occurs while MSTPD2 or MSTPD3 is 0, the operation of AGT1 or AGT0 cannot be guaranteed. Set the registers associated with AGT again.

24.4.11 When Count Source Clock Frequency is over 32 kHz

1. AGT1 can be used but not AGT0. MSTPCRD.MSTPD3 bit must be set to 1. The count source of AGT0 is available for an underflow event signal.
2. While AGT operates, the VBATT backup register cannot be accessed. When using the VBATT back register, you must stop the AGT operation. The MSTPCRD.MSTPD3 and MSTPCRD.MSTPD2 bits must be set to 1.
3. For event counter mode with input frequency (t_{ACYC}), see section 52, Electrical Characteristics.

25. Realtime Clock (RTC)

25.1 Overview

The RTC has two counting modes, calendar count mode and binary count mode, that are used by switching register settings. For calendar count mode, the RTC has a 100 year calendar from 2000 to 2099 and automatically adjusts dates for leap years. For binary count mode, the RTC counts seconds and retains the information as a serial value. Binary count mode can be used for calendars other than the Gregorian (Western) calendar.

The sub-clock or LOCO can be selected as the count source of the time counters. The RTC uses a 128-Hz clock acquired by dividing the count source by a prescaler. Year, month, date, day-of-week, a.m./p.m. (in 12-hour mode), hour, minute, second, or 32-bit binary is counted by 1/128 second.

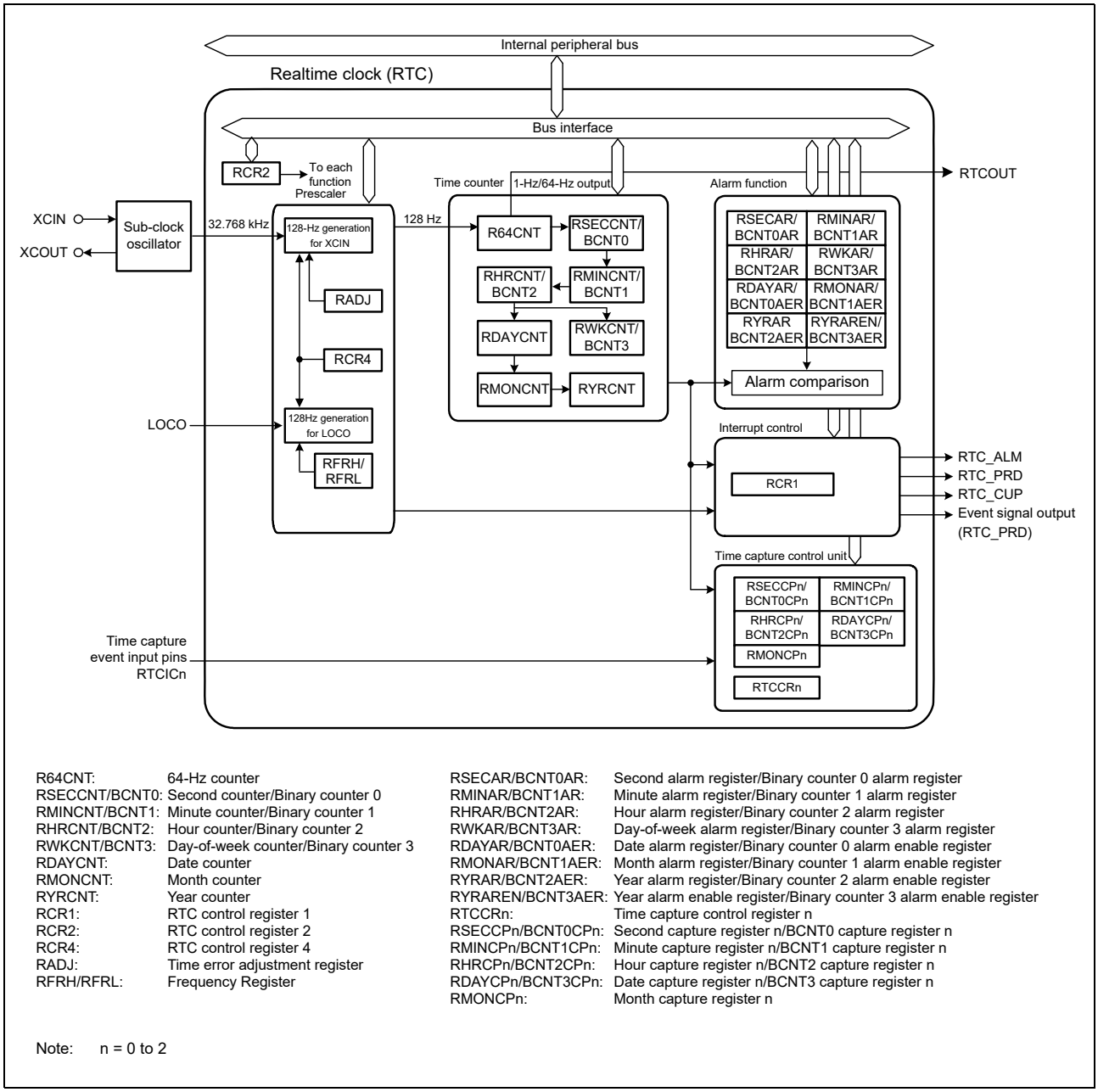
Note: Regardless of the use of VBATT function, set the VBTCR1.BPWSWSTP bit to 1 before the accessing to RTC registers after cold start. For details, see Figure 12.2, Setting flow of the VBTCR1.BPWSWSTP bit, in section 12, Battery Backup Function.

Table 25.1 lists the specifications of the RTC, Figure 25.1 shows a block diagram of the RTC, and Table 25.2 shows the pin configuration of the RTC.

Table 25.1 RTC specifications

Parameter	Description
Count mode	Calendar count mode/binary count mode
Count source*1	Sub-clock (XCIN) or LOCO
Clock and calendar functions	• Calendar count mode Year, month, date, day of week, hour, minute, second are counted, BCD display 12 hours/24 hours mode switching function 30 seconds adjustment function (a number less than 30 is rounded down to 00 seconds, and 30 seconds or more are rounded up to 1 minute) Automatic adjustment function for leap years • Binary count mode Count seconds in 32 bits, binary display • Common to both modes Start/stop function The sub-second digit is displayed in binary units (1 Hz, 2 Hz, 4 Hz, 8 Hz, 16 Hz, 32 Hz, or 64 Hz). Clock error correction function Clock (1-Hz/64-Hz) output
Interrupts	• Alarm interrupt (RTC_ALM) As an alarm interrupt condition, selectable for comparison with the following: Calendar count mode: Year, month, date, day-of-week, hour, minute, or second can be selected Binary count mode: Each bit of the 32-bit binary counter • Periodic interrupt (RTC_PRD) 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second can be selected as an interrupt period. • Carry interrupt (RTC_CUP) An interrupt is generated at either of the following conditions: - When a carry from the 64-Hz counter to the second counter is generated. - When the 64-Hz counter is changed and the R64CNT register is read at the same time. • Recovery from Software Standby mode can be performed by an alarm interrupt or periodic interrupt
Time capture function	• Times can be captured when the edge of the time capture event input pin is detected. For every event input, month, date, hour, minute, and second are captured or the 32-bit binary counter value is captured.
Event link function	Periodic event output (RTC_PRD)

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source clock.



R64CNT: 64-Hz counter
RSECCNT/BCNT0: Second counter/Binary counter 0
RMINCNT/BCNT1: Minute counter/Binary counter 1
RHRCNT/BCNT2: Hour counter/Binary counter 2
RWKCNT/BCNT3: Day-of-week counter/Binary counter 3
RDAYCNT: Date counter
RMONCNT: Month counter
RYRCNT: Year counter
RCR1: RTC control register 1
RCR2: RTC control register 2
RCR4: RTC control register 4
RADJ: Time error adjustment register
RFRH/RFRL: Frequency Register

RSECAR/BCNT0AR: Second alarm register/Binary counter 0 alarm register
RMINAR/BCNT1AR: Minute alarm register/Binary counter 1 alarm register
RHRAR/BCNT2AR: Hour alarm register/Binary counter 2 alarm register
RWKAR/BCNT3AR: Day-of-week alarm register/Binary counter 3 alarm register
RDAYAR/BCNT0AER: Date alarm register/Binary counter 0 alarm enable register
RMONAR/BCNT1AER: Month alarm register/Binary counter 1 alarm enable register
RYRAR/BCNT2AER: Year alarm register/Binary counter 2 alarm enable register
RYRAREN/BCNT3AER: Year alarm enable register/Binary counter 3 alarm enable register
RTCCRn: Time capture control register n
RSECCPn/BCNT0CPn: Second capture register n/BCNT0 capture register n
RMINCPn/BCNT1CPn: Minute capture register n/BCNT1 capture register n
RHRCPn/BCNT2CPn: Hour capture register n/BCNT2 capture register n
RDAYCPn/BCNT3CPn: Date capture register n/BCNT3 capture register n
RMONCPn: Month capture register n

Note: n = 0 to 2

Figure 25.1 RTC block diagram

Table 25.2 Pin configuration of RTC

Pin name	I/O	Function
XCIN	Input	Connect a 32.768-kHz crystal to these pins.
XCOUT	Output	
RTCOUT	Output	This pin is used to output a 1-Hz/64-Hz waveform.
RTCIC0	Input	Time capture event input pins
RTCIC1	Input	
RTCIC2	Input	

25.2 Register Descriptions

Write or read from the RTC registers in accordance with section 25.6.5, Notes When Writing to and Reading from Registers.

If the value in an RTC register after a reset is given as x (undefined bits) in the list, it is not initialized by a reset. When RTC enters the reset state or a low power consumption state during counting operations, for example while the RCR2.START bit is 1, the year, month, day of the week, date, hours, minutes, seconds, and 64-Hz counters continue to operate.

Note: A reset generated while writing to a register might destroy the register value. In addition, do not allow the chip to enter Software Standby mode immediately after setting any of these registers. For details, see section 25.6.4, Transitions to Low Power Consumption Modes after Setting Registers.

25.2.1 64-Hz Counter

Address(es): RTC.R64CNT 4004 4000h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	F1HZ	F2HZ	F4HZ	F8HZ	F16HZ	F32HZ	F64HZ
Value after reset:	0	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	F64HZ	64 Hz	Indicate the state between 1 Hz and 64 Hz of the sub-second digit.	R
b1	F32HZ	32 Hz		R
b2	F16HZ	16 Hz		R
b3	F8HZ	8 Hz		R
b4	F4HZ	4 Hz		R
b5	F2HZ	2 Hz		R
b6	F1HZ	1 Hz		R
b7	—	Reserved	This bit is read as 0.	R

The R64CNT counter is used in both calendar count mode and in binary count mode. The 64-Hz counter (R64CNT) generates the period for a second by counting up periods of the 128-Hz clock. The state in the sub-second range can be confirmed by reading this counter.

This counter is set to 00h by an RTC software reset or an execution of a 30-second adjustment. To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

25.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0)

(1) In calendar count mode:

Address(es): RTC.RSECCNT 4004 4002h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SEC10[2:0]			SEC1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	SEC1[3:0]	1-Second Count	Counts from 0 to 9 every second. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	SEC10[2:0]	10-Second Count	Counts from 0 to 5 for 60-second counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

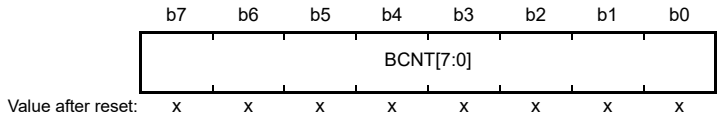
The RSECCNT counter sets and counts the BCD-coded second value. It counts carries generated once per second in the 64-Hz counter.

The setting range is decimal 00 to 59. The RTC does not operate normally if any other value is set. Before writing to this register, be sure to stop the count operation using the START bit in RCR2.

To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode:

Address(es): RTC.BCNT0 4004 4002h



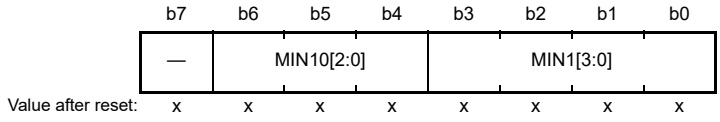
x: Undefined

BCNT0 is a read/write 32-bit binary counter b7 to b0. The 32-bit binary counter performs count operation by a carry generated for each second of the 64-Hz counter. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

25.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1)

(1) In calendar count mode:

Address(es): RTC.RMINCNT 4004 4004h



x: Undefined

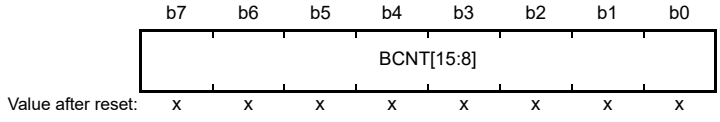
Bit	Symbol	Bit name	Description	R/W
b3 to b0	MIN1[3:0]	1-Minute Count	Counts from 0 to 9 every minute. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	MIN10[2:0]	10-Minute Count	Counts from 0 to 5 for 60-minute counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RMINCNT counter sets and counts the BCD-coded minute value. It counts carries generated once per minute in the second counter.

A value from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode:

Address(es): RTC.BCNT1 4004 4004h



x: Undefined

The BCNT1 counter is a readable/writable 32-bit binary counter b15 to b8. The 32-bit binary counter performs count operation by a carry generated for each second of the 64-Hz counter. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

25.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2)

(1) In calendar count mode:

Address(es): RTC.RHRCNT 4004 4006h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	HR1[3:0]	1-Hour Count	Counts from 0 to 9 once every hour. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	HR10[1:0]	10-Hour Count	Counts from 0 to 2 once per carry from the ones place.	R/W
b6	PM	PM	Time counter setting for a.m./p.m. 0: a.m. 1: p.m.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RHRCNT counter sets and counts the BCD-coded hour value. It counts carries generated once every hour in the minute counter. The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24):

- When the RCR2.HR24 bit is 0 – from 00 to 11 (in BCD)
- When the RCR2.HR24 bit is 1 – from 00 to 23 (in BCD).

If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. The PM bit is only enabled when the RCR2.HR24 bit is 0. Otherwise, the setting in the PM bit has no effect. To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode:

Address(es): RTC.BCNT2 4004 4006h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT2 counter is a read/write 32-bit binary counter b23 to b16. The 32-bit binary counter performs count operation by a carry generated for each second of the 64-Hz counter. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

25.2.5 Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3)

(1) In calendar count mode:

Address(es): RTC.RWKCNT 4004 4008h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b2 to b0	DAYW[2:0]	Day-of-Week Counting	b2 b0 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting prohibited.	R/W
b7 to b3	—	Reserved	Set these bits to 0. They are read as the set value.	R/W

The RWKCNT counter sets and counts in the coded day-of-week value. It counts carries generated once per day in the hour counter. A value from 0 through 6 can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

(2) In binary count mode:

Address(es): RTC.BCNT3 4004 4008h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT3 is a read/write 32-bit binary counter b31 to b24 that performs count operation by a carry generated for each second of the 64-Hz counter. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

25.2.6 Day Counter (RDAYCNT)

Address(es): RTC.RDAYCNT 4004 400Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	0	0	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	DATE1[3:0]	1-Day Count	Counts from 0 to 9 once per day. When a carry is generated, 1 is added to the tens place	R/W
b5, b4	DATE10[1:0]	10-Day Count	Counts from 0 to 3 once per carry from the ones place	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RDAYCNT counter is used in calendar count mode to set and count the BCD-coded date value. It counts carries generated once per day in the hour counter. The count operation depends on the month and whether the year is a leap year. Leap years are determined according to whether the year counter (RYRCNT) value is divisible by 400, 100, and 4.

A value from 01 through 31 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. When specifying a value, the range of specifiable days depends on the month and whether the year is a leap year. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

25.2.7 Month Counter (RMONCNT)

Address(es): RTC.RMONCNT 4004 400Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MON10	MON1[3:0]			
Value after reset:	0	0	0	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	MON1[3:0]	1-Month Count	Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place	R/W
b4	MON10	10-Month Count	Counts from 0 to 1 once per carry from the ones place	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RMONCNT counter is used in calendar count mode to set and count the BCD-coded month value. It counts carries generated once per month in the date counter.

A value from 01 through 12 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

25.2.8 Year Counter (RYRCNT)

Address(es): RTC.RYRCNT 4004 400Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	YR1[3:0]	1-Year Count	Counts from 0 to 9 once per year. When a carry is generated, 1 is added to the tens place.	R/W
b7 to b4	YR10[3:0]	10-Year Count	Counts from 0 to 9 once per carry from ones place. When a carry is generated in the tens place, 1 is added to the hundreds place.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

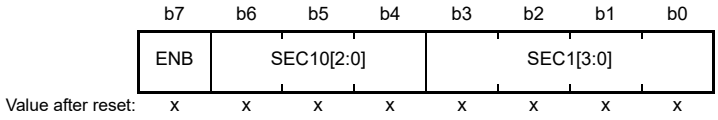
The RYRCNT counter is used in calendar count mode to set and count the BCD-coded year value. It counts carries generated once per year in the month counter.

A value from 00 through 99 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time.

25.2.9 Second Alarm Register (RSECAR)/Binary Counter 0 Alarm Register (BCNT0AR)

(1) In calendar count mode:

Address(es): RTC.RSECAR 4004 4010h



x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	SEC1[3:0]	1 Second	Value for the ones place of seconds	R/W
b6 to b4	SEC10[2:0]	10 Seconds	Value for the tens place of seconds	R/W
b7	ENB	ENB	0: The register value is not compared with the RSECCNT counter value 1: The register value is compared with the RSECCNT counter value.	R/W

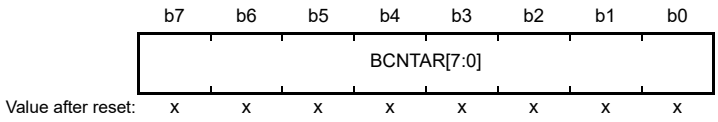
RSECAR is an alarm register associated with the BCD-coded second counter RSECCNT. When the ENB bit is set to 1, the RSECAR value is compared with the RSECCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN.

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. The RSECAR values from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT0AR 4004 4010h



x: Undefined

BCNT0AR is a read/write alarm register associated with the 32-bit binary counter b7 to b0. This register is set to 00h by an RTC software reset.

25.2.10 Minute Alarm Register (RMINAR)/Binary Counter 1 Alarm Register (BCNT1AR)

(1) In calendar count mode:

Address(es): RTC.RMINAR 4004 4012h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	MIN10[2:0]			MIN1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	MIN1[3:0]	1 Minute	Value for the ones place of minutes	R/W
b6 to b4	MIN10[2:0]	10 Minutes	Value for the tens place of minutes	R/W
b7	ENB	ENB	0: The register value is not compared with the RMINCNT counter value 1: The register value is compared with the RMINCNT counter value.	R/W

RMINAR is an alarm register associated with the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN.

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. The RMINAR values from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT1AR 4004 4012h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT1AR is a read/write alarm register associated with the 32-bit binary counter from b15 to b8. This register is set to 00h by an RTC software reset.

25.2.11 Hour Alarm Register (RHRAR)/Binary Counter 2 Alarm Register (BCNT2AR)

(1) In calendar count mode:

Address(es): RTC.RHRAR 4004 4014h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	HR1[3:0]	1 Hour	Value for the ones place of hours	R/W
b5, b4	HR10[1:0]	10 Hours	Value for the tens place of hours	R/W
b6	PM	PM	Time Alarm Setting for a.m./p.m. 0: a.m. 1: p.m.	R/W
b7	ENB	ENB	0: The register value is not compared with the RHRCNT counter value 1: The register value is compared with the RHRCNT counter value.	R/W

RHRAR is an alarm register associated with the BCD-coded hour counter RHRCNT. When the ENB bit is set to 1, the RHRAR value is compared with the RHRCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN.

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24):

- When the RCR2.HR24 bit is 0 – From 00 to 11 (in BCD)
- When the RCR2.HR24 bit is 1 – From 00 to 23 (in BCD).

If a value outside of this range is specified, the RTC does not operate correctly. When the RCR2.HR24 bit is 0, be sure to set the PM bit. When the RCR2.HR24 bit is 1, the setting in the PM bit has no effect. This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT2AR 4004 4014h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT2AR is a read/write alarm register associated with the 32-bit binary counter b23 to b16. This register is set to 00h by an RTC software reset.

25.2.12 Day-of-Week Alarm Register (RWKAR)/Binary Counter 3 Alarm Register (BCNT3AR)

(1) In calendar count mode:

Address(es): RTC.RWKAR 4004 4016h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b2 to b0	DAYW[2:0]	Day-of-Week Setting	b2 b0 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting prohibited.	R/W
b6 to b3	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RWKCNT counter value 1: The register value is compared with the RWKCNT counter value.	R/W

RWKAR is an alarm register associated with the coded day-of-week counter RWKCNT. When the ENB bit is set to 1, the RWKAR value is compared with the RWKCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the corresponding counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN.

When all the respective values all match, the IR flag associated with the RTC_ALM interrupt is set to 1. The RWKAR values from 0 through 6 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT3AR 4004 4016h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT3AR is a read/write alarm register associated with the 32-bit binary counter b31 to b24. This register is set to 00h by an RTC software reset.

25.2.13 Date Alarm Register (RDAYAR)/Binary Counter 0 Alarm Enable Register (BCNT0AER)

(1) In calendar count mode:

Address(es): RTC.RDAYAR 4004 4018h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	DATE1[3:0]	1 Day	Value for the ones place of days	R/W
b5, b4	DATE10[1:0]	10 Days	Value for the tens place of days	R/W
b6	—	Reserved	Set this bit to 0. It is read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RDAYCNT counter value 1: The register value is compared with the RDAYCNT counter value.	R/W

RDAYAR is an alarm register associated with the BCD-coded date counter RDAYCNT. When the ENB bit is set to 1, the RDAYAR value is compared with the RDAYCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the corresponding counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN.

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. The RDAYAR values from 01 through 31 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT0AER 4004 4018h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT0AER is a read/write register to set the alarm enable associated with the 32-bit binary counter b7 to b0. The binary counter (BCNT[31:0]) associated with the ENB[31:0] bits that are set to 1 is compared with the binary alarm register (BCNTAR[31:0]), and when all match, the IR flag associated with the RTC_ALM interrupt becomes 1. This register is set to 00h by an RTC software reset.

25.2.14 Month Alarm Register (RMONAR)/Binary Counter 1 Alarm Enable Register (BCNT1AER)

(1) In calendar count mode:

Address(es): RTC.RMONAR 4004 401Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	MON10	MON1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	MON1[3:0]	1 Month	Value for the ones place of months	R/W
b4	MON10	10 Months	Value for the tens place of months	R/W
b6, b5	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RMONCNT counter value 1: The register value is compared with the RMONCNT counter value.	R/W

RMONAR is an alarm register associated with the BCD-coded month counter RMONCNT. When the ENB bit is set to 1, the RMONAR value is compared with the RMONCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN.

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1. The RMONAR values from 01 through 12 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT1AER 4004 401Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT1AER is a read/write register for setting the alarm enable associated with the 32-bit binary counter b15 to b8. The binary counter (BCNT[31:0]) associated with the ENB[31:0] bits that are set to 1 is compared with the binary alarm register (BCNTAR[31:0]), and when all match, the IR flag associated with the RTC_ALM interrupt becomes 1. This register is set to 00h by an RTC software reset.

25.2.15 Year Alarm Register (RYRAR)/Binary Counter 2 Alarm Enable Register (BCNT2AER)

(1) In calendar count mode:

Address(es): RTC.RYRAR 4004 401Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	YR1[3:0]	1 Year	Value for the ones place of years	R/W
b7 to b4	YR10[3:0]	10 Years	Value for the tens place of years	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RYRAR is an alarm register associated with the BCD-coded year counter RYRCNT. The RYRAR values from 00 through 99 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0000h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT2AER 4004 401Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ENB[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

BCNT2AER is a read/write register for setting the alarm enable associated with the 32-bit binary counter b23 to b16. The binary counter (BCNT[31:0]) associated with the ENB[31:0] bits that are set to 1 is compared with the binary alarm register (BCNTAR[31:0]), and when all match, the IR flag associated with the RTC_ALM interrupt becomes 1. This register is set to 0000h by an RTC software reset.

25.2.16 Year Alarm Enable Register (RYRAREN)/Binary Counter 3 Alarm Enable Register (BCNT3AER)

(1) In calendar count mode:

Address(es): RTC.RYRAREN 4004 401Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b6 to b0	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RYRCNT counter value 1: The register value is compared with the RYRCNT counter value.	R/W

When the ENB bit in RYRAREN is set to 1, the RYRAR value is compared with the RYRCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN.

When all the respective values match, the IR flag associated with the RTC_ALM interrupt is set to 1.This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT3AER 4004 401Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT3AER is a read/write register for setting the alarm enable associated with the 32-bit binary counter b31 to b24. The binary counter (BCNT[31:0]) associated with the ENB[31:0] bits that are set to 1 is compared with the binary alarm register (BCNTAR[31:0]), and when all match, the IR flag associated with the RTC_ALM interrupt becomes 1. This register is set to 00h by an RTC software reset.

25.2.17 RTC Control Register 1 (RCR1)

Address(es): RTC.RCR1 4004 4022h

	b7	b6	b5	b4	b3	b2	b1	b0
	PES[3:0]				RTCOS	PIE	CIE	AIE
Value after reset:	x	x	x	x	0	x	0	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	AIE	Alarm Interrupt Enable	0: An alarm interrupt request is disabled 1: An alarm interrupt request is enabled.	R/W
b1	CIE	Carry Interrupt Enable	0: A carry interrupt request is disabled 1: A carry interrupt request is enabled.	R/W
b2	PIE	Periodic Interrupt Enable	0: A periodic interrupt request is disabled 1: A periodic interrupt request is enabled.	R/W
b3	RTCOS	RTCOUT Output Select	0: RTCOUT outputs 1 Hz 1: RTCOUT outputs 64 Hz.	R/W
b7 to b4	PES[3:0]	Periodic Interrupt Select	b7 b6 b5 b4 0 1 1 0: A periodic interrupt is generated every 1/256 second*1 0 1 1 1: A periodic interrupt is generated every 1/128 second 1 0 0 0: A periodic interrupt is generated every 1/64 second 1 0 0 1: A periodic interrupt is generated every 1/32 second 1 0 1 0: A periodic interrupt is generated every 1/16 second 1 0 1 1: A periodic interrupt is generated every 1/8 second 1 1 0 0: A periodic interrupt is generated every 1/4 second 1 1 0 1: A periodic interrupt is generated every 1/2 second 1 1 1 0: A periodic interrupt is generated every 1 second 1 1 1 1: A periodic interrupt is generated every 2 seconds. Other than above: No periodic interrupts are generated.	R/W

Note 1. When LOCO is selected (RCR4.RCKSEL = 1) while PES[3:0] = 0110b, a periodic interrupt is generated every 1/128 second.

The RCR1 register is used in both calendar count mode and in binary count mode. Bits AIE, PIE, and PES[3:0] are updated synchronously with the count source. When the RCR1 register is modified, check that all the bits are updated before proceeding.

AIE bit (Alarm Interrupt Enable)

This bit enables or disables alarm interrupt requests.

CIE bit (Carry Interrupt Enable)

This bit enables and disables interrupt requests when a carry to the RSECCNT/BCNT0 register occurs, or when a carry to the 64-Hz counter (R64CNT) occurs while reading the 64-Hz counter.

PIE bit (Periodic Interrupt Enable)

This bit enables or disabled a periodic interrupt.

RTCOS bit (RTCOUT Output Select)

This bit selects the RTCOUT output period. The RTCOS bit must be rewritten while the count operation is stopped (the RCR2.START bit is 0) and the RTCOUT output is disabled (the RCR2.RTCOE bit is 0). When the RTCOUT is output to an external pin, the RCR2.RTCOE bit must be enabled. For details on controlling I/O Ports, see section 20.5.1, Procedure for Specifying the Pin Functions.

PES[3:0] bits (Periodic Interrupt Select)

These bits specify the period for the periodic interrupt. A periodic interrupt is generated with the period specified by these bits.

25.2.18 RTC Control Register 2 (RCR2)

(1) In calendar count mode:

Address(es): RTC.RCR2 4004 4024h

	b7	b6	b5	b4	b3	b2	b1	b0
	CNTMD	HR24	AADJP	AADJE	RTCOE	ADJ30	RESET	START
Value after reset:	x	x	x	x	0	0	0	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	START	Start	0: Prescaler and time counter are stopped 1: Prescaler and time counter operate normally.	R/W
b1	RESET	RTC Software Reset	• In writing: 0: Writing is invalid 1: The prescaler and the target registers for RTC software reset *1 are initialized • In reading: 0: In normal time operation, or an RTC software reset has completed 1: During an RTC software reset.	R/W
b2	ADJ30	30-Second Adjustment	• In writing: 0: Writing is invalid 1: 30-second adjustment is executed. • In reading: 0: In normal time operation, or 30-second adjustment has completed 1: During 30-second adjustment.	R/W
b3	RTCOE	RTCOUT Output Enable	0: RTCOUT output disabled 1: RTCOUT output enabled.	R/W
b4	AADJE	Automatic Adjustment Enable*2	0: Automatic adjustment is disabled 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select*2	0: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every minute 1: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every 10 seconds.	R/W
b6	HR24	Hours Mode	0: The RTC operates in 12-hour mode 1: The RTC operates in 24-hour mode.	R/W
b7	CNTMD	Count Mode Select	0: Calendar count mode 1: Binary count mode.	R/W

Note 1. R64CNT, RSECAR/BCNT0AR, RMINAR/BCNT1AR, RHRAR/BCNT2AR, RWKAR/BCNT3AR, RDAYAR/BCNT0AER, RMONAR/BCNT1AER, RYRAR/BCNT2AER, RYRAREN/BCNT3AER, RADJ, RTCCRy, RSECCPy/BCNT0CPy, RMINCPy/BCNT1CPy, RHRCPy/BCNT2CPy, RDAYCPy/BCNT3CPy, RMONCPy, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP.

Note 2. When LOCO is selected, the setting of this bit is disabled.

The RCR2 register is related to hours mode, automatic adjustment function, enabling RTCOUT output, 30-second adjustment, RTC software reset, and controlling count operation.

START bit (Start)

This bit stops or restarts the prescaler or time counter operation.

The START bit is updated in synchronization with the next cycle of the count source. When the START bit is modified, check that the bit is updated before proceeding.

RESET bit (RTC Software Reset)

This bit initializes the prescaler and registers to be reset by RTC software.

When 1 is written to the RESET bit, initialization starts in synchronization with the count source. When the initialization is completed, the RESET bit is automatically set to 0. Check that this bit is set to 0 before proceeding.

ADJ30 bit (30-Second Adjustment)

This bit is for 30-second adjustment.

When 1 is written to the ADJ30 bit, the RSECCNT value of 30 seconds or less is rounded down to 00 second and the value of 30 seconds or more is rounded up to 1 minute.

The 30-second adjustment is performed in synchronization with the count source. When 1 is written to this bit, the ADJ30 bit is automatically set to 0 after the 30-second adjustment is completed. If 1 is written to the ADJ30 bit, check that the bit is set to 0 before proceeding. When the 30-second adjustment is performed, the prescaler and R64CNT are also reset. The ADJ30 bit is set to 0 by an RTC software reset.

RTCOE bit (RTCOUT Output Enable)

This bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.

Use the START bit to stop counting before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time.

When RTCOUT is to be output from an external pin, enable the RTCOE bit and set up the port control for the pin.

AADJE bit (Automatic Adjustment Enable)

This bit controls (enables or disables) automatic adjustment.

Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit.

The AADJE bit is set to 0 by an RTC software reset.

AADJP bit (Automatic Adjustment Period Select)

This bit selects the automatic-adjustment period.

Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit.

The AADJP bit is set to 0 by an RTC software reset.

HR24 bit (Hours Mode)

This bit specifies whether the RTC operates in 12- or 24-hour mode.

Use the START bit to stop counting before changing the value of the HR24 bit. Do not stop counting (write 0 to the START bit) and change the value of the HR24 bit at the same time.

CNTMD bit (Count Mode Select)

This bit specifies whether the RTC count mode operates in calendar count mode or in binary count mode.

When setting the count mode, execute an RTC software reset and start again from the initial settings. This bit is updated synchronously with the count source, and its value is fixed before the RTC software reset is complete.

For details on initial settings, see section 25.3.1, Outline of Initial Settings of Registers after Power On.

(2) In binary count mode:

Address(es): RTC.RCR2 4004 4024h

	b7	b6	b5	b4	b3	b2	b1	b0
	CNTMD	—	AADJP	AADJE	RTCOE	—	RESET	START
Value after reset:	x	x	x	x	0	0	0	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	START	Start	0: The 32-bit binary counter, 64-Hz counter, and prescaler are stopped 1: The 32-bit binary counter, 64-Hz counter, and prescaler are in normal operation.	R/W
b1	RESET	RTC Software Reset	• In writing 0: Writing is invalid 1: The prescaler and the target registers for RTC software reset*1 are initialized • In reading 0: In normal time operation, or an RTC software reset has completed. 1: During an RTC software reset	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	RTCOE	RTCOUT Output Enable	0: RTCOUT output disabled 1: RTCOUT output enabled.	R/W
b4	AADJE	Automatic Adjustment Enable*2	0: Automatic adjustment is disabled 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select*2	0: Adds or subtracts the RADJ.ADJ[5:0] bits from the prescaler count value every 32 seconds 1: Adds or subtracts the RADJ.ADJ[5:0] bits from the prescaler count value every 8 seconds	R/W
b6	—	Reserved	This bit is undefined. The write value should be 0.	R/W
b7	CNTMD	Count Mode Select	0: The calendar count mode 1: The binary count mode.	R/W

Note 1. R64CNT, RSECAR/BCNT0AR, RMINAR/BCNT1AR, RHRAR/BCNT2AR, RWKAR/BCNT3AR, RDAYAR/BCNT0AER, RMONAR/BCNT1AER, RYRAR/BCNT2AER, RYRAREN/BCNT3AER, RADJ, RTCCRy, RSECCPy/BCNT0CPy, RMINCPy/BCNT1CPy, RHRCPy/BCNT2CPy, RDAYCPy/BCNT3CPy, RMONCPy, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP

Note 2. When LOCO is selected, the setting of this bit is disabled.

START bit (Start)

This bit stops or restarts the prescaler or counter (clock) operation.

The START bit is updated in synchronization with the count source. When the START bit is modified, check that the bit is updated before proceeding.

RESET bit (RTC Software Reset)

This bit initializes the prescaler and registers to be reset by RTC software.

When 1 is written to the RESET bit, initialization starts in synchronization with the count source. When the initialization is completed, the RESET bit is automatically set to 0. When 1 is written to the RESET bit, check that the bit is set to 0 before proceeding.

RTCOE bit (RTCOUT Output Enable)

This bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.

Use the START bit to stop counting before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time. When an RTCOUT signal is to be output from an external pin, enable the port control as well as setting this bit.

AADJE bit (Automatic Adjustment Enable)

This bit controls (enables or disables) automatic adjustment.

Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit. The AADJE bit is set to 0 by an RTC software reset.

AADJP bit (Automatic Adjustment Period Select)

This bit selects the automatic-adjustment period.

Correction period can be selected from 32 second units or 8 second units in binary count mode.

Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit. The AADJP bit is set to 0 by an RTC software reset.

CNTMD bit (Count Mode Select)

This bit specifies whether the RTC count mode operates in calendar count mode or in binary count mode.

When setting the count mode, execute an RTC software reset and start again from the initial settings. This bit is updated synchronously with the count source, and its value is fixed before the RTC software reset is completed.

For details on initial settings, see section 25.3.1, Outline of Initial Settings of Registers after Power On.

25.2.19 RTC Control Register 4 (RCR4)

Address(es): RTC.RCR4 4004 4028h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RCKSEL
Value after reset:	0	0	0	0	0	0	0	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	RCKSEL	Count Source Select	0: Sub-clock oscillator is selected 1: LOCO is selected.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RCR4 register selects the count source and is used in both calendar count mode and binary count mode.

When the RCKSEL bit is set to 0, the time is counted with the sub-clock. When the bit is set to 1, the time is counted with LOCO.

RCKSEL bit (Count Source Select)

This bit selects the count source from the sub-clock and LOCO.

The count source must be selected only once before making the initial settings of the RTC registers at power on.

25.2.20 Frequency Register (RFRH/RFRL)

Address(es): RTC.RFRH 4004 402Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	RFC16	Reserved	Write 0 before writing to the RFRL register after a cold start.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Address(es): RTC.RFRL 4004 402Ch

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

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b15 to b0	RFC[15:0]	Frequency Comparison Value	Write 00FFh to this register when using the LOCO.	R/W

RFRL is a register for controlling the prescaler when LOCO is selected.

The RTC time counter operates on a 128-Hz clock signal as the base clock. Therefore, when LOCO is selected, LOCO is divided by the prescaler to generate a 128-Hz clock signal. Set the frequency comparison value in the RFC[15:0] bits to generate a 128-Hz clock from the LOCO frequency. Before writing to RFC[15:0] after a cold start, write 0000h to the RFRH.

A value from 0007h through 01FFh can be specified as the frequency comparison value. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2. The operating frequency of the peripheral module clock and the LOCO should be such that the peripheral module clock is greater than or equal to the LOCO.

Calculation method of frequency comparison value:
RFC[15:0] = (LOCO clock frequency) / 128 - 1

When the LOCO frequency is 32.768 kHz, the RFRL register should be set to 00FFh.

25.2.21 Time Error Adjustment Register (RADJ)

Address(es): RTC.RADJ 4004 402Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	PMADJ[1:0]		ADJ[5:0]					
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b5 to b0	ADJ[5:0]	Adjustment Value	These bits specify the adjustment value from the prescaler	R/W
b7, b6	PMADJ[1:0]	Plus–Minus	b7 b6 0 0: Adjustment is not performed 0 1: Adjustment is performed by the addition to the prescaler 1 0: Adjustment is performed by the subtraction from the prescaler 1 1: Setting prohibited.	R/W

Adjustment is performed by the addition to or subtraction from the prescaler. If the automatic adjustment enable (RCR2.AADJE) bit is 0, adjustment is performed when writing to the RADJ. If the RCR2.AADJE bit is 1, adjustment is performed in the interval specified by the automatic adjustment period select (RCR2.AADJP) bit.

The current adjustment by software (disabling automatic adjustment) might be invalid if the following adjustment value is specified within 320 cycles of the count source after the register setting. To perform adjustment consecutively, wait for 320 cycles or more of the count source after the register setting, then specify the next adjustment value.

RADJ is updated in synchronization with the count source. When RADJ is modified, check that all the bits are updated before continuing with further processing. This register is set to 00h by an RTC software reset. The setting of this register is enabled only when the sub-clock is selected. When LOCO is selected, adjustment is not performed.

ADJ[5:0] bits (Adjustment Value)

These bits specify the adjustment value (the number of sub-clock cycles) from the prescaler.

PMADJ[1:0] bits (Plus–Minus)

These bits select whether the clock is set ahead or back depending on the error-adjustment value set in the ADJ[5:0] bits.

25.2.22 Time Capture Control Register y (RTCCRy) (y = 0 to 2)

Address(es): RTC.RTCCR0 4004 4040h, RTC.RTCCR1 4004 4042h, RTC.RTCCR2 4004 4044h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCEN	—	TCNF[1:0]		—	TCST	TCCT[1:0]	
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b1, b0	TCCT[1:0]	Time Capture Control	b1 b0 0 0: No event is detected 0 1: Rising edge is detected 1 0: Falling edge is detected 1 1: Both edges are detected.	R/W
b2	TCST	Time Capture Status	0: No event is detected 1: An event is detected.*1	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b5, b4	TCNF[1:0]	Time Capture Noise Filter Control	b5 b4 0 0: Noise filter is off 0 1: Setting prohibited 1 0: Noise filter is on (count source) 1 1: Noise filter is on (count source by divided by 32).	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	TCEN	Time Capture Event Input Pin Enable	0: The RTCICn pin is disabled as the time capture event input 1: The RTCICn pin is enabled as the time capture event input. (n = 0 to 2).	R/W

Note 1. Indicates that an event has been detected. Writing 1 to this bit has no effect. Writing 0 sets this bit to 0.

The RTCCRy register is used both in calendar count mode and in binary count mode. RTCCR0, RTCCR1, and RTCCR2 control the RTCIC0, RTCIC1, and RTCIC2 pins, respectively.

RTCCRy is updated in synchronization with the count source. When RTCCRy is modified, check that all the bits except for the TCST bit are updated before continuing with further processing. This register is set to 00h by an RTC software reset. When RTCICn is used as the time capture pin, VBTICTLR.VCHnIEN (n = 0 to 2) must be set to 1. For more information, see section 12, Battery Backup Function.

TCCT[1:0] bits (Time Capture Control)

These bits control the edge detection of the time capture event input pins, RTCIC0, RTCIC1, and RTCIC2. The detection edge is selectable. The TCCT[1:0] bits must be set while the VBTICTLR.VCHnIEN bit is 1.

TCST bit (Time Capture Status)

This bit indicates that an event of the time capture event input pins, RTCIC0, RTCIC1, and RTCIC2, has been detected. When the TCST bit is 0, no event is detected. When the TCST bit is 1, this bit indicates that an event of the associated pin has been detected and the capture register is valid. When multiple events have been detected, the capture time for the first event is retained.

If an event is detected while the count operation stops, that is, the RCR2.START bit is 0, the captured value is not guaranteed. In this case, set the TCST bit to 0 to delete the captured value. Writing 0 sets the TCST bit to 0. Writing any other value except 0 has no effect.

Set the TCST bit while the TCCT[1:0] bits are 00b (no event is detected). The TCST bit is set to 0 in synchronization with the count source. When the TCST bit is set to 0, check that the bit is updated before continuing with further processing.

TCNF[1:0] bits (Time Capture Noise Filter Control)

These bits control the noise filter of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2).

When the noise filter is on, the count source divided by 1 or divided by 32 is selectable. In this case, when the input level on the time capture event input pin matches three consecutive times at the set sampling period, the input level is determined.

Set the TCNF[1:0] bits while the TCCT[1:0] bits are 00b (no event is detected). When the noise filter is used, set the TCNF[1:0] bits, wait for 3 cycles of the specified sampling period, then set the TCCT[1:0] bits. Set the TCNF[1:0] bits when the VBTICTLR.VCHnIEN bit is 1.

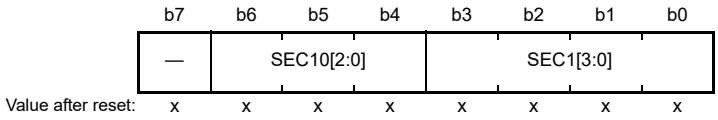
TCEN bit (Time Capture Event Input Pin Enable)

The TCEN bit enables or disables the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2). When the functions of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2) are multiplexed, set VBTICTLR first. If the TCEN bit is set to 0, also set the TCCT[1:0] bits to 00b.

25.2.23 Second Capture Register y (RSECCPy) (y = 0 to 2)/BCNT0 Capture Register y (BCNT0CPy) (y = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RSECCP0 4004 4052h, RTC.RSECCP1 4004 4062h, RTC.RSECCP2 4004 4072h



x: Undefined

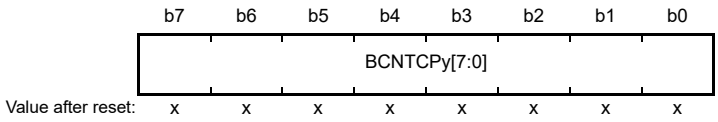
Bit	Symbol	Bit name	Description	R/W
b3 to b0	SEC1[3:0]	1-Second Capture	Capture value for the ones place of seconds	R
b6 to b4	SEC10[2:0]	10-Second Capture	Capture value for the tens place of seconds	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset	R

RSECCPy is a read-only register that captures the RSECCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RSECCP0, RSECCP1, and RSECCP2 registers, respectively. This register is set to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT0CP0 4004 4052h, RTC.BCNT0CP1 4004 4062h, RTC.BCNT0CP2 4004 4072h



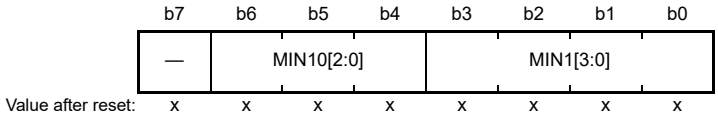
x: Undefined

BCNT0CPy is a read-only register that captures the BCNT0 value when a time capture event is detected. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT0CP0, BCNT0CP1, and BCNT0CP2 registers, respectively. This register is set to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

25.2.24 Minute Capture Register y (RMINCPy) (y = 0 to 2)/BCNT1 Capture Register y (BCNT1CPy) (y = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RMINCP0 4004 4054h, RTC.RMINCP1 4004 4064h, RTC.RMINCP2 4004 4074h



x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	MIN1[3:0]	1-Minute Capture	Capture value for the ones place of minutes	R
b6 to b4	MIN10[2:0]	10-Minute Capture	Capture value for the tens place of minutes	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset.	R

RMINCPy is a read-only register that captures the RMINCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMINCP0, RMINCP1, and RMINCP2 registers, respectively.

This register is set to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT1CP0 4004 4054h, RTC.BCNT1CP1 4004 4064h, RTC.BCNT1CP2 4004 4074h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTCPy[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT1CPy is a read-only register that captures the BCNT1 value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT1CP0, BCNT1CP1, and BCNT1CP2 registers, respectively.

This register is set to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

25.2.25 Hour Capture Register y (RHRCPy) (y = 0 to 2)/BCNT2 Capture Register y (BCNT2CPy) (y = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RHRCP0 4004 4056h, RTC.RHRCP1 4004 4066h, RTC.RHRCP2 4004 4076h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	HR1[3:0]	1-Hour Capture	Capture value for the ones place of hours	R
b5, b4	HR10[1:0]	10-Hour Capture	Capture value for the tens place of hours	R
b6	PM	PM	0: a.m. 1: p.m.	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset	R

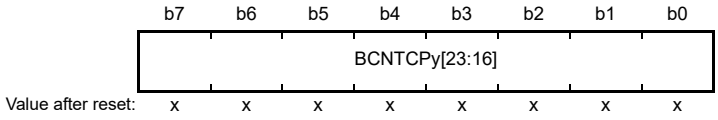
RHRCPy is a read-only register that captures the RHRCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RHRCP0, RHRCP1, and RHRCP2 registers, respectively. The PM bit is only enabled when the RCR2.HR24 bit is 0 (in 12-hour mode).

This register is set to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT2CP0 4004 4056h, RTC.BCNT2CP1 4004 4066h, RTC.BCNT2CP2 4004 4076h



x: Undefined

BCNT2CPy is a read-only register that captures the BCNT2 value when a time capture event is detected.

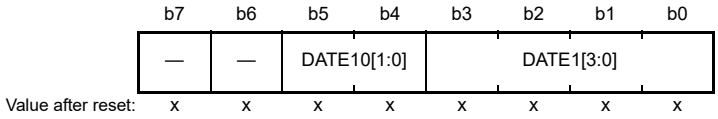
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT2CP0, BCNT2CP1, and BCNT2CP2 registers, respectively.

This register is set to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

25.2.26 Date Capture Register y (RDAYCPy) (y = 0 to 2)/BCNT3 Capture Register y (BCNT3CPy) (y = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RDAYCP0 4004 405Ah, RTC.RDAYCP1 4004 406Ah, RTC.RDAYCP2 4004 407Ah



x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	DATE1[3:0]	1-Day Capture	Capture value for the ones place of days	R
b5, b4	DATE10[1:0]	10-Day Capture	Capture value for the tens place of days	R
b7, b6	—	Reserved	These bits are read as 0 after an RTC software reset.	R

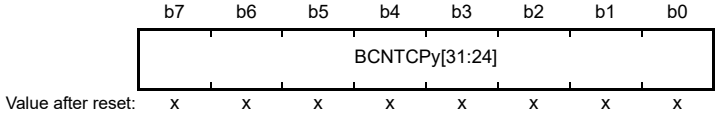
RDAYCPy is a read-only register that captures the RDAYCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RDAYCP0, RDAYCP1, and RDAYCP2 registers, respectively.

This register is set to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT3CP0 4004 405Ah, RTC.BCNT3CP1 4004 406Ah, RTC.BCNT3CP2 4004 407Ah



x: Undefined

BCNT3CPy is a read-only register that captures the BCNT3 value when a time capture event is detected.

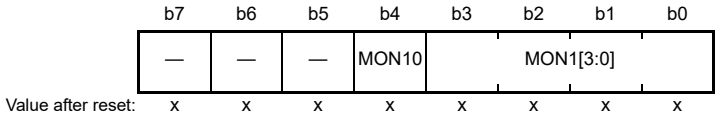
The event detection times detected by the RTCTC0, RTCTC1, and RTCTC2 pins are stored in the BCNT3CP0, BCNT3CP1, and BCNT3CP2 registers, respectively.

This register is set to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

25.2.27 Month Capture Register y (RMONCPy) (y = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RMONCP0 4004 405Ch, RTC.RMONCP1 4004 406Ch, RTC.RMONCP2 4004 407Ch



x: Undefined

Bit	Symbol	Bit name	Description	R/W
b3 to b0	MON1[3:0]	1-Month Capture	Capture value for the ones place of months	R
b4	MON10	10-Month Capture	Capture value for the tens place of months	R
b7 to b5	—	Reserved	These bits are read as 0	R

RMONCPy is a read-only register that captures the RMONCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMONCP0, RMONCP1, and RMONCP2 registers, respectively.

This register is set to 00h by an RTC software reset. Before reading from this register, be sure to stop the time capture event detection using the RTCCRy.TCCT[1:0] bits.

25.3 Operation

25.3.1 Outline of Initial Settings of Registers after Power On

After the power is turned on, the initial settings for the clock setting, count mode setting, time error adjustment, time setting, alarm, interrupt, and time capture control register should be performed.

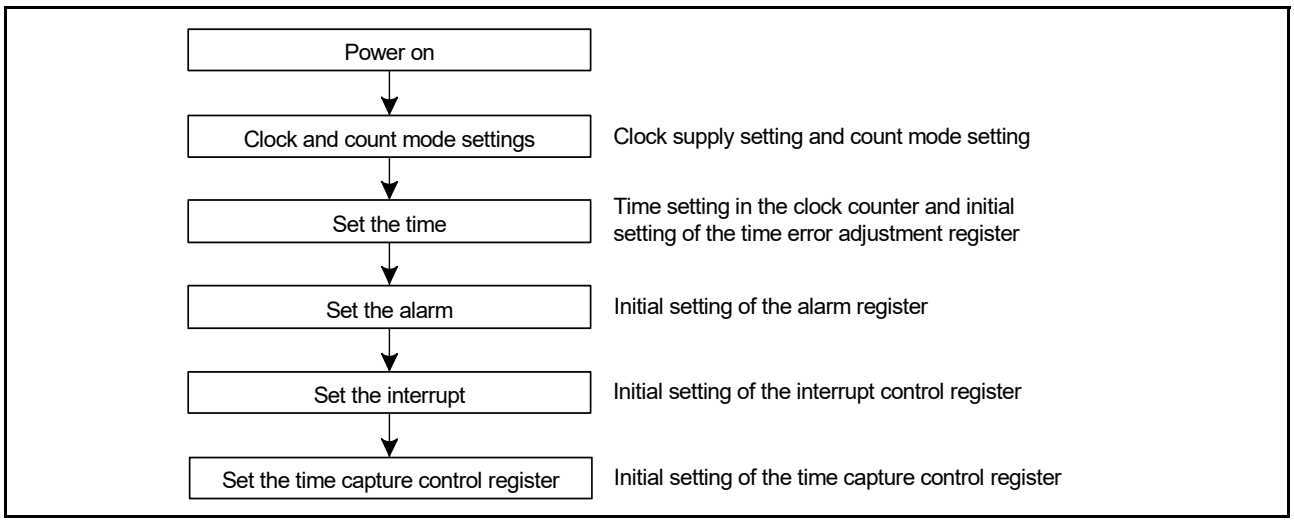


Figure 25.2 Outline of initial settings after a power on

25.3.2 Clock and Count Mode Setting Procedure

Figure 25.3 shows how to set the clock and the count mode.

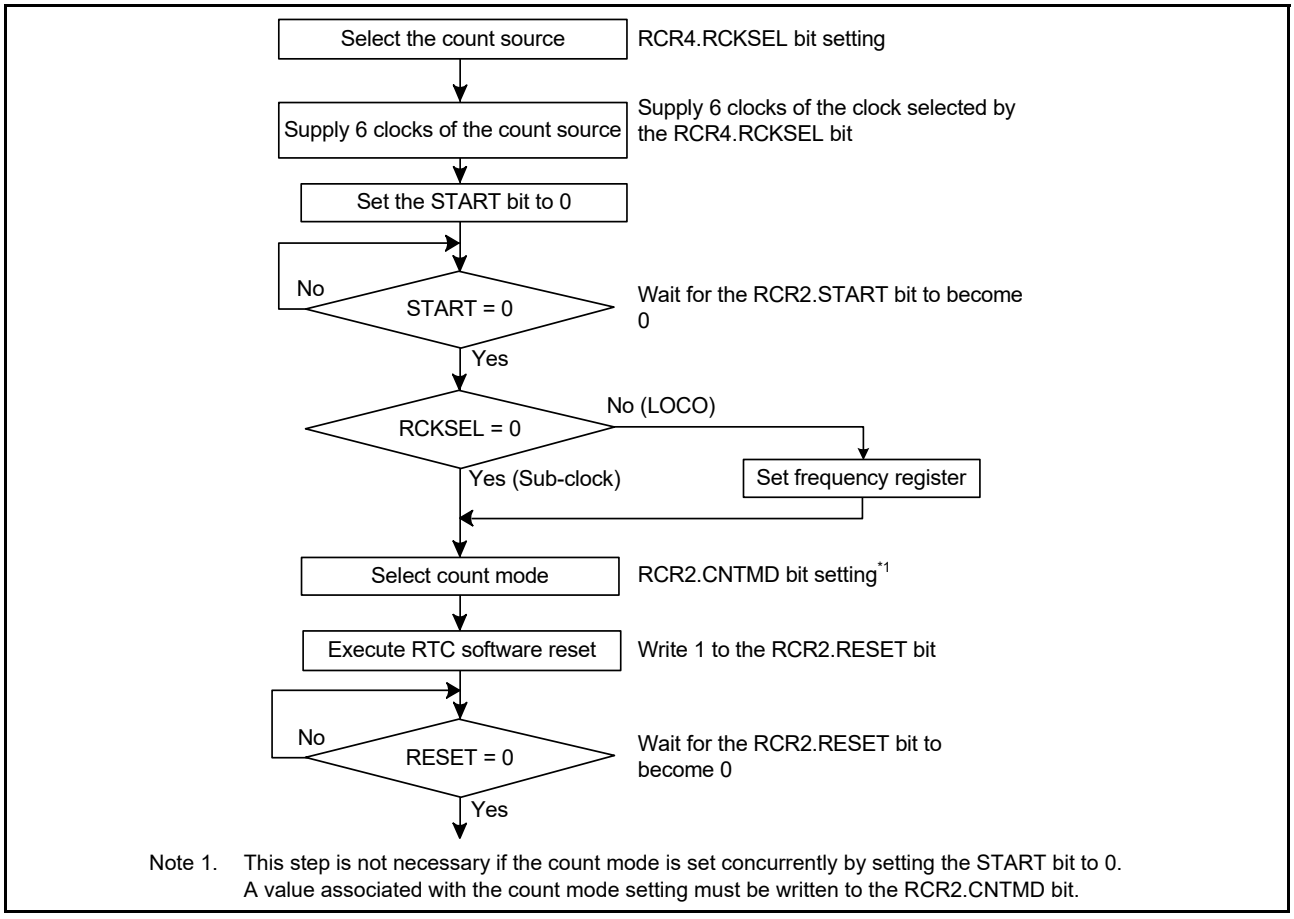
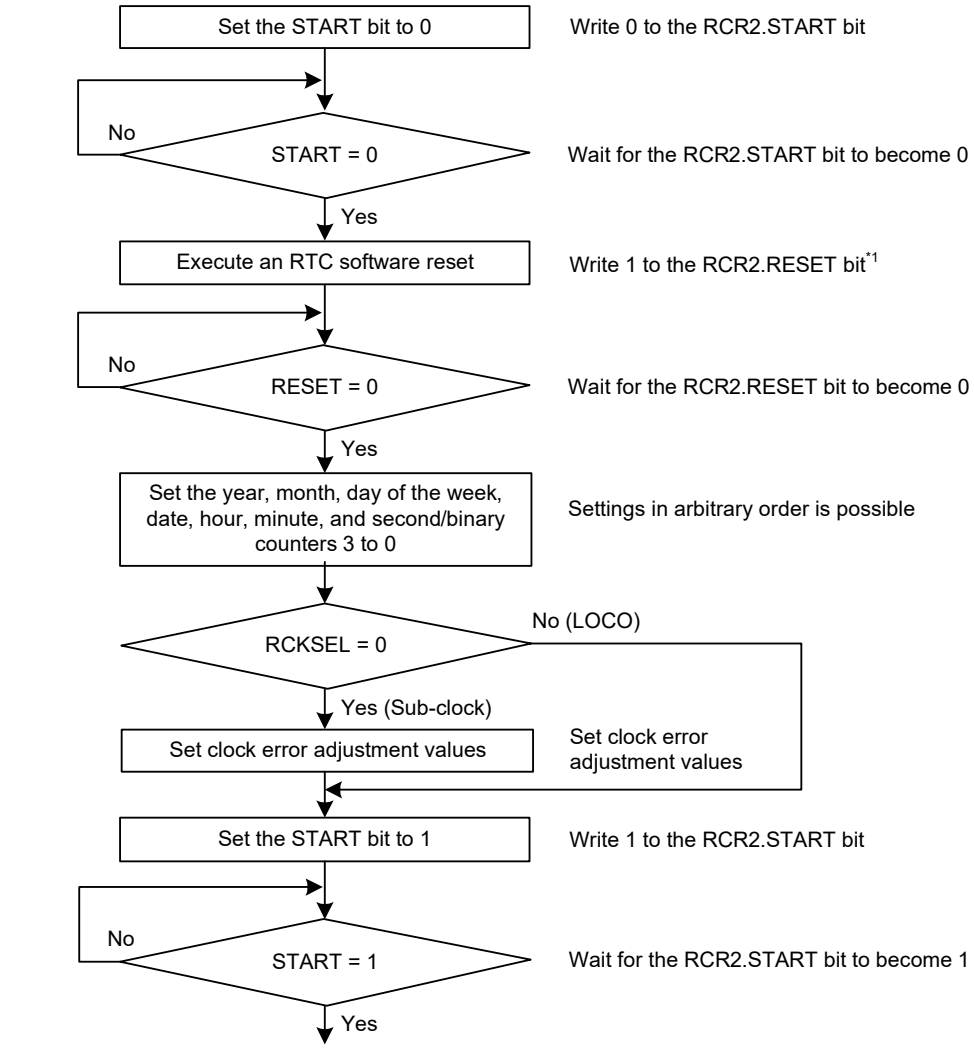


Figure 25.3 Clock and count mode setting procedure

25.3.3 Setting the Time

Figure 25.4 shows how to set the time.



Note 1. This step is not necessary for the time-setting procedure because an RTC software reset is executed in the clock setting procedure of the initial settings for the power supply.

Figure 25.4 Setting the time

25.3.4 30-Second Adjustment

Figure 25.5 shows how to execute a 30-second adjustment.

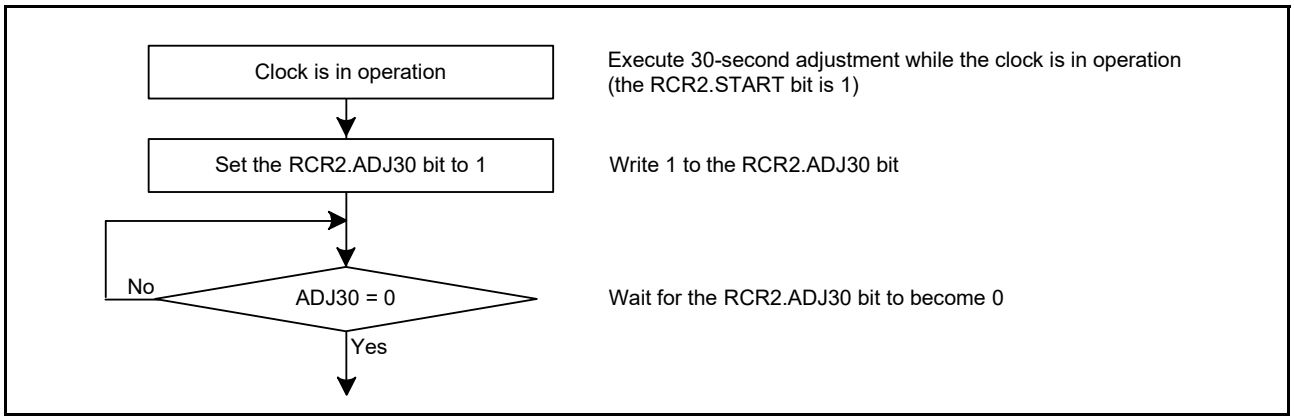


Figure 25.5 30-Second adjustment

25.3.5 Reading 64-Hz Counter and Time

Figure 25.6 shows how to read a 64-Hz counter and time.

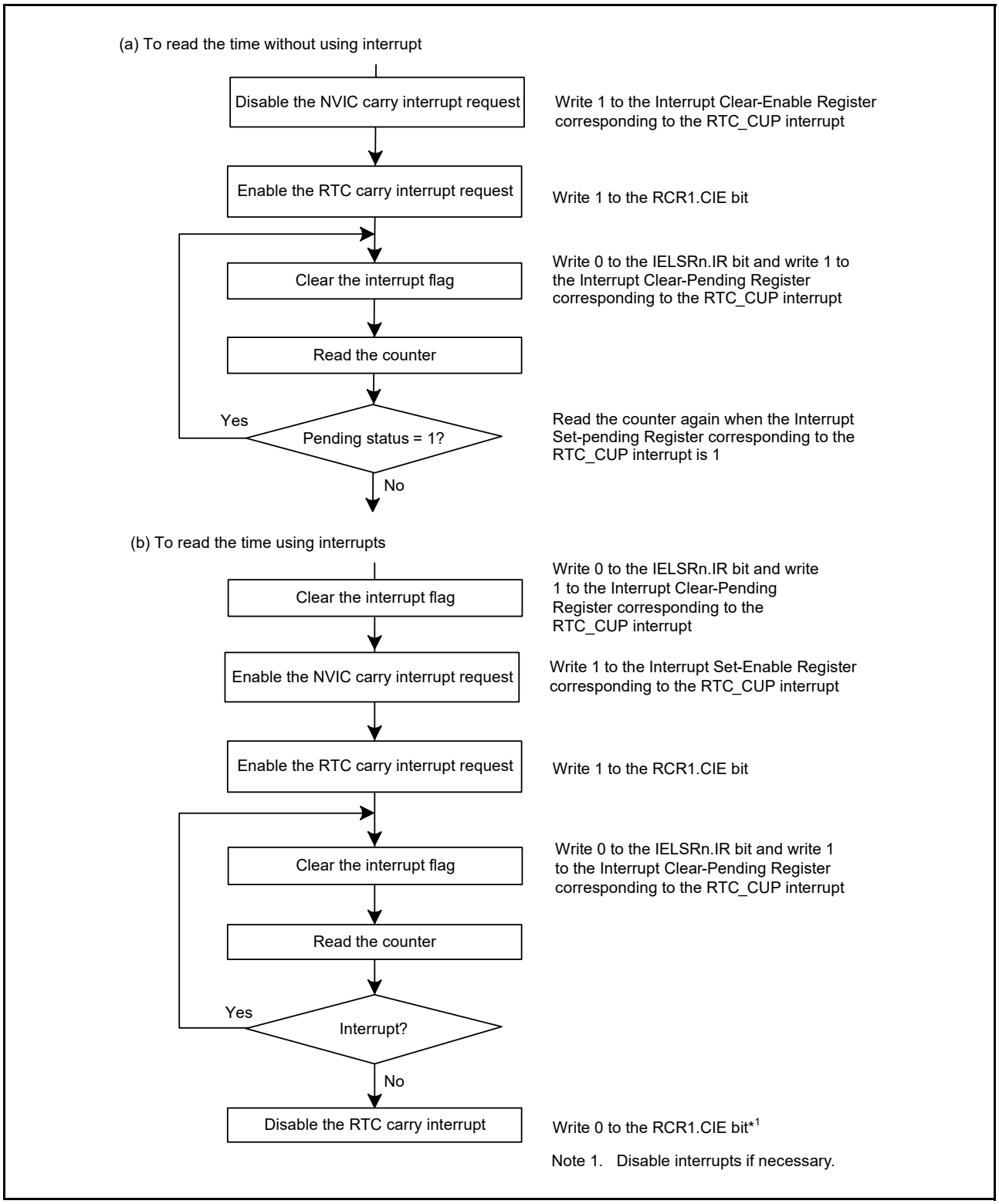


Figure 25.6 Reading time

If a carry occurs while the 64-Hz counter and time are read, the correct time is not obtained, therefore they must be read again. The procedure for reading the time without using interrupts is shown in (a) in Figure 25.6, and the procedure using carry interrupts is shown in (b). To keep the program simple, method (a) should be used in most cases.

25.3.6 Alarm Function

Figure 25.7 shows how to use the alarm function.

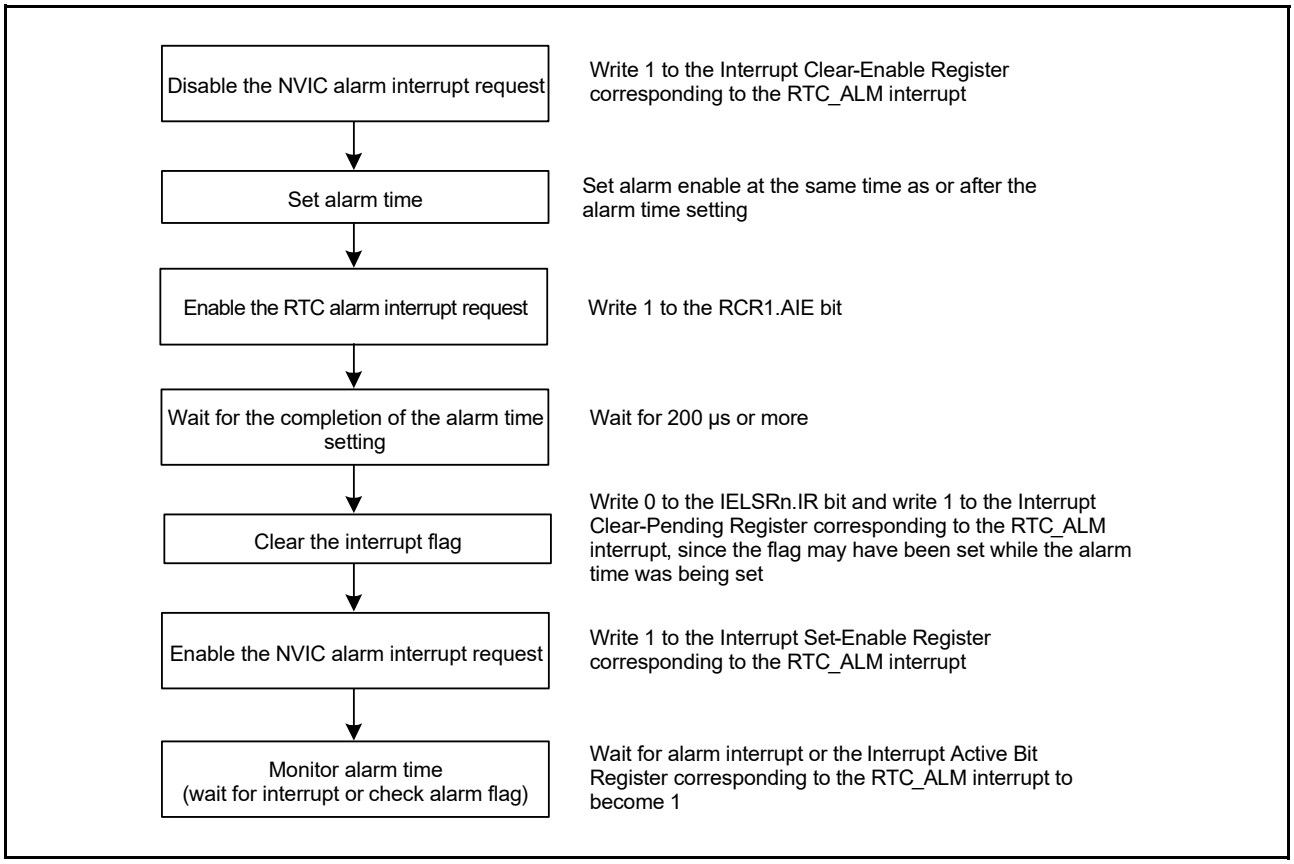


Figure 25.7 Using alarm function

In calendar count mode, an alarm can be generated by any one of year, month, date, day-of-week, hour, minute or second, or any combination of those. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

In binary count mode, an alarm can be generated in any bit combination of 32 bits. Write 1 to the ENB bit of the alarm enable register associated with the target bit of the alarm, and set the alarm time in the alarm register. For bits that are not the target of the alarm, write 0 to the ENB bit of the alarm enable register.

When the counter and the alarm time match, the IELSRn.IR bit and Interrupt Set-Pending/Clear-Pending Register corresponding to the RTC_ALM interrupt is set to 1. Alarm detection can be confirmed by reading the Interrupt Set-Pending Register corresponding to the RTC_ALM interrupt, but an interrupt should be used in most cases. If 1 is set in the Interrupt Set-Enable Register associated with the RTC_ALM interrupt, an alarm interrupt is generated in the event of the alarm, enabling the alarm to be detected.

Writing 0 sets the IELSRn.IR bit associated with the RTC_ALM interrupt to 0. If interrupt is enabled, the Interrupt Set-Pending/Clear-Pending Register and Interrupt Active Bit Register corresponding to the RTC_ALM interrupt is cleared automatically after exiting the interrupt handler. Otherwise, write 1 to the Interrupt Clear-Pending Register associated with the RTC_ALM interrupt to clear it.

When the counter and the alarm time match in a low power consumption state, the MCU returns from the low power consumption state.

25.3.7 Procedure for Disabling Alarm Interrupt

Figure 25.8 shows the procedure for disabling the enabled alarm interrupt request.

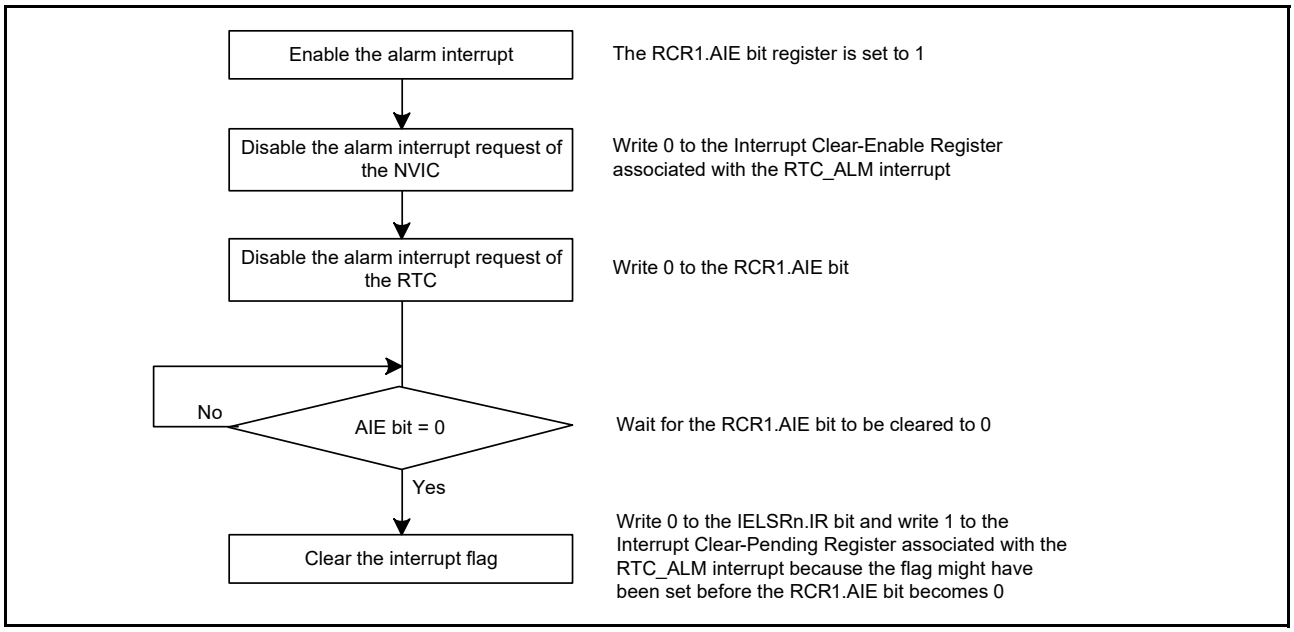


Figure 25.8 Procedure for disabling alarm interrupt request

25.3.8 Time Error Adjustment Function

The time error adjustment function is used to correct errors, running fast or slow, in the time due to variation in the precision of oscillation by the sub-clock. Because 32,768 cycles of the sub-clock constitute 1 second of operation when the sub-clock is selected, the clock runs fast if the sub-clock frequency is high and slow if the sub-clock frequency is low. There are two types of time error adjustment functions:

- Automatic Adjustment
- Adjustment by Software.

Use the RCR2.AADJE bit to select automatic adjustment or adjustment by software.

25.3.8.1 Automatic Adjustment

Enable automatic adjustment by setting the RCR2.AADJE bit to 1. Automatic adjustment is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register every time the adjustment period selected by the RCR2.AADJP bit elapses.

(1) [Example 1] Sub-clock running at 32.769 kHz

(a) Adjustment procedure

When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by 60 clock cycles per minute, so adjustment can take the form of setting the clock back by 60 cycles every minute.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 0 (adjustment every minute)
- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler)
- RADJ.ADJ[5:0] = 60 (3Ch).

(2) [Example 2] Sub-clock running at 32.766 kHz

(a) Adjustment procedure

When the sub-clock is running at 32.766 kHz, 1 second elapses every 32,766 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs slow by 2 clock cycles every second. The time on the clock is slow by 20 clock cycles every 10 seconds, so adjustment can take the form of setting the clock forward by 20 cycles every 10 seconds.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 1 (adjustment every 10 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 20 (14h).

(3) [Example 3] Sub-clock running at 32.764 kHz

(a) Adjustment procedure

At 32.764 kHz, 1 second elapses on 32,764 clock cycles. Because the RTC operates for 32,768 clock cycles as 1 second, the clock is delayed for 4 clock cycles per second. In 8 seconds, the delay is 32 clock cycles, therefore correction can be made by proceeding the clock for 32 clock cycles every 8 seconds.

Register settings when the RCR2.CNTMD bit is 1

- RCR2.AADJP = 1 (adjustment every 8 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler)
- RADJ.ADJ[5:0] = 32 (20h).

25.3.8.2 Adjustment by Software

Enable adjustment by software by setting the RCR2.AADJE bit to 0. Adjustment by software is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register at the time of execution of a write instruction to the RADJ register.

(1) [Example 1] Sub-clock running at 32.769 kHz

(a) Adjustment procedure

When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by one clock cycle per second, so adjustment can take the form of setting the clock back by one cycle every second.

(b) Register settings

- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler)
- RADJ.ADJ[5:0] = 1 (01h)
  This is written to the RADJ register once per 1-second interrupt.

25.3.8.3 Procedure for Changing the Mode of Adjustment

When changing the mode of adjustment, change the value of the AADJE bit in RCR2 after setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

To change adjustment by software to automatic adjustment:

1. Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
2. Set the RCR2.AADJE bit to 1 (automatic adjustment is enabled).
3. Use the RCR2.AADJP bit to select the period of adjustment.
4. In RADJ, set the PMADJ[1:0] bits for addition or subtraction and the ADJ[5:0] bits to the value for use in time error adjustment.

To change automatic adjustment to adjustment by software:

1. Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
2. Set the RCR2.AADJE bit to 0 (adjustment by software is enabled).
3. Proceed with the adjustment by setting the RADJ.PMADJ[1:0] bits for addition or subtraction and the RADJ.ADJ[5:0] bits to the value for use in time error adjustment at the desired time. After that, the time is adjusted every time a value is written to the RADJ register.

25.3.8.4 Procedure for Stopping Adjustment

Stop the adjustment by setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

25.3.8.5 Capturing the Time

The RTC is capable of storing the month, date, hour, minute and second/binary counters 3 to 0 by detecting an edge of a signal on a time capture event input pin.

A noise filter can also be used on a time capture event input pin. If the noise filter is enabled, the TCST bit is set to 1 when the input level on the pin matches three times.

The noise filter can be switched on or off for each of the time capture event input pins. VBTICTLR.VCHnIEN (n = 0 to 2) should be set to 1 to enable the RTCICn input. Operation when the noise filter is off is shown in Figure 25.9 and operation when the noise filter is on is shown in Figure 25.10.

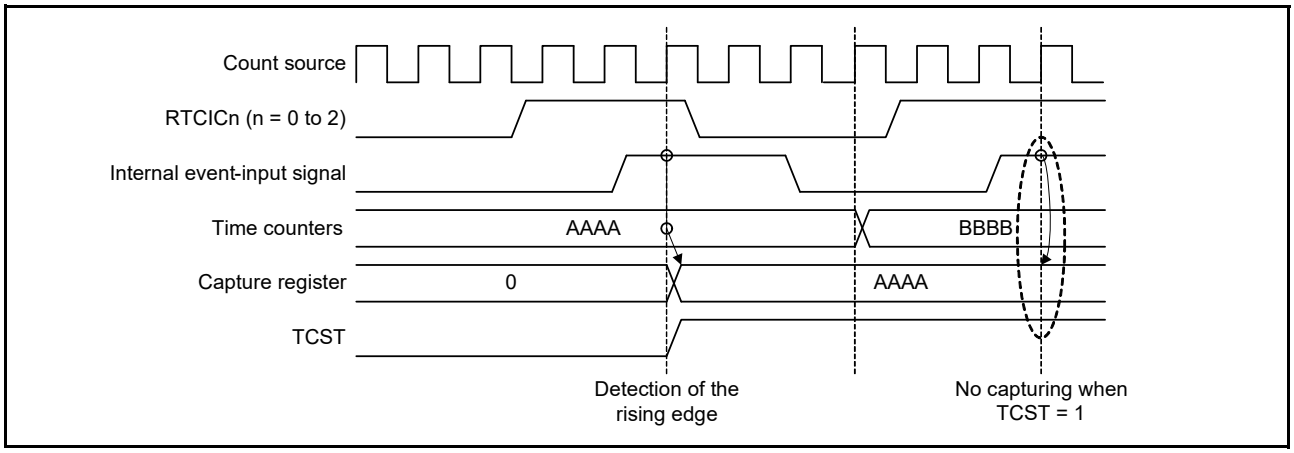


Figure 25.9 Timing of a time capture operation with the filter off

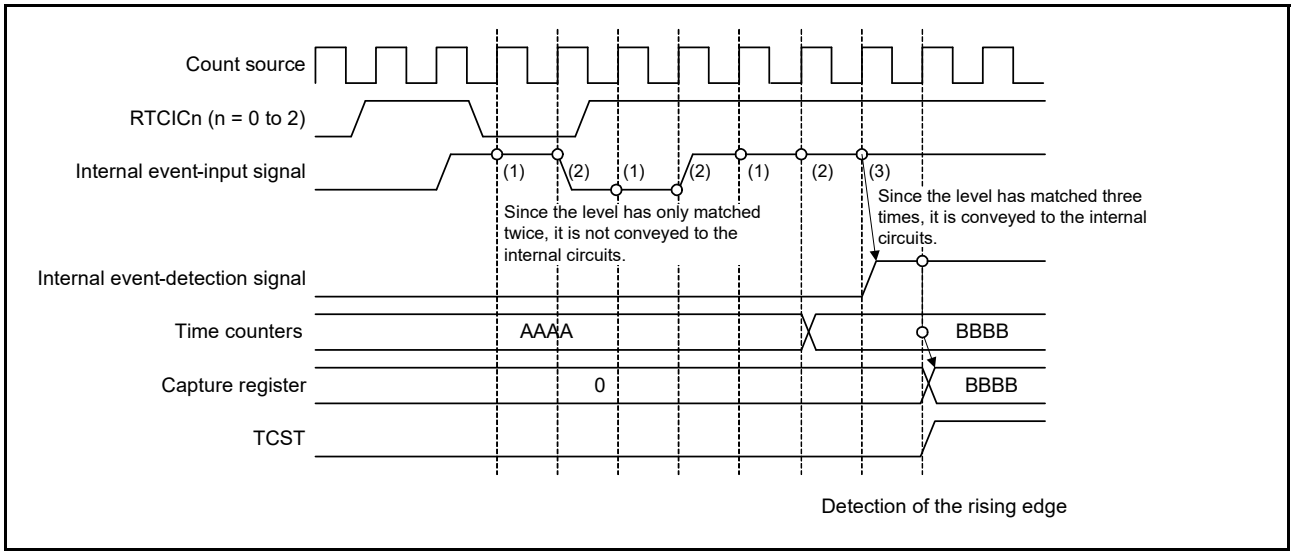


Figure 25.10 Timing of a time capture operation with the filter on

25.4 Interrupt Sources

The RTC has three interrupt sources and are listed in Table 25.3.

Table 25.3 RTC Interrupt sources

Name	Interrupt sources
RTC_ALM	Alarm interrupt
RTC_PRD	Periodic interrupt
RTC_CUP	Carry interrupt

(1) Alarm interrupt (RTC_ALM)

This interrupt is generated according to the result of comparison between the alarm registers and realtime clock counters. For details, see section 25.3.6, Alarm Function.

Because there is a possibility that the interrupt flag might be set to 1 when the settings of the alarm registers match the clock counters, wait for the alarm time settings to be confirmed and clear the IELSRn.IR bit and the interrupt Set-Pending Register corresponding to the RTC_ALM interrupt to 0 again after modifying values of the alarm registers. After the interrupt flag for the alarm interrupt is set to 1 and the state is returned to non-matching of the alarm registers and clock counters, the flag does not set again until there is a further match or the values of the alarm registers are modified again.

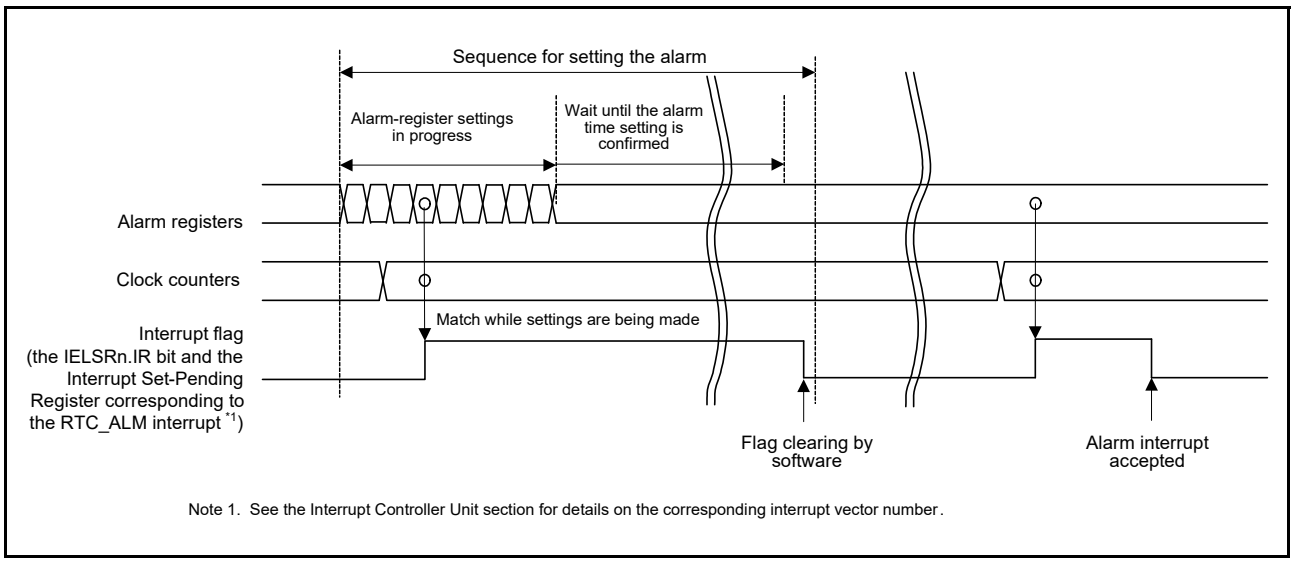


Figure 25.11 Timing chart for the alarm interrupt (RTC_ALM)

(2) Periodic interrupt (RTC_PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second. The interrupt interval can be selected through the RCR1.PES[3:0] bits.

(3) Carry interrupt (RTC_CUP)

This interrupt is generated when a carry to the second counter/binary counter 0 occurred or a carry to the R64CNT counter occurred during read access to the 64-Hz counter.

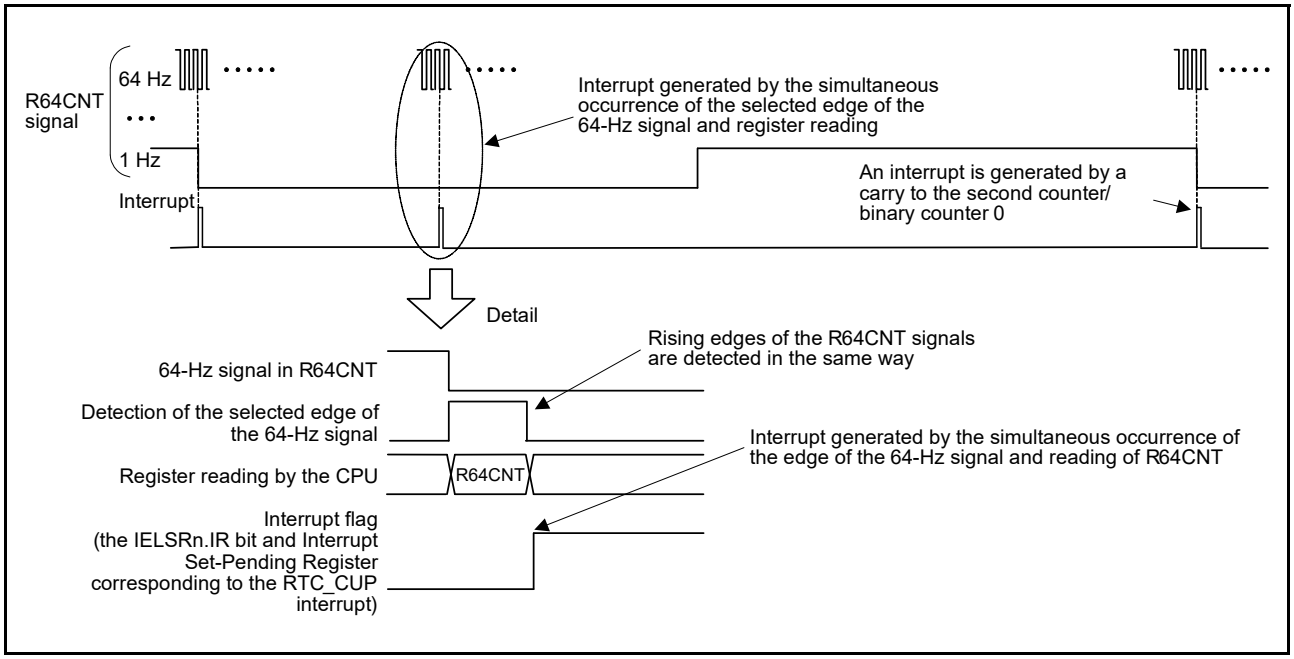


Figure 25.12 Carry interrupt (RTC_CUP) timing chart

25.5 Event Link Output

The RTC generates periodic event output (RTC_PRD) event signals for the Event Link Controller (ELC), and these can be used to initiate operations by other modules selected in advance.

The periodic event signal is output at the interval selected from 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, and 2 seconds by setting the RCR1.PES[3:0] bits. The event generation period immediately after the event generation is selected is not guaranteed.

Note: If event linking from the RTC is used, only set the ELC after setting the RTC, for example initialization and time settings. Setting the RTC after the ELC can lead to output of unexpected event signals.

25.5.1 Interrupt Handling and Event Linking

The RTC has a bit to enable or disable periodic interrupts. An interrupt request signal is output to the CPU when an interrupt source is generated while the corresponding enable bit is enabled.

In contrast, an event link output signal is sent to other modules as an event signal through the ELC when an interrupt source is generated, regardless of the setting of the associated interrupt enable bit.

Note: Although alarm and periodic interrupts can still be output during Software Standby mode, the periodic event signals for the ELC are not output.

25.6 Usage Notes

25.6.1 Register Writing during Counting

The following registers must not be written to during counting, that is, while the RCR2.START bit = 1.

- RSECCNT/BCNT0
- RMINCNT/BCNT1
- RHRCNT/BCNT2
- RDAYCNT
- RWKCNT/BCNT3
- RMONCNT
- RYRCNT
- RCR1.RTCOS
- RCR2.RTCOE
- RCR2.HR24
- RFRL.

The counter must be stopped before writing to any of the these registers.

25.6.2 Use of Periodic Interrupts

The procedure for using periodic interrupts is shown in Figure 25.13.

The generation and period of the periodic interrupt can be changed by setting the RCR1.PES[3:0] bits. However, because the prescaler R64CNT and RSECCNT/BCNT0 are used to generate interrupts, the interrupt period is not guaranteed immediately after setting the RCR1.PES[3:0] bits. In addition, any of the following can affect the interrupt period:

- Stopping/restarting or resetting counter operation
- Reset by RTC software
- 30-second adjustment by changing the RCR2 value.

When the time error adjustment function is used, the interrupt generation period after adjustment is added or subtracted according to the adjustment value.

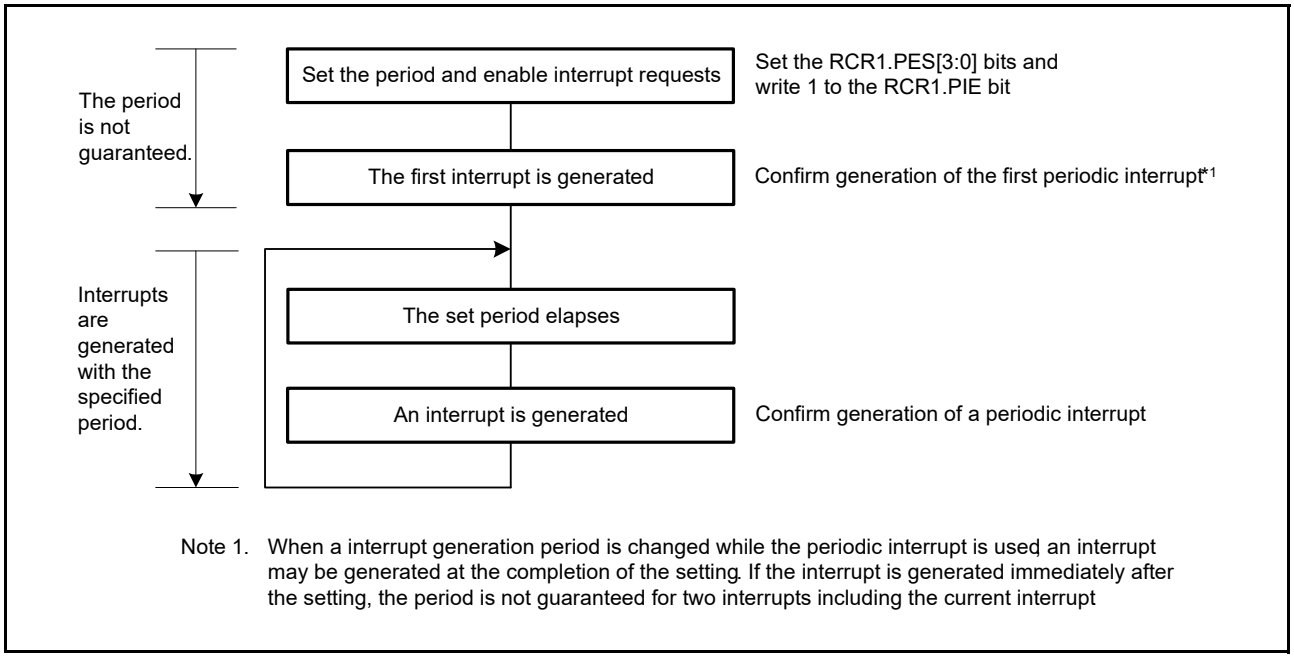


Figure 25.13 Using periodic interrupt function

25.6.3 RTCOUT (1-Hz/64-Hz) Clock Output

Stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the period of RTCOUT (1-Hz/64-Hz) output. When the time error adjustment function is used, the period of RTCOUT (1-Hz/64-Hz) output after adjustment is added or subtracted according to the adjustment value.

25.6.4 Transitions to Low Power Consumption Modes after Setting Registers

A transition to a low power consumption state (Software Standby mode or battery backup) during writing to an RTC register might destroy the value of the register. After setting the register, confirm that the setting is in place before initiating a transition to a low power consumption state.

25.6.5 Notes When Writing to and Reading from Registers

- When reading a counter register such as the second counter after having written to the counter register, follow the procedure in section 25.3.5, Reading 64-Hz Counter and Time
- The value written to the count registers, alarm registers, year alarm enable register, bits RCR2.AADJE, AADJP, and HR24, RCR4 register, or frequency register is reflected when four read operations are performed after writing
- The values written to the RCR1.CIE, RCR1.RTCOS, and RCR2.RTCOE bits can be read immediately after writing
- To read the value from the timer counter after return from a reset, period in Software Standby mode, or the battery backup state, wait for 1/128 second while the clock is operating (RCR2.START bit = 1)
- After a reset is generated, write to the RTC register after 6 cycles of the count source clock elapse.

25.6.6 Changing the Count Mode

When changing the count mode (calendar/binary), set the RCR2.START bit to 0, stop the counting operation, then restart it from the initial setting. For details on the initial setting, see section 25.3.1, Outline of Initial Settings of Registers after Power On.

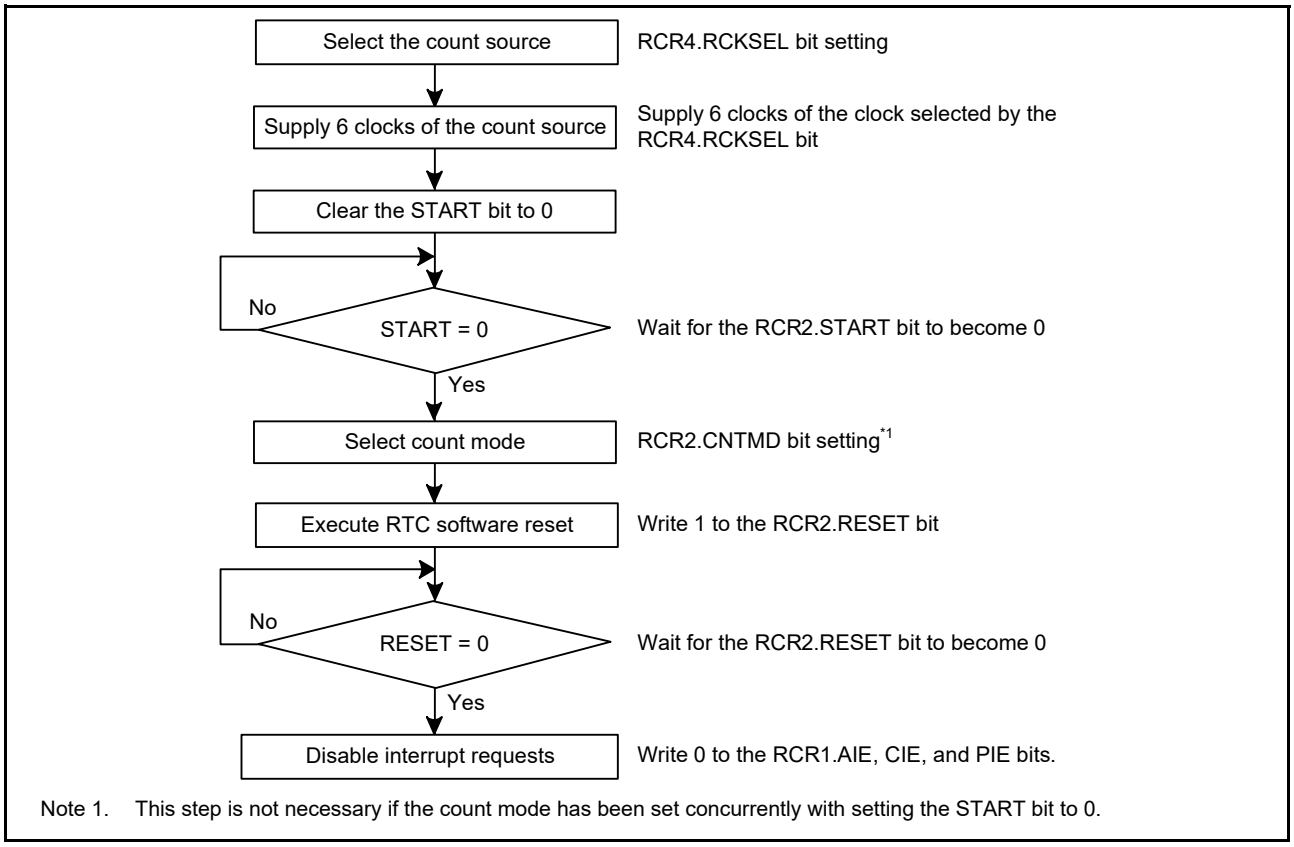
25.6.7 Initialization Procedure when the Realtime Clock is not to be Used

Registers in the RTC are not initialized by a reset. Depending on the initial state, the generation of an unintentional interrupt request or operation of the counter might lead to increased power consumption.

For applications that do not require a realtime clock, initialize the registers by following the initialization procedure shown in Figure 25.14.

Alternatively, when the sub-clock is not used as the system clock or realtime clock, the counter can be stopped by writing 0 (sub-clock oscillator is selected) to the RCR4.RCKSEL bit and stopping the sub-clock. To stop the sub-clock, write 1 to the SOSCCR.SOSTP bit.

For details on the setting of the SOSCCR.SOSTP bit, see section 9, Clock Generation Circuit.



Note 1. This step is not necessary if the count mode has been set concurrently with setting the START bit to 0.

Figure 25.14 Initialization procedure

26. Watchdog Timer (WDT)

26.1 Overview

The Watchdog Timer (WDT) is a 14-bit down-counter and 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. The refresh-permitted period can be set to refresh the counter and to detect when the system runs out of control. Table 26.1 lists the WDT specifications and Figure 26.1 shows the block diagram.

Table 26.1 WDT specifications

Parameter	Specifications
Count source	Peripheral clock (PCLKB)
Clock division ratio	Divide by 4, 64, 128, 512, 2,048, or 8,192
Counter operation	Counting down using a 14-bit down-counter
Conditions 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 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.

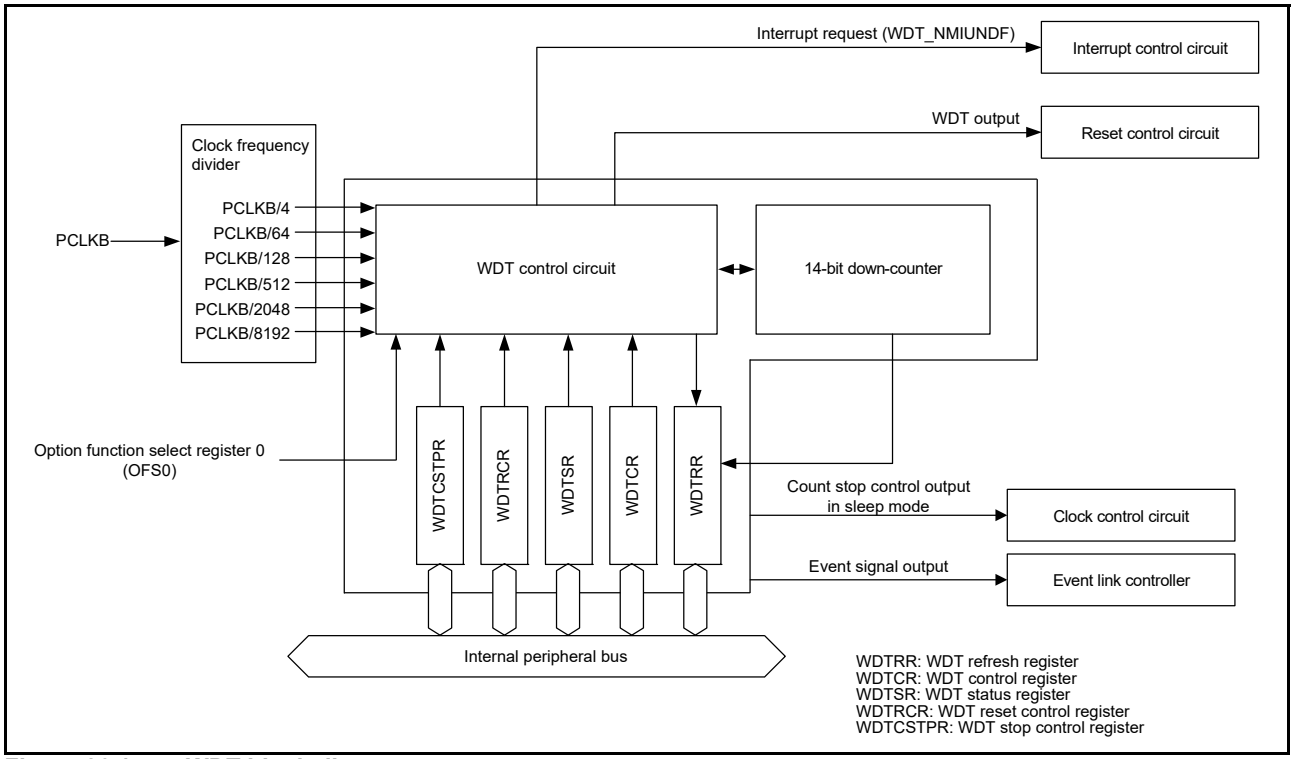


Figure 26.1 WDT block diagram

26.2 Register Descriptions

26.2.1 WDT Refresh Register (WDTRR)

Address(es): WDT.WDTRR 4004 4200h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

Bit	Description	R/W
b7 to b0	The down-counter is refreshed by writing 00h and then writing FFh 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 00h and then writing FFh to WDTRR (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 selection bits (WDTCR.TOPS[1:0]) in the WDT Control Register.

When 00h is written, the read value is 00h. When a value other than 00h is written, the read value is FFh. For details of the refresh operation, see section 26.3.3, Refresh Operation.

26.2.2 WDT Control Register (WDTCR)

Address(es): WDT.WDTCR 4004 4202h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	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	Bit name	Description	R/W
b1, b0	TOPS[1:0]	Timeout Period Selection	b1 b0 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh).	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b4	CKS[3:0]	Clock Division Ratio Selection	b7 b4 0 0 0 1: PCLKB/4 0 1 0 0: PCLKB/64 1 1 1 1: PCLKB/128 0 1 1 0: PCLKB/512 0 1 1 1: PCLKB/2048 1 0 0 0: PCLKB/8192. Other setting are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Selection	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified)	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b13, b12	RPSS[1:0]	Window Start Position Selection	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified)	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

There are some restrictions in writing to the WDTCR register. For details, see section 26.3.2, Control over Writing 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 for the OFS0 register. For details, see section 26.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

TOPS[1:0] bits (Timeout Period Selection)

These bits select the timeout period (the period until the down-counter underflows) from 1024, 4096, 8192, and 16384 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.

After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of PCLKB cycles) until the counter underflows.

Table 26.2 lists the relationship between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLKB cycles.

Table 26.2 Timeout period settings

CKS[3:0] bits				TOPS[1:0] bits				
b7	b6	b5	b4	b1	b0	Clock division ratio	Timeout period (Number of cycles)	PCLKB clock cycles
0	0	0	1	0	0	PCLKB/4	1,024	4,096
				0	1		4,096	16,384
				1	0		8,192	32,768
				1	1		16,384	65,536
0	1	0	0	0	0	PCLKB/64	1,024	65,536
				0	1		4,096	262,144
				1	0		8,192	524,288
				1	1		16,384	1,048,576
1	1	1	1	0	0	PCLKB/128	1,024	131,072
				0	1		4,096	524,288
				1	0		8,192	1,048,576
				1	1		16,384	2,097,152
0	1	1	0	0	0	PCLKB/512	1,024	524,288
				0	1		4,096	2,097,152
				1	0		8,192	4,194,304
				1	1		16,384	8,388,608
0	1	1	1	0	0	PCLKB/2048	1,024	2,097,152
				0	1		4,096	8,388,608
				1	0		8,192	16,777,216
				1	1		16,384	33,554,432
1	0	0	0	0	0	PCLKB/8192	1,024	8,388,608
				0	1		4,096	33,554,432
				1	0		8,192	67,108,864
				1	1		16,384	134,217,728

CKS[3:0] bits (Clock Division Ratio Selection)

These bits specify the division ratio of the clock used for the down-counter. The division ratio can be selected from the peripheral clock (PCLKB) divided by 4, 64, 128, 512, 2048, and 8,192. Combined with the TOPS[1:0] bit setting, a count period between 4,096 and 134,217,728 cycles of the PCLKB clock can be selected for the WDT.

RPES[1:0] bits (Window End Position Selection)

These 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. The selected window end position should be a value smaller than the value for the window start position (that is, window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.

RPSS[1:0] bits (Window Start Position Selection)

These 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 end position. The window start position should be set to a value greater than the value for the window end position. If the window start position is set to a value smaller than or equal to the window end position, the window end position is set to 0%.

Table 26.3 lists the counter values for the window start and end positions and Figure 26.2 shows the refresh-permitted period set by the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 26.3 Relationship between Timeout Period and Window Start and End Counter Values

TOPS[1:0] Bits		Timeout Period		Window Start and End Counter Value			
		Cycles	Counter Value	100%	75%	50%	25%
0	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
0	1	4096	0FFFh	0FFFh	0BFFh	07FFh	03FFh
1	0	8192	1FFFh	1FFFh	17FFh	0FFFh	07FFh
1	1	16384	3FFFh	3FFFh	2FFFh	1FFFh	0FFFh

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 setting ≥ window start setting, the window end setting is set to 0%.

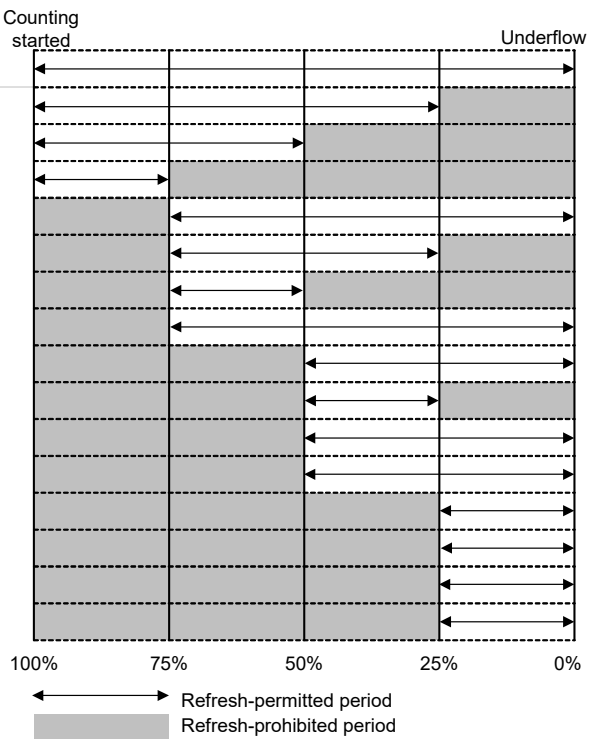


Figure 26.2 RPSS[1:0] and RPES[1:0] bit settings and refresh-permitted period

26.2.3 WDT Status Register (WDTSR)

Address(es): WDT.WDTSR 4004 4204h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Down-Counter Value	Value counted by the down-counter	R
b14	UNDFF	Underflow Flag	0: No underflow occurred 1: Underflow occurred.	R(/W) *1
b15	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.

CNTVAL[13:0] bits (Down-Counter Value)

Read these bits to confirm the value of the down-counter. The read value might differ from the actual count by a value of one count.

UNDFF flag (Underflow Flag)

Read this 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+1) PCLKB cycles. In addition, clearing of this 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] = 0001b, N = 4
- When WDTCR.CKS[3:0] = 0100b, N = 64
- When WDTCR.CKS[3:0] = 1111b, N = 128
- When WDTCR.CKS[3:0] = 0110b, N = 512
- When WDTCR.CKS[3:0] = 0111b, N = 2048
- When WDTCR.CKS[3:0] = 1000b, N = 8192

REFEF flag (Refresh Error Flag)

Read this flag to confirm whether a refresh error occurred. 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+1) PCLKB cycles. In addition, clearing of this 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] = 0001b, N = 4
- When WDTCR.CKS[3:0] = 0100b, N = 64
- When WDTCR.CKS[3:0] = 1111b, N = 128
- When WDTCR.CKS[3:0] = 0110b, N = 512
- When WDTCR.CKS[3:0] = 0111b, N = 2048
- When WDTCR.CKS[3:0] = 1000b, N = 8192

26.2.4 WDT Reset Control Register (WDTRCR)

Address(es): WDT.WDTRCR 4004 4206h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTIRQS	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	RSTIRQS	Reset Interrupt Request Selection	0: Non-maskable interrupt request or interrupt request output is enabled 1: Reset output is enabled.	R/W

There are some restrictions with writing to the WDTRCR register. For details, see section 26.3.2, Control over Writing 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 WDTCR register can also be made for the OFS0 register. For details, see section 26.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

26.2.5 WDT Count Stop Control Register (WDTCSTPR)

Address(es): WDT.WDTCSTPR 4004 4208h

	b7	b6	b5	b4	b3	b2	b1	b0
	SLCSTP	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SLCSTP	Sleep-Mode Count Stop Control	0: Count stop is disabled 1: Count is stopped when transition to Sleep mode.	R/W

The WDTCSTPR register controls whether to stop the WDT counter in a low power consumption state. There are some restrictions with writing to the WDTCSTPR register. For details, see section 26.3.2, Control over Writing 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 26.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

SLCSTP bit (Sleep-Mode Count Stop Control)

This bit selects whether to stop counting when transition to Sleep mode.

26.2.6 Option Function Select Register 0 (OFS0)

For details on the OFS0 register, see section 26.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

26.3 Operation

26.3.1 Count Operation in Each Start Mode

The WDT has two start modes: auto-start mode, in which counting automatically starts after release from the reset state, and register start mode, in which counting starts with a refresh by writing to the register.

In auto-start mode, counting automatically starts after release from the reset state in accordance with the settings in the Option Function Select Register 0 (OFS0) in the flash.

In register start mode, counting start with a refresh by writing to the register after the respective registers are set after 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.

26.3.1.1 Register Start Mode

When the WDT start mode select bit (OFS0.WDTSTRT) is 1, register start mode is selected 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 to Sleep mode in the WDTCSTPR register:

- Clock division ratio
- Window start and end positions
- Timeout period in the WDTCR register
- Reset output or interrupt request output in the WDTRCR register
- Counter stop control during transitions.

Refresh the down-counter to start counting down from the value set by the timeout period selection bits (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 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 a reset signal or a non-maskable interrupt request/interrupt request (WDT_NMIUNDF). Reset output or interrupt request output can be selected by setting the WDT reset interrupt request selection bit (WDTRCR.RSTIRQS). Non-maskable interrupt request or interrupt request can be selected by setting the WDT Underflow/Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 26.3 shows an example of operation under the following conditions:

- Register start mode (OFS0.WDTSTRT = 1)
- Reset output is enabled (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).

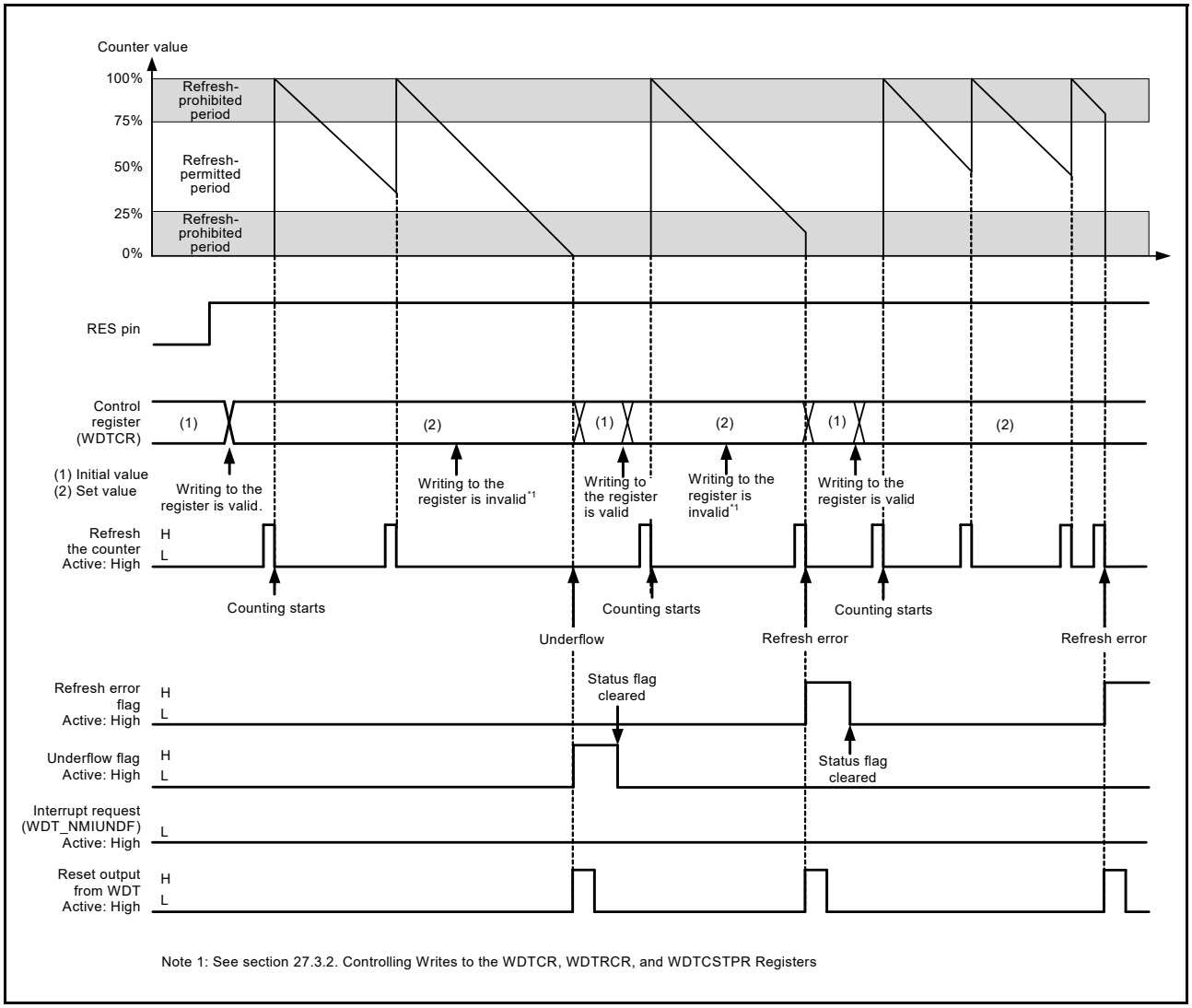


Figure 26.3 Operation example in register start mode

26.3.1.2 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 while the settings in the OFS0 register are enabled.

Within the reset state, the setting values (clock division ratio, window start and end positions, timeout period, and reset output or interrupt request), and counter stop control at transitions to sleep mode of option function select register 0 (OFS0) are set in the WDT registers.

When the reset state is released, the down-counter automatically starts counting down from the value set by 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 as long as 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 a 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 one 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 by setting the WDT Underflow/Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 26.4 shows an example of operation (non-maskable interrupt) under the following conditions.

- Auto start mode (OFS0.WDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.WDTRSTIRQS = 0)
- The window start position is 75% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b).

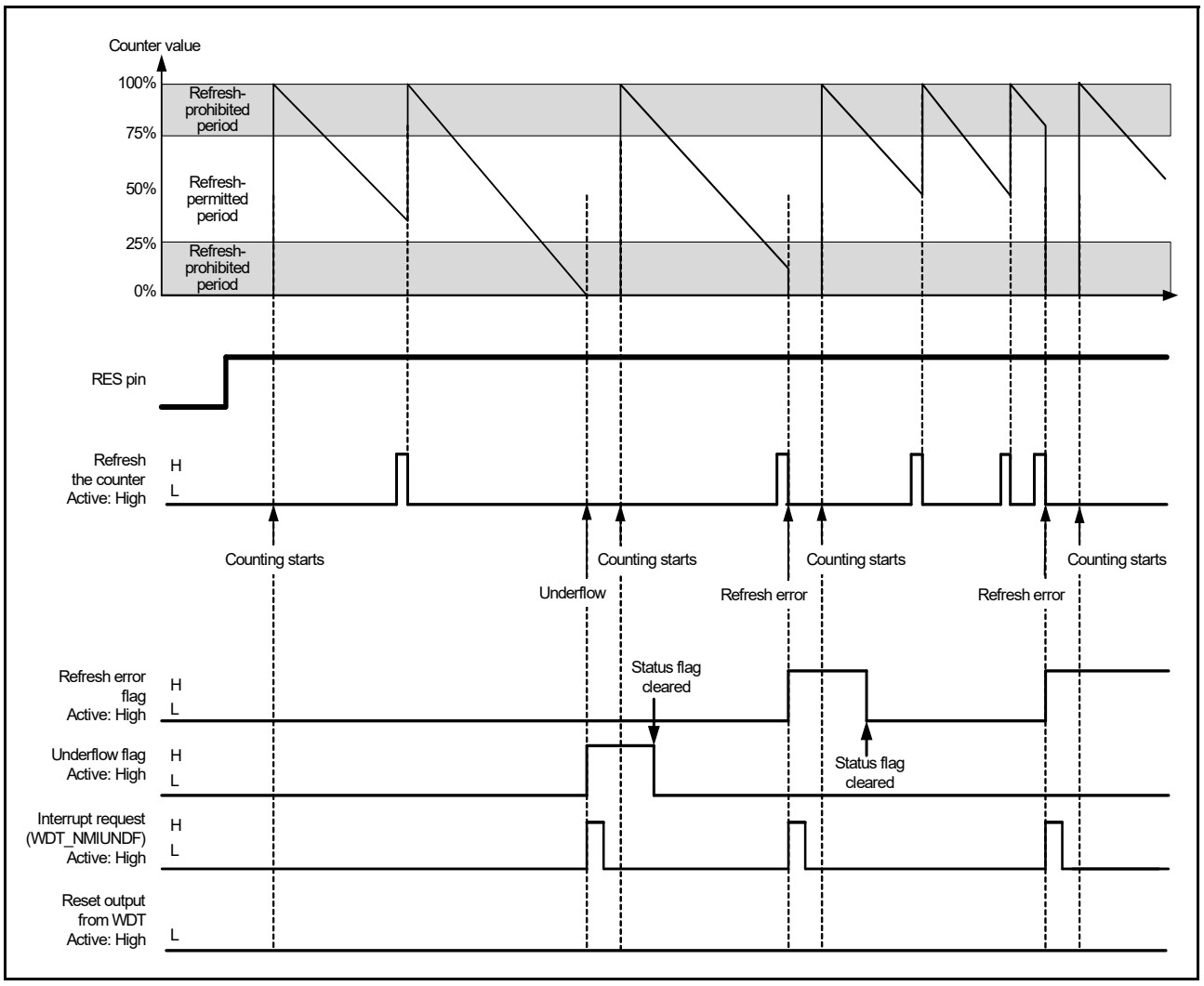


Figure 26.4 Operation example in auto-start mode

26.3.2 Control over Writing 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 between the release from the reset state and the first refresh operation.

After a refresh, (counting starts) or writing to WDTCR, WDTRCR or WDTCSTPR, the protection signal in the WDT becomes 1 to protect WDTCR, WDTRCR and WDTCSTPR against subsequent writing attempts. This protection is released by a reset source of the WDT. With other reset sources, the protection is not released.

Figure 26.5 shows control waveforms produced in response to writing to the WDTCR.

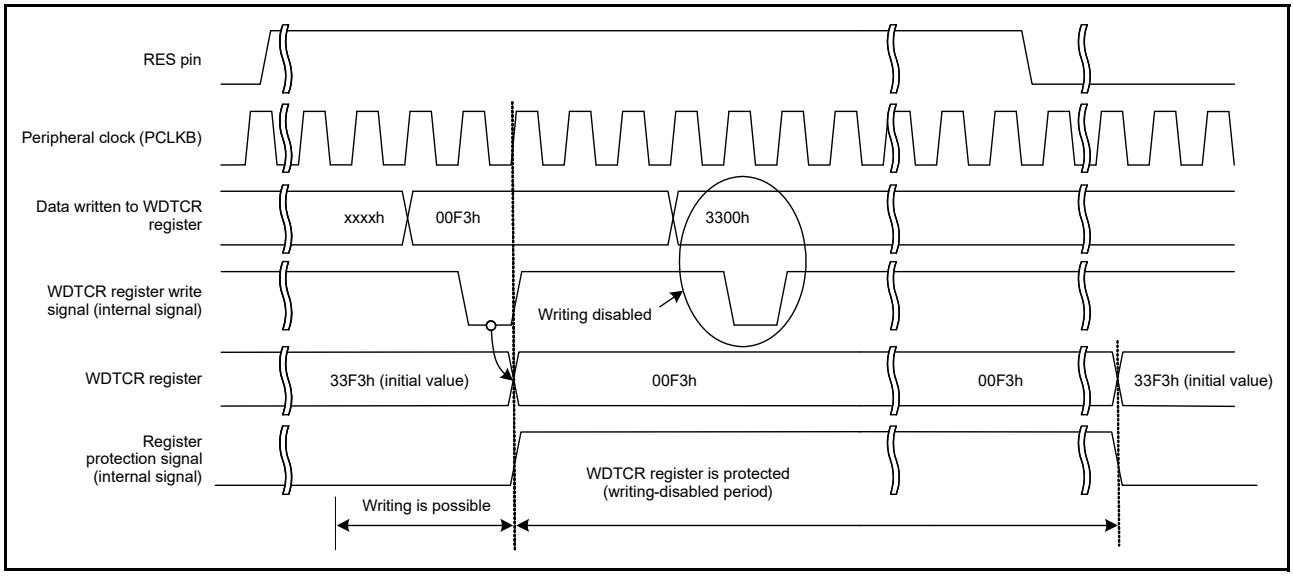


Figure 26.5 Control waveforms produced in response to writing to WDTCR register

26.3.3 Refresh Operation

The down-counter is refreshed by writing the values 00h and FFh to the WDT Refresh Register (WDTRR). If a value other than FFh is written after 00h, the down-counter is not refreshed. After an invalid value is written, correct refreshing resumes by writing to 00h and FFh to the WDTRR Register.

When a register other than WDTRR is accessed or WDTRR is read between writing 00h and writing FFh to WDTRR, correct refreshing is performed.

Writing to refresh the counter must be performed within the refresh-permitted period and whether this is done is determined by writing FFh. For this reason, correct refreshing is performed even when 00h is written outside the refresh-permitted period.

[Sample sequences of writing that are valid when refreshing the counter]

- 00h → FFh
- 00h (n–1-th time) → 00h (nth time) → FFh
- 00h → access to another register or read from WDTRR → FFh.

[Sample sequences of writing that are not valid when refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h→ AAh (00h and a value other than FFh) → FFh.

After FFh is written to the WDT Refresh Register (WDTRR), refreshing the down-counter requires up to 4 cycles of the signal for counting. Therefore, writing FFh to the WDTRR should be completed four cycle counts before the down-counter underflows.

Figure 26.6 shows the WDT refresh-operation waveforms when the clock division ratio = PCLKB/64.

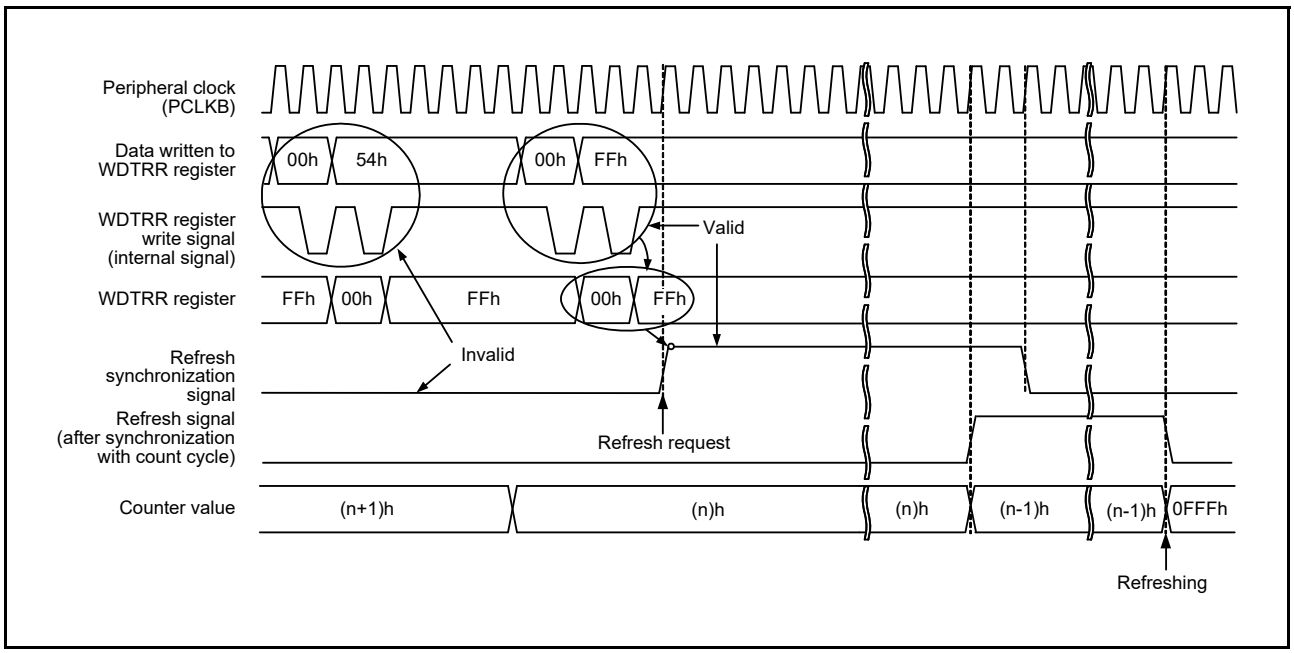


Figure 26.6 WDT refresh operation waveforms (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

26.3.4 Reset Output

When the reset interrupt selection 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 one 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 restarts, the counter is set up and counting down starts again with a refresh. In auto-start mode, counting down automatically starts after the reset state is released.

26.3.5 Interrupt Sources

When the reset interrupt selection 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 14, Interrupt Controller Unit (ICU).

Table 26.4 WDT interrupt sources

Name	Interrupt source	DTC activation
WDT_NMIUNDF	Down-counter underflow Refresh error	Not possible

26.3.6 Reading Down-Counter Value

The WDT stores the counter value in the down-counter value (WDTSR.CNTVAL[13:0]) bits of the WDT Status Register. Therefore, the counter value can be checked with the WDTSR.CNTVAL[13:0] bits.

Figure 26.7 shows the processing for reading the WDT down-counter value when the clock division ratio = PCLKB/64.

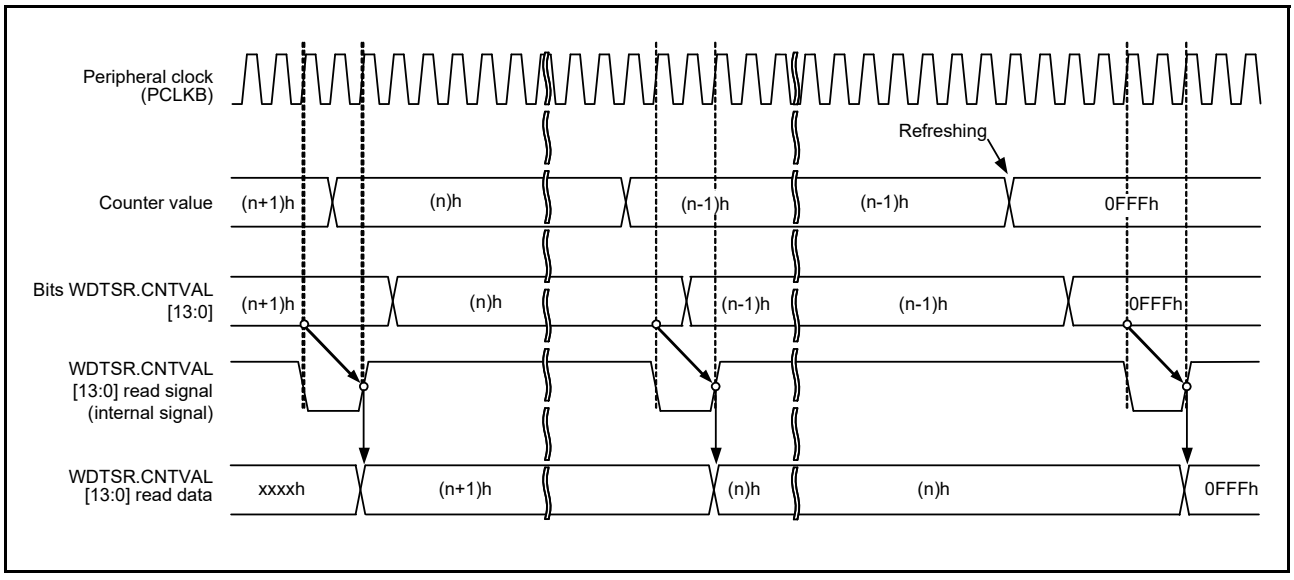


Figure 26.7 Processing for reading WDT down-counter value (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

26.3.7 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Table 26.5 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 7.2.1, Option Function Select Register 0 (OFS0).

Table 26.5 Association between Option Function Select Register 0 (OFS0) and WDT Registers

Target of control	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	Reset output or interrupt request output selection	OFS0.WDTRSTIRQS	WDTRCR.RSTIRQS
Count stop	Sleep-mode count stop control	OFS0.WDTSTPCTL	WDTCSTPR.SLCSTP

26.4 Link Operation by 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 setting of the reset interrupt request selection bit (WDTRCR.RSTIRQS) in register start mode or auto-start mode. An event signal can also be output when the next interrupt source is generated and while the refresh error flag (WDTSR.REFEF) or underflow flag (WDTSR.UNDFF) is 1. For details, see section 19, Event Link Controller (ELC).

26.5 Usage Notes

26.5.1 ICU Event Link Setting Register n (IELSRn) setting

Setting 47h to the ICU Event Link Setting Register n (IELSRn.IELS[7:0]) is prohibited when enabling the WDT reset assertion (OFS0.WDTRSTIRQS = 1 or WDTRCR.RSTIRQS = 1) or when enabling the event link operation (47h is set to ELSRm.ELS[7:0]).

27. Independent Watchdog Timer (IWDT)

27.1 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/ interrupt on a timer underflow. Because the timer operates using an independent, dedicated clock source, it is particularly useful in returning the MCU to a known state as a failsafe mechanism when the system runs out of control. The watchdog timer can be triggered automatically by a reset, underflow, or refresh error, or a refresh of the count value in the registers.

The functions of the IWDT are different from those of the WDT in the following respects:

- The divided IWDT-dedicated clock (IWDTCLK) is used as the count source (not affected by the PCLKB)
- IWDT does not support the register start mode
- When transitioning to low power mode, the OFS0.IWDTSTPCTL bit can be used to select whether to stop the counter or not.

Table 27.1 lists the IWDT specifications and Figure 27.1 shows the block diagram.

Table 27.1 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 (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.
Auto-start mode	• Selecting the clock frequency division ratio after a reset (OFS0.IWDTCKS[3:0] bits) • Selecting the timeout period of the Independent Watchdog Timer (OFS0.IWDTTOPS[1:0] bits) • Selecting the window start position in the Independent Watchdog Timer (OFS0.IWDTRPSS[1:0] bits) • Selecting the window end position in the Independent Watchdog Timer (OFS0.IWDTRPES[1:0] bits) • Selecting the Reset Output or Interrupt Request Output (OFS0.IWDTRSTIRQS bit) • Selecting the down-count stop function at transition to Sleep mode, Software Standby mode, or Snooze 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).

To use the IWDT, the IWDT-dedicated clock (IWDTCLK) should be supplied. The bus interface and registers operate with PCLKB, and the 14-bit counter and control circuits operate with IWDTCLK.

Figure 27.1 shows a block diagram of the IWDT.

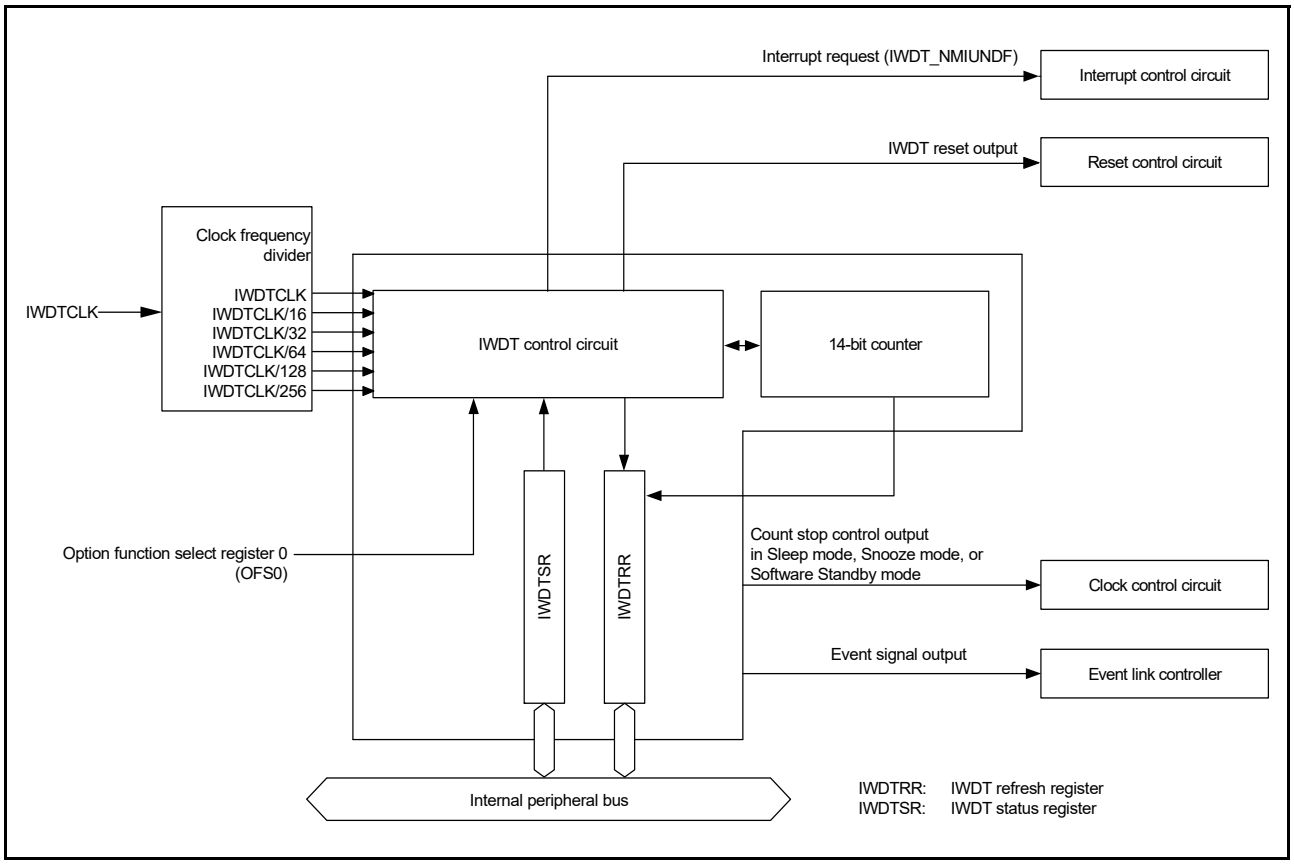


Figure 27.1 IWDT block diagram

27.2 Register Descriptions

27.2.1 IWDT Refresh Register (IWDTRR)

Address(es): IWDT.IWDTRR 4004 4400h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

Bit	Description	R/W
b7 to b0	The counter is refreshed by writing 00h and then writing FFh 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 00h and then writing FFh to IWDTRR (refresh operation) within the refresh-permitted period. After the down-counter is refreshed, it starts counting down from the value selected with the IWDT timeout period select bits (OFS0.IWDTTOPS[1:0]) in the Option Function Select Register 0 (OFS0).

When 00h is written, the read value is 00h. When a value other than 00h is written, the read value is FFh. For details of the refresh operation, see section 27.3.2, Refresh Operation.

27.2.2 IWDT Status Register (IWDTSR)

Address(es): IWDT.IWDTSR 4004 4404h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit name	Description	R/W
b13 to b0	CNTVAL[13:0]	Counter Value	Value counted by the down-counter	R
b14	UNDFF	Underflow Flag	0: Underflow not occurred 1: Underflow occurred.	R/(W)*1
b15	REFEF	Refresh Error Flag	0: Refresh error not occurred 1: Refresh error occurred.	R/(W)*1

Note 1. Only 0 can be written to clear the flag.

CNTVAL[13:0] bits (Counter Value)

Read these bits to confirm the value of the down-counter, but note that the read value might differ from the actual count by 1.

UNDFF flag (Underflow Flag)

Read this 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 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 IWDTCKS[3:0] = 0000b, N = 1
- When IWDTCKS[3:0] = 0010b, N = 16
- When IWDTCKS[3:0] = 0011b, N = 32
- When IWDTCKS[3:0] = 0100b, N = 64
- When IWDTCKS[3:0] = 1111b, N = 128
- When IWDTCKS[3:0] = 0101b, N = 256

REFEF flag (Refresh Error Flag)

Read this flag to confirm whether a refresh error occurred. 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 the REFEF flag takes (N+2) IWDTCLK cycles and 2 PCLKB cycles. In addition, clearing of the flag is ignored for (N+2) IWDTCLK cycles after a refresh error. This number is specified in the IWDTCKS[3:0] bits as follows:

- When IWDTCKS[3:0] = 0000b, N = 1
- When IWDTCKS[3:0] = 0010b, N = 16
- When IWDTCKS[3:0] = 0011b, N = 32
- When IWDTCKS[3:0] = 0100b, N = 64
- When IWDTCKS[3:0] = 1111b, N = 128
- When IWDTCKS[3:0] = 0101b, N = 256

27.2.3 Option Function Select Register 0 (OFS0)

For information on the Option Function Select Register 0 (OFS0), see section 7.2.1, Option Function Select Register 0 (OFS0).

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

These 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 by the IWDTCKS[3:0] bits as one cycle.

After the down-counter is refreshed, the combination of the IWDTCKS[3:0] and IWDTTOPS[1:0] bits determines the time, that is, the number of IWDTCLK cycles until the counter underflows. Table 27.2 lists the relationship between the IWDTCKS[3:0] and IWDTTOPS[1:0] bit setting, the timeout period, and the number of IWDTCLK cycles.

Table 27.2 Timeout period settings

IWDTCKS[3:0] bits				IWDTTOPS[1:0] bits				
b7	b6	b5	b4	b1	b0	Clock Division Ratio	Timeout period (number of cycles)	Cycles of IWDTCLK
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)

These 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. Combination with the IWDTTOPS[1:0] bit setting, a count period between 128 and 524288 cycles of the IWDTCLK clock can be selected for the IWDT.

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

These 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. The selected window end position should be a value smaller than the window start position. If the window end position is greater than the window start position, only the window start position setting is enabled.

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

These 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. The window start position should be a value greater than the window end position. If the window start position is smaller than or equal to the window end position, the window end position is set to 0%.

Table 27.3 lists the counter values for the window start and end positions and Figure 27.2 shows the refresh-permitted period set by the IWDTRPSS[1:0], IWDTRPES[1:0], and IWDTTOPS[1:0] bits.

Table 27.3 Relationship between timeout period and window start and end counter values

IWDTTOPS[1:0] bits		Timeout period		Window start and end counter value			
b1	b0	Cycles	Counter value	100%	75%	50%	25%
0	0	128	007Fh	007Fh	005Fh	003Fh	001Fh
0	1	512	01FFh	01FFh	017Fh	00FFh	007Fh
1	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
1	1	2048	07FFh	07FFh	05FFh	03FFh	01FFh

IWDTRPSS[1:0] bits		IWDTRPES[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 setting ≥ window start setting, the window end setting is set to 0%.

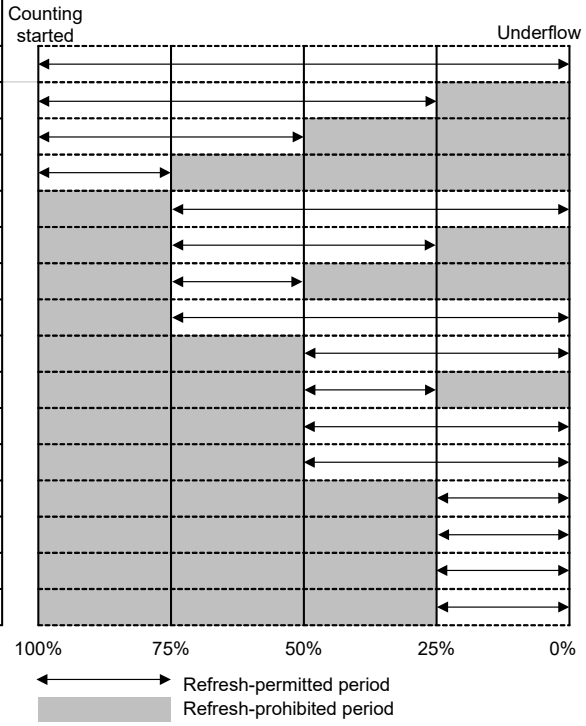


Figure 27.2 IWDTRPSS[1:0] and [IWDTRPES[1:0] bit settings and refresh-permitted period

IWDTRSTIRQS bit (IWDT Reset Interrupt Request Select)

This bit specifies the behavior when an underflow or a refresh error occurred. The value 1 indicates that reset output is selected. The value 0 indicates that a non-maskable interrupt/interrupt is selected.

IWDTSTPCTL bit (IWDT Stop Control)

This bit selects whether the stop counting at a transition to Sleep mode, Snooze mode, or Software Standby mode.

27.3 Operation

27.3.1 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 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
- Window start and end positions
- Timeout period
- Reset output or interrupt request
- Counter stop control at transitions to low power mode.

When the reset state is released, the counter automatically starts counting down from the value selected by 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 counting down 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 due to an attempt 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 one cycle, and restarts the count. The reset output or interrupt request can be selected by setting the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS). Non-maskable interrupt request or interrupt request can be selected by setting the IWDT Underflow/Refresh Error Interrupt Enable bit (NMIER.IWDTEN).

Figure 27.3 shows an example of operation under the following conditions:

- Auto-start mode (OFS0.IWDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.IWDTRSTIRQS = 0)
- The window start position is 75% (OFS0.IWDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.IWDTRPES[1:0] = 10b).

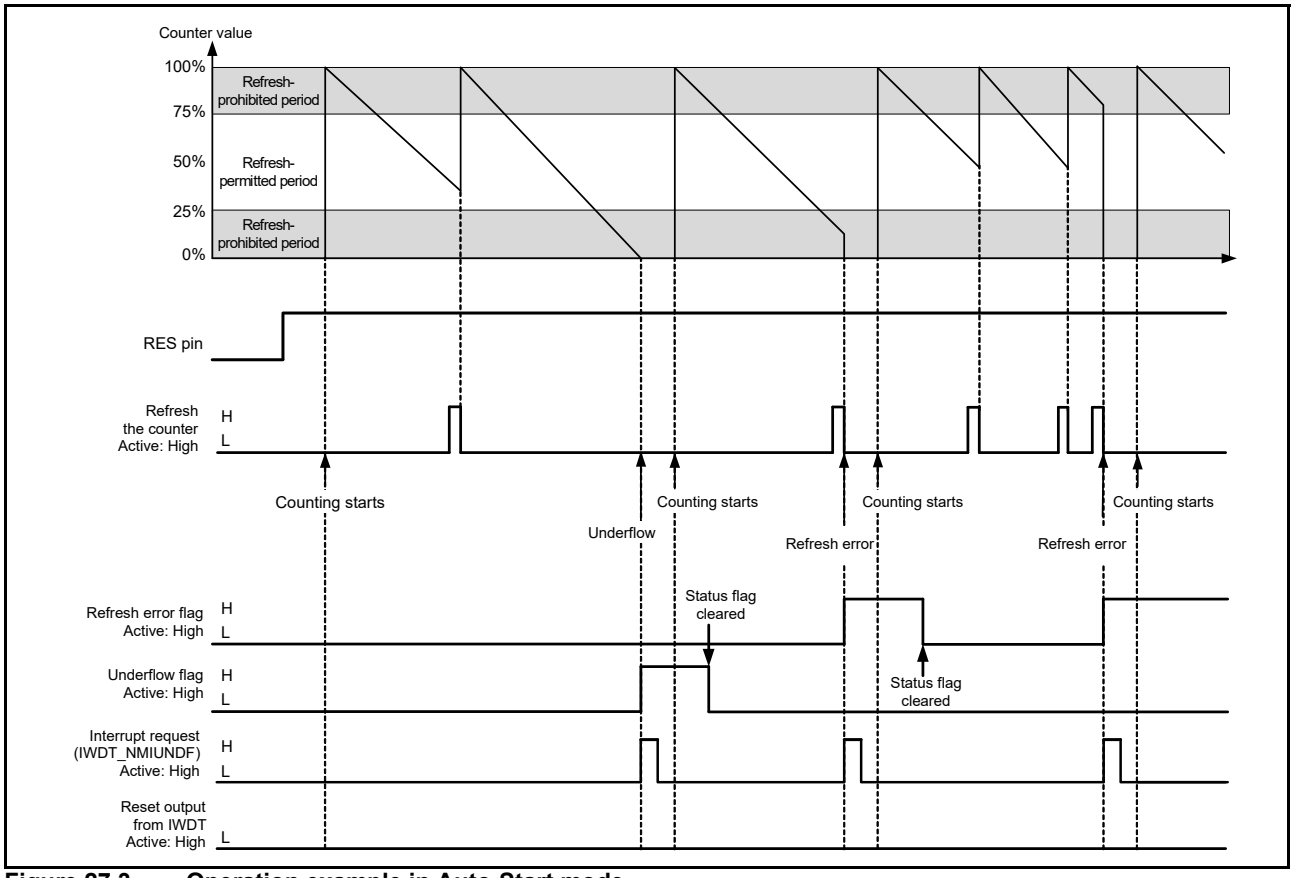


Figure 27.3 Operation example in Auto-Start mode

27.3.2 Refresh Operation

The down-counter is refreshed by writing the values 00h and FFh to the IWDT Refresh Register (IWDTRR). If a value other than FFh is written after 00h, the counter is not refreshed. If an invalid value is written, correct refreshing is performed by writing 00h and FFh to the IWDT Refresh Register (IWDTRR) again.

When writing is done in the order of 00h (first time) → 00h (second time), and if FFh is written after that, the writing order 00h → FFh is satisfied. Writing 00h (n–1th time) → 00h (nth time) → FFh is valid and the correct refresh is done. When the first value written before 00h is not 00h, correct refreshing is done if the operation contains the sequence of writing 00h → FFh. Additionally, correct refreshing is done regardless of whether a register other than IWDTRR is accessed or IWDTRR is read between writing 00h and writing FFh to IWDTRR.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h → FFh
- 00h (n–1th time) → 00h (nth time) → FFh
- 00h → access to another register or read from IWDTRR → FFh.

[Sample sequences of writing that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h → AAh (00h and a value other than FFh) → FFh.

When 00h is written to IWDTRR outside the refresh-permitted period, if FFh is written to IWDTRR in the refresh-permitted period, the writing sequence is valid and refreshing is done.

After FFh 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]) determine how many cycles of the IWDT-dedicated clock (IWDTCLK) make up one cycle for counting). Therefore, writing FFh to the IWDTRR must be completed four count cycles before the end of the refresh-permitted period or a counter underflow. The value of the counter can be checked with the counter bits (IWDTSR.CNTVAL[13:0]).

[Sample refreshing timings]

- When the window start position is set to 1FFFh, even if 00h is written to IWDTRR before 1FFFh is reached (2002h, for example), refreshing is done if FFh is written to IWDTRR after the value of the IWDTSR.CNTVAL[13:0] bits has reached 1FFFh.
- When the window end position is set to 1FFFh, refreshing is done if 2003h (four count cycles before 1FFFh) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to IWDTRR.
- When the refresh-permitted period continues until count 0000h, refreshing can be done immediately before an underflow. In this case, if 0003h (four count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to IWDTRR, no underflow occurs and refreshing is done.

Figure 27.4 shows the IWDT refresh-operation waveforms when PCLKB > IWDTCLK and the clock division ratio is IWDTCLK.

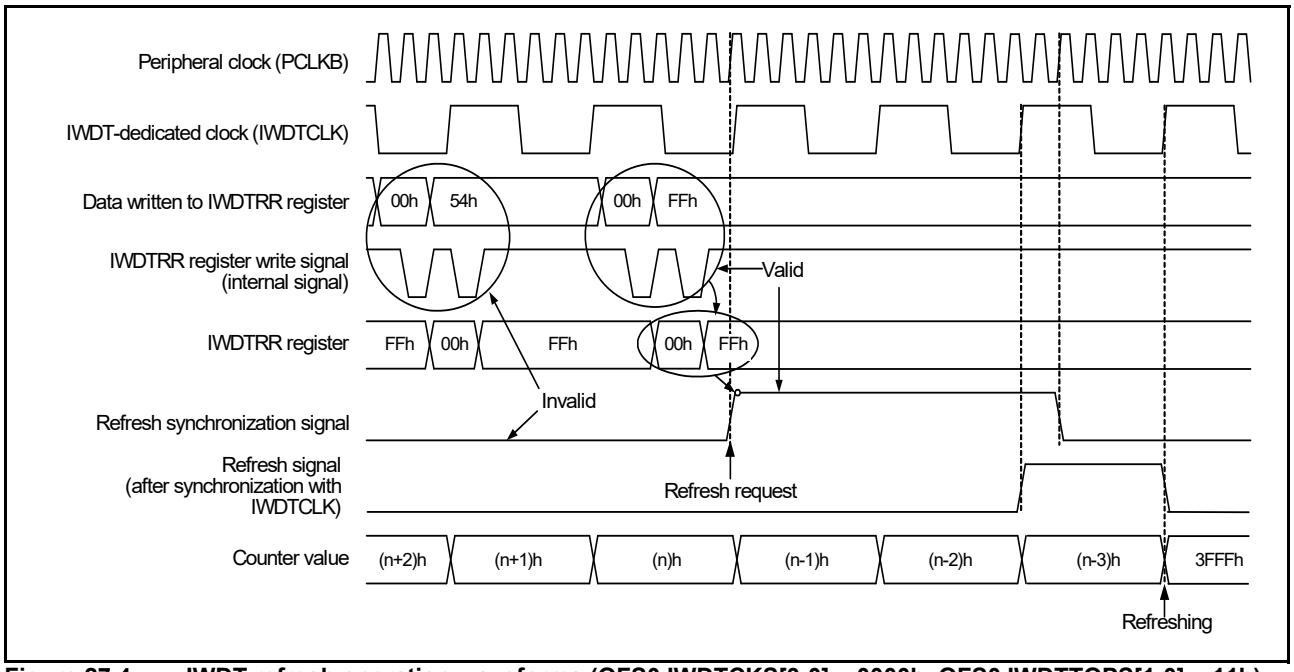


Figure 27.4 IWDT refresh operation waveforms (OFS0.IWDTCKS[3:0] = 0000b, OFS0.IWDTTOPS[1:0] = 11b)

27.3.3 Status Flags

The refresh error (IWDTSR.REFEF) and underflow (IWDTSR.UNDFF) flags retain the source of the reset signal output from the IWDT or the source of the interrupt request from the IWDT. Therefore, after a release from the reset state or interrupt request generation, read the IWDTSR.REFEF and UNDFF flags to check for the reset or 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 reset or interrupt request from the IWDT, the earlier reset or interrupt source is cleared and the new reset or interrupt source is written. After 0 is written to each flag, up to three IWDTCLK cycles and two PCLKB cycles are required before the value is reflected.

27.3.4 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.

27.3.5 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 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 14, Interrupt Controller Unit (ICU).

Table 27.4 IWDT interrupt source

Name	Interrupt source	DTC activation
IWDT_NMIUNDF	Down-counter underflow Refresh error	Not possible

27.3.6 Reading the Down-counter Value

As the counter in 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. Thus, the counter value can be checked indirectly through the IWDTSR.CNTVAL[13:0] bits.

Reading the counter value requires multiple PCLKB clock cycles (up to four clock cycles), and the read counter value might differ from the actual counter value by a value of one count.

Figure 27.5 shows the processing for reading the IWDT counter value when PCLKB > IWDTCLK and the clock division ratio is IWDTCLK.

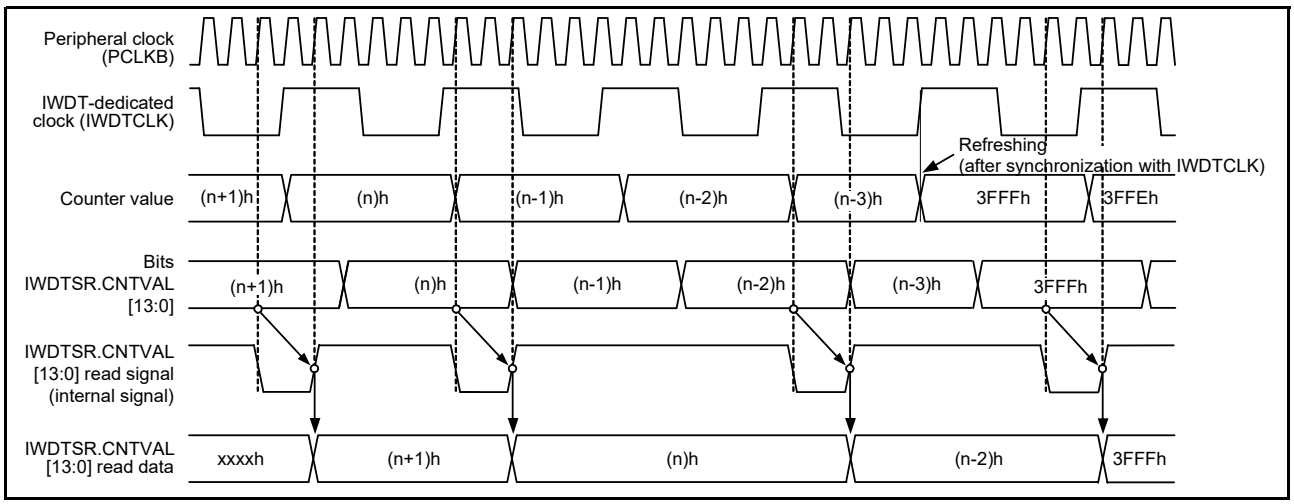


Figure 27.5 Processing for reading IWDT counter value (OFS0.IWDTCKS[3:0] = 0000b, OFS0.IWDTTOPS[1:0] = 11b)

27.4 Link Operation by 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 and refresh error.

An event signal is output regardless of the setting of the OFS0.WDTRSTIRQS 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 19, Event Link Controller (ELC).

27.5 Usage Notes

27.5.1 Refresh Operations

While configuring the refresh time, consider variations in the range of errors due to the accuracy of PCLKB and IWDTCLK and set values to ensure refreshing is possible.

27.5.2 Clock Division Ratio Setting

Satisfy the frequency of the peripheral module clock (PCLKB) ≥ 4 × (the frequency of the count clock source after division).

28. USB 2.0 Full-Speed Module (USBFS)

28.1 Overview

The MCU provides a USB 2.0 Full-Speed module (USBFS) that operates as a host or device controller compliant with the Universal Serial Bus (USB) specification revision 2.0. The host controller supports USB 2.0 full-speed and low-speed transfers, and the device controller supports USB 2.0 full-speed transfers. The USBFS has an internal USB transceiver and supports all of the transfer types defined in the USB 2.0 specification.

The USBFS has FIFO buffer for data transfers, providing a maximum of 10 pipes. Any endpoint number can be assigned to pipes 1 to 9, based on the peripheral devices or the communication requirements for your system.

The MCU supports revision 1.2 of the Battery Charging specification. Because the MCU can be powered at 5 V, the USB LDO regulator provides the internal USB transceiver power supply 3.3 V.

Table 28.1 lists the USBFS specifications, Figure 28.1 shows the block diagram, and Table 28.2 lists the I/O pins.

Table 28.1 USBFS specifications

Parameter	Specifications
Features	• USB Device Controller (UDC) and USB 2.0 transceiver supporting host controller, device controller, and On-The-Go (OTG) functions (one channel) • Host and device controller can be switched by the software • Self-power or bus power mode selectable • Revision 1.2 of battery charging specification is supported • The USB LDO regulator is used to power the internal USB transceiver.
	Host controller features: • Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) • Automatic scheduling for SOF and packet transmissions • Programmable intervals for isochronous and interrupt transfers.
	Device controller features: • Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) • Control transfer stage control function • Device state control function • Auto response function for SET_ADDRESS request • SOF interpolation.
Communication data transfer type	• Control transfer • Bulk transfer • Interrupt transfer • Isochronous transfer.
Pipe configuration	• FIFO buffer for USB communication • Up to 10 pipes selectable, including the default control pipe • Pipes 1 to 9 assignable to any endpoint number.
	Transfer conditions specifiable for each pipe: • Pipe 0: Control transfer with 64-byte single buffer • Pipes 1 and 2: Selectable to bulk transfer with 64-byte double buffer or isochronous transfer with 256-byte double buffer • Pipes 3 to 5: Bulk transfer with 64-byte double buffer • Pipes 6 to 9: Interrupt transfer with 64-byte single buffer.
Others	• Reception end function using transaction count • Function that changes the BRDY interrupt event notification timing (BFRE) • Automatic clearing of the FIFO buffer after the data for the pipe specified in the DnFIFO port (n = 0, 1) is read (DCLRM) • NAK setting function for response PID generated on transfer end (SHTNAK) • On-chip pull-up and pull-down resistors for USB_DP/USB_DM
Module-stop function	Module-stop state can be set to reduce power consumption

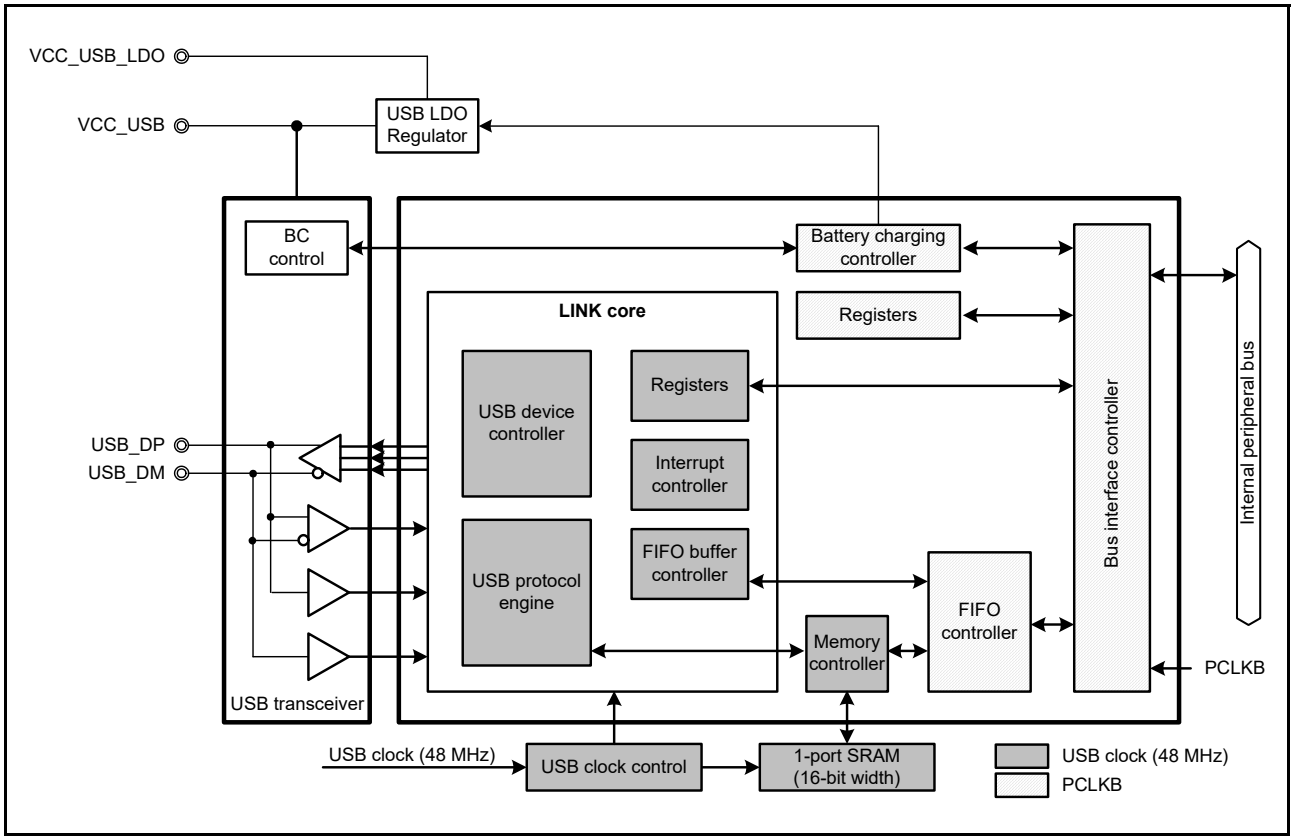


Figure 28.1 USBFS block diagram

Table 28.2 USBFS pin configuration

Port	Pin name	I/O	Function
USBFS	USB_DP	I/O	D+ I/O pin for the on-chip USB transceiver. Must be connected to the D+ data line of the USB bus.
	USB_DM	I/O	D- I/O pin for the on-chip USB transceiver. Must be connected to the D- data line of the USB bus.
	USB_VBUS	Input	USB cable connection monitor pin. Must be connected to VBUS signal on the USB bus. VBUS pin status (connected or disconnected) can be detected when the USBFS is a device controller.*1
	USB_EXICEN	Output	Low power control signal for the OTG power supply IC
	USB_VBUSEN	Output	VBUS (5 V) enable signal for the external power supply IC
	USB_OVRCURA USB_OVRCURB	Input	Overcurrent pins for USBFS. Must be connected to external overcurrent detection signals. When the OTG power supply chip is connected, must be connected to the VBUS comparator signals.
	USB_ID	Input	Must be connected to MicroAB connector ID input signal in OTG mode
Common	VCC_USB	I/O	Input: Power supply for USB transceiver. Output: USB LDO regulator output pin. This pin should be connected to an external capacitor.
	VCC_USB_LDO	Input	Power supply pin for USB LDO regulator
	VSS_USB	Input	USB ground pin

Note 1. P407 is 5-V tolerant.

28.2 Register Descriptions

28.2.1 System Configuration Control Register (SYSCFG)

Address(es): USBFS.SYSCFG 4009 0000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKE	—	CNEN	—	DCFM	DRPD	DPRPU	DMRPU	—	—	USBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	USBE	USBFS Operation Enable	0: Disabled 1: Enabled.	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	DMRPU	D- Line Resistor Control*1	0: Disable line pull-up 1: Enable line pull-up.	R/W
b4	DPRPU	D+ Line Resistor Control*1	0: Disable line pull-up 1: Enable line pull-up.	R/W
b5	DRPD	D+/D- Line Resistor Control	0: Disable line pull-down 1: Enable line pull-down.	R/W
b6	DCFM	Controller Function Select	0: Select device controller 1: Select host controller.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	CNEN	CNEN Single-Ended Receiver Enable	0: Disable single-ended receiver 1: Enable single-ended receiver.	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	SCKE	USB Clock Enable*2	0: Stop clock supply to the USBFS 1: Enable clock supply to the USBFS.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not enable the DMRPU and DPRPU bits at the same time.
Note 2. After writing 1 to the SCKE bit, read it to confirm that it is set to 1.

USBE bit (USBFS Operation Enable)

The USBE bit enables or disables operation of the USBFS.

Changing the USBE bit from 1 to 0 initializes the bits listed in Table 28.3. Only change this bit while the SCKE bit is 1. In host controller mode, this bit must be set to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] bit chattering, and confirming that the USB bus state is stable.

Table 28.3 Registers initialized by writing 0 to SYSCFG.USBE bit

Selected function	Register	Bit	Remarks
Device controller	SYSSTS0	LNST[1:0]	Value is saved in host controller mode
	DVSTCTR0	RHST[2:0]	-
	INTSTS0	DVSQ[2:0]	Value is saved in host controller mode
	USBREQ	BREQUEST[7:0], BMREQUESTTYPE[7:0]	Value is saved in host controller mode
	USBVAL	WVALUE[15:0]	Value is saved in host controller mode
	USBINDX	WINDEX[15:0]	Value is saved in host controller mode
	USBLENG	WLENTUH[15:0]	Value is saved in host controller mode
Host controller	DVSTCTR0	RHST[2:0]	-
	FRMNUM	FRNM[10:0]	Value is saved in device controller mode

DMRPU bit (D- Line Resistor Control*1)

The DMRPU bit enables or disables pulling up the D- line in device controller mode.

When the DMRPU bit is set to 1 in device controller mode, the USBFS pulls up the D- line to notify the USB host that it attached as a low-speed device. Changing the DMRPU bit from 1 to 0 releases the pull-up, thereby notifying the USB host that it detached.

Set this bit to 0 in host controller mode.

DPRPU bit (D+ Line Resistor Control*1)

The DPRPU bit enables or disables pulling up the D+ line in device controller mode.

When the DPRPU bit is set to 1 in device controller mode, the USBFS pulls up the D+ line to notify the USB host that it attached. Changing the DPRPU bit from 1 to 0 releases the pull-up, thereby notifying the USB host that it detached.

Set this bit to 0 in host controller mode.

DRPD bit (D+/D- Line Resistor Control)

The DRPD bit enables or disables pulling down D+ and D- lines in host controller mode.

Set this bit to 1 in host controller mode and to 0 in device controller mode.

DCFM bit (Controller Function Select)

The DCFM bit selects the host or device function of the USBFS.

Only change this bit when the DMRPU, DPRPU, and DRPD bits are both 0.

CNEN bit (CNEN Single-Ended Receiver Enable)

Setting the CNEN bit to 1 enables the single-ended receiver and sets the LNST bit to monitor the status of D+ and D- lines.

The CNEN bit is used when the USBFS operates as a portable device for battery charging.

SCKE bit (USB Clock Enable*2)

The SCKE bit stops or enables supplying 48-MHz clock supply to the USBFS.

When this bit is 0, only SYSCFG is permitted to be read from and written to. The other USB-related registers should not be read from or written to.

28.2.2 System Configuration Status Register 0 (SYSSTS0)

Address(es): USBFS.SYSSTS0 4009 0004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVCMON[1:0]		—	—	—	—	—	—	—	HTACT	—	—	—	IDMON	LNST[1:0]	
Value after reset:	0*1	0*1	0	0	0	0	0	0	0	0	0	0	0	0*1	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	LNST[1:0]	USB Data Line Status Monitor	Indicates the status of the USB data lines, see Table 28.4	R
b2	IDMON	External ID0 Input Pin Monitor	0: USB_ID pin is low 1: USB_ID pin is high.	R
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b6	HTACT	USB Host Sequencer Status Monitor	0: Host sequencer completely stopped 1: Host sequencer not completely stopped.	R
b13 to b7	—	Reserved	These bits are read as 0 and cannot be modified	R
b15, b14	OVCMON[1:0]	External USB_OVRCURA/ USB_OVRCURB Input Pin Monitor	OVCMON[1] bit indicates the USB_OVRCURA pin status OVCMON[0] bit indicates the USB_OVRCURB pin status.	R

Note 1. Depends on the status of the USB_OVRCURA/USB_OVRCURB and USB_ID pins.

LNST[1:0] bits (USB Data Line Status Monitor)

The LNST[1:0] bits indicate the state of the USB data lines, D+ and D-. For details, see Table 28.4.

In device controller mode, read the LNST[1:0] bits after connection processing (SYSCFG.DPRPU bit = 1). In host controller mode, read them after enabling pull-down of the lines (SYSCFG.DRPD bit = 1).

HTACT bit (USB Host Sequencer Status Monitor)

The HTACT bit is 0 when the host sequencer of the USBFS is completely stopped.

In host controller mode, check that the HTACT bit is 0 before setting the DVSTCTR0.UACT bit to 0 to place the USBFS in the suspended state or setting the SCKE bit to 0 to stop the clock supply during communication.

OVCMON[1:0] bits (External USB_OVRCURA/ USB_OVRCURB Input Pin Monitor)

The OCVMON[1:0] bits indicate the status of the overcurrent signals from an external power supply IC.

Table 28.4 Status of USB data bus lines (D+ Line, D- Line)

LNST[1:0] bits	During full-speed operation	During low-speed operation
00b	SE0	SE0
01b	J-State	K-State
10b	K-State	J-State
11b	SE1	SE1

28.2.3 Device State Control Register 0 (DVSTCTR0)

Address(es): USBFS.DVSTCTR0 4009 0008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HNPBTOA	EXICEN	VBUSEN	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	RHST[2:0]	USB Bus Reset Status	• In host controller mode: b2 b0 0 0 0: Communication speed indeterminate (powered state or no connection) 1 x x: USB bus reset in progress 0 0 1: Low-speed connection 0 1 0: Full-speed connection. • In device controller mode: b2 b0 0 0 0: Communication speed indeterminate 0 0 1: USB bus reset in progress or low-speed connection 0 1 0: USB bus reset in progress or full-speed connection.	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	UACT	USB Bus Enable	0: Disable downstream port (disable SOF transmission) 1: Enable downstream port (enable SOF transmission).	R/W
b5	RESUME	Resume Output	0: Do not output resume signal 1: Output resume signal.	R/W
b6	USBRST	USB Bus Reset Output	0: Do not output USB bus reset signal 1: Output USB bus reset signal.	R/W
b7	RWUPE	Wakeup Detection Enable	0: Disable downstream port wakeup 1: Enable downstream port wakeup.	R/W
b8	WKUP	Wakeup Output	0: Do not output remote wakeup signal 1: Output remote wakeup signal.	R/W

Bit	Symbol	Bit name	Description	R/W
b9	VBUSEN	USB_VBUSEN Output Pin Control	0: Output low on external USB_VBUSEN pin 1: Output high on external USB_VBUSEN pin.	R/W
b10	EXICEN	USB_EXICEN Output Pin Control	0: Output low on external USB_EXICEN pin 1: Output high on external USB_EXICEN pin.	R/W
b11	HNPBTOA	Host Negotiation Protocol (HNP) Control	This bit is used when switching from device B to device A in OTG mode. If the HNPBTOA bit is 1, the internal function control remains in the suspended state until the HNP processing ends even if SYSCFG.DPRPU = 0 or SYSCFG.DCFM = 1.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

RHST[2:0] bits (USB Bus Reset Status)

The RHST[2:0] bits indicate the status of the USB bus reset.

In host controller mode, writing 1 to the USBRST bit causes the RHST[2:0] bits to set to 100b. When 0 is written to the USBRST bit and the USBFS ends the SE0 state, the RHST[2:0] bits update to a new value.

In device controller mode, if the USBFS detects a USB bus reset, the RHST[2:0] bits are set to 010b if the DPRPU bit is 1 or 001b if the DMRPU is 1, and a DVST interrupt is generated.

UACT bit (USB Bus Enable)

When set to 1 in host controller mode, the UACT bit enables USB bus operation by controlling SOF packet transmission to the USB bus in addition to data and reception. The USBFS starts SOF packet output within one frame period after this bit is set to 1. When UACT is set to 0, the USBFS enters the idle state after the SOF packet output.

The USBFS sets the UACT bit to 0 on any of the following conditions:

- A DTCH interrupt is detected during communication (when UACT = 1)
- An EOFERR interrupt is detected during communication (when UACT = 1).

Always write 1 to the UACT bit at the end of the USB bus reset processing (writing 0 to the USBRST bit) or at the end of resume processing from the suspended state (writing 0 to the RESUME bit).

In device controller mode, always set this bit to 0.

RESUME bit (Resume Output)

The RESUME bit controls the resume signal output in host controller mode.

When this bit is set to 1, the USBFS drives the USB port to the K-state and outputs the resume signal. The USBFS sets the bit to 1 on detection of a remote wakeup signal while the RWUPE bit is 1 and in the USB Suspend state.

The USBFS continues outputting the K-state while the RESUME bit is 1, until the bit is set to 0 by software. The RESUME bit must be 1 (resume period) for the time defined in the USB 2.0 specification. Only set this bit to 1 while the interface is in the suspended state. Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to the RESUME bit).

Always set this bit to 0 in device controller mode.

USBRST bit (USB Bus Reset Output)

The USBRST bit controls the output of the USB bus reset signal in host controller mode. When this bit is set to 1, the USBFS drives the USB port to the SE0 state to reset the USB bus. The USBFS continues outputting SE0 while the USBRST bit is 1, until the bit is set to 1 by software. The USBRST bit must be 1 (USB bus reset period) for the time defined in the USB 2.0 specification.

Writing 1 to this bit during communication (UACT bit = 1) or during resume processing (RESUME bit = 1) prevents the USBFS from starting USB bus reset processing until both the UACT and RESUME bits become 0. Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to the USBRST bit).

Always set this bit to 0 in device controller mode.

RWUPE bit (Wakeup Detection Enable)

The RWUPE bit enables or disables remote wakeup signals (resume signals) from downstream peripheral devices in host controller mode. When this bit is set to 1, the USBFS detects a remote wakeup signal (K-state for 2.5 µs) from a downstream peripheral device, and it performs resume processing, driving the K-state.

When this bit set to 0, the USBFS ignores remote wakeup signals (K-states) from peripheral devices connected to the USB port. Do not stop the internal clock while the RWUPE bit is 1, even in the suspended state(SYSCFG.SCKE bit must be set to 1).

Always set this bit to 0 in device controller mode.

WKUP bit (Wakeup Output)

The WKUP bit enables or disables remote wakeup signals (resume signals) to the USB bus in device controller mode.

The USBFS controls the output timing of the remote wakeup signals. When this bit is set to 1, the USBFS clears it to 0 after outputting the K-state for 10 ms. The USB 2.0 specification specifies that the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If the USB writes 1 to this bit immediately after detecting the suspended state, the K-state is output after 2 ms.

Only write 1 to this bit when the device is in the suspended state (INTSTS0.DVSQ[2:0] bits = 1xxb) and the USB host enables the remote wakeup signal. Do not stop the internal clock while this bit is 1, even in the suspended state (SYSCFG.SCKE bit is 1).

Always set this bit to 0 in host controller mode.

HNPBTOA bit (Host Negotiation Protocol (HNP) Control)

The HNPBTOA bit is used when switching from device B to device A while in OTG mode.

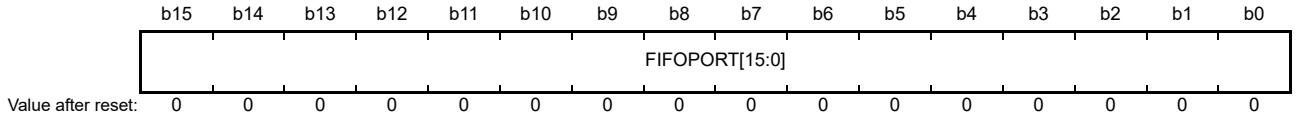
If the HNPBTOA bit is 1, the internal function control maintains the suspended state until the HNP processing ends, even if the SYSCFG.DPRPU bit is 0 or the SYSCFG.DCFM bit is set to 1. Resume (RESM) interrupts are not generated even if the falling edge of the D+ signal is detected.

The HNP processing ends when a host attach event is detected, because of a pull-up by the initiating party, or the HNPBTOA bit is set to 0 by software because the HNP processing times out.

28.2.4 CFIFO Port Register (CFIFO/CFIFOL)
D0FIFO Port Register (D0FIFO/D0FIFOL)
D1FIFO Port Register (D1FIFO/D1FIFOL)

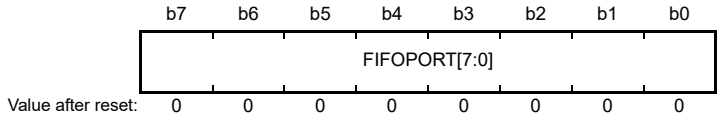
(1) When the MBW bit is 1

Address(es): USBFS.CFIFO 4009 0014h, USBFS.D0FIFO 4009 0018h, USBFS.D1FIFO 4009 001Ch



(2) When the MBW bit is 0

Address(es): USBFS.CFIFOL 4009 0014h, USBFS.D0FIFOL 4009 0018h, USBFS.D1FIFOL 4009 001Ch



Bit	Symbol	Bit name	Description	R/W
b15 to b0	FIFOPORT[15:0]*1	FIFO Port	Read receive data from the FIFO buffer or write transmit data to the FIFO buffer by accessing these bits	R/W

Note 1. The valid bits depend on the MBW settings (CFIFOSEL.MBW, D0FIFOSEL.MBW, and D1FIFOSEL.MBW) and BIGEND settings (CFIFOSEL.BIGEND, D0FIFOSEL.BIGEND, and D1FIFOSEL.BIGEND) in the associated port select register. See Table 28.5 and Table 28.6.

Three FIFO ports are available:

- CFIFO
- D0FIFO
- D1FIFO.

Each FIFO port is configured with:

- A port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer and writing of data to the FIFO buffer
- A port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that selects the pipe assigned to the FIFO port
- A port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR).

Each FIFO port has the following constraints:

- Access to the FIFO buffer for DCP control transfers is through the CFIFO port
- Access to the FIFO buffer for DMA or DTC transfers is through the D0FIFO or D1FIFO port
- The D0FIFO and D1FIFO ports can also be accessed by the CPU
- When using functions specific to the FIFO port, such as the DMA or DTC transfer function, the pipe number selected in the CURPIPE[3:0] bits of the port select register cannot be changed
- Registers configuring a FIFO port do not affect other FIFO ports
- The same pipe must not be assigned to two or more FIFO ports
- There are two FIFO buffer states, one giving access rights to the CPU and the other to the serial interface engine (SIE). When the SIE has access rights, the FIFO buffer cannot be accessed by the CPU.

FIFOPORT[15:0] bits (FIFO Port)

When the FIFOPORT bit is accessed, the USBFS reads the received data from the FIFO buffer or writes the transmit data to the FIFO buffer. The FIFO port register can be accessed only when the FRDY bit in the associated port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in the FIFO port register depend on the MBW and BIGEND settings in the port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL). See Table 28.5 and Table 28.6.

Table 28.5 Endian operation in 16-bit access

CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit D1FIFOSEL.BIGEND Bit	Bits 15 to 8	Bits 7 to 0
0	N + 1 data	N + 0 data
1	N + 0 data	N + 1 data

Table 28.6 Endian operation in 8-bit access

CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit D1FIFOSEL.BIGEND Bit	Bits 15 to 8	Bits 7 to 0
0	Access prohibited*1	N + 0 data
1	Access prohibited*1	N + 0 data

Note 1. Writing to or reading from these areas is not allowed.

28.2.5 CFIFO Port Select Register (CFIFOSEL) D0FIFO Port Select Register (D0FIFOSEL) D1FIFO Port Select Register (D1FIFOSEL)

CFIFOSEL

Address(es): USBFS.CFIFOSEL 4009 0020h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCNT	REW	—	—	—	MBW	—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	CURPIPE[3:0]	CFIFO Port Access Pipe Specification	b3 b0 0 0 0 0: DCP (default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9. Other settings are prohibited.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	ISEL	CFIFO Port Access Direction When DCP is Selected	0: Reading from the buffer memory is selected 1: Writing to the buffer memory is selected.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	CFIFO Port Endian Control	0: Little endian 1: Big endian.	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	CFIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width.	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewound 1: The buffer pointer is rewound.	R/W*1
b15	RCNT	Read Count Mode	0: The DTLN[8:0] bits (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) are cleared when all receive data is read from the CFIFO. In double buffer mode, the DTLN[8:0] bit value is cleared when all data is read from only a single plane. 1: The DTLN[8:0] bits are decremented each time the receive data is read from the CFIFO.	R/W

Note 1. Only 0 can be read.

Do not specify the same pipe number in the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

Do not change the pipe number while DMA or DTC transfer is enabled.

CURPIPE[3:0] bits (CFIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number to use for reading or writing data through the CFIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

During FIFO buffer access, even when an attempt is made to change the CURPIPE[3:0] setting, the current access setting is retained until access is complete.

ISEL bit (CFIFO Port Access Direction When DCP is Selected)

After writing a new value to the ISEL bit with the DCP as the selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process. Set this bit and the CURPIPE[3:0] bits simultaneously.

MBW bit (CFIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the CFIFO port.

When the selected pipe is receiving, set the CURPIPE[3:0] bits and MBW bits simultaneously. After a write to these bits starts a data read from the FIFO buffer, do not change the bits until all of the data is read. When you read the FIFO buffer, read with the access size, which is set in the MBW bit.

When the selected pipe is transmitting, the bit width cannot be changed from 8-bit width to 16-bit width while data is written to the buffer memory.

An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

REW bit (Buffer Pointer Rewind)

The REW bit specifies whether to rewind the buffer pointer.

When the selected pipe is receiving, setting this bit to 1 while the FIFO buffer is being read allows re-reading of the FIFO buffer from the first data. In double buffering, this setting enables re-reading of the currently-read FIFO buffer plane from the first entry.

Do not set this bit to 1 while simultaneously changing the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY bit is 1.

To rewrite to the FIFO buffer from the first data for the transmitting pipe, use the BCLR bit.

D0FIFOSEL, D1FIFOSEL

Address(es): USBFS.D0FIFOSEL 4009 0028h, USBFS.D1FIFOSEL 4009 002Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCNT	REW	DCLRM	DREQE	—	MBW	—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	CURPIPE [3:0]	FIFO Port Access Pipe Specification	b3 b0 0 0 0 0: No pipe specification 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9. Other settings are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	FIFO Port Endian Control	0: Little endian 1: Big endian.	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	FIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	DREQE	DMA/DTC Transfer Request Enable	0: DMA/DTC transfer request is disabled 1: DMA/DTC transfer request is enabled.	R/W

Bit	Symbol	Bit name	Description	R/W
b13	DCLRM	Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read	0: Auto buffer clear mode is disabled 1: Auto buffer clear mode is enabled.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewound 1: The buffer pointer is rewound.	R/W*1
b15	RCNT	Read Count Mode	0: DTLN[8:0] bits (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) cleared when all receive data is read from DnFIFO. (In double buffer mode, the DTLN bit Value is cleared when all data is read from only a single plane.) 1: DTLN[8:0] bits decrement each time receive data is read from DnFIFO. (n = 0, 1)	R/W

Note 1. Only 0 can be read.

The same pipe must not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected. The pipe number must not be changed while DMA or DTC transfer is enabled.

CURPIPE[3:0] bits (FIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number to use for reading or writing data through the D0FIFO port or D1FIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

During FIFO buffer access, even when an attempt is made to change the CURPIPE[3:0] setting, the current access setting is retained until access is complete.

MBW bit (FIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the D0FIFO port or D1FIFO port.

When the selected pipe is receiving, after a write to these bits starts a data read from the FIFO buffer, do not change the bits until all of the data is read. When you read the FIFO buffer, read with the access size which is set in MBW bit. Set the CURPIPE[3:0] bits and the MBW bit simultaneously.

When the selected pipe is transmitting, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the FIFO memory.

An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

DREQE bit (DMA/DTC Transfer Request Enable)

The DREQE bit enables or disables issuing of DMA or DTC transfer requests. To enable DMA or DTC transfer requests, set this bit to 1 after setting the CURPIPE[3:0] bits. To change the CURPIPE[3:0] setting, first set this bit to 0.

DCLRM bit (Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read)

The DCLRM bit enables or disables automatic FIFO buffer clearing after data in the selected pipe is read.

When this bit is set to 1, on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or when reading of a received short packet is complete while the PIPECFG.BFRE bit is 1, the USBFS sets the BCLR bit in the FIFO port control register to 1.

When using the USBFS with the SOFCFG.BRDYM bit set to 1, set this bit to 0.

REW bit (Buffer Pointer Rewind)

The REW bit specifies whether to rewind the buffer pointer.

When the selected pipe is receiving, setting this bit to 1 while the FIFO buffer is being read allows re-reading of the FIFO buffer from the first data. In double buffering, this setting enables re-reading of the currently-read FIFO buffer plane from the first entry.

Do not set this bit to 1 while simultaneously changing the CURPIPE[3:0] bits. Before setting this bit to 1, always check that the FRDY bit is 1. To rewrite to the FIFO buffer from the first data for the transmitting pipe, use the BCLR bit.

RCNT bit (Read Count Mode)

The RCNT bit specifies the read mode for the value in the CFIFOCTR.DTLN bit. When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

28.2.6 CFIFO Port Control Register (CFIFOCTR) D0FIFO Port Control Register (D0FIFOCTR) D1FIFO Port Control Register (D1FIFOCTR)

Address(es): USBFS.CFIFOCTR 4009 0022h, USBFS.D0FIFOCTR 4009 002Ah, USBFS.D1FIFOCTR 4009 002Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	DTLN[8:0]	Receive Data Length	Receive data length. The meaning of the values differs depending on the RCNT bit setting in the port select register. For details, see the description of the DTLN[8:0] bits.	R
b12 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FRDY	FIFO Port Ready	0: FIFO port access disabled 1: FIFO port access enabled.	R
b14	BCLR	CPU Buffer Clear	0: Does not operate 1: FIFO buffer cleared on the CPU side.	R/W*1
b15	BVAL	Buffer Memory Valid Flag	0: Invalid 1: Writing ended.	R/W

Note 1. Only 0 can be read.

The CFIFOCTR, D0FIFOCTR, and D1FIFOCTR registers correspond to the CFIFO, D0FIFO, and D1FIFO buffers.

DTLN[8:0] bits (Receive Data Length)

The DTLN[8:0] bits indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the DnFIFOSEL.RCNT bit (n = 0, 1), as follows:

- RCNT = 0
  The USBFS sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU or DMA/DTC has read all of the received data from a single FIFO buffer plane.
  While the PIPECFG.BFRE bit = 1, the USBFS retains the length of the receive data until the BCLR bit is set to 1 even after all the data is read.
- RCNT = 1
  The USBFS decrements the value indicated by the DTLN[8:0] bits each time data is read from the FIFO buffer. The value is decremented by 1 when the MBW bit is 0, and by 2 when the MBW bit is 1.
  The USBFS sets these bits to 0 when all the data is read from one FIFO buffer plane. However, in double buffer mode, if data is received in one FIFO buffer plane before all the data is read from the other plane, the USBFS sets these bits to indicate the length of the receive data in the former plane when all of the data is read from the latter plane.

FRDY bit (FIFO Port Ready)

The FRDY bit indicates whether the FIFO port can be accessed by the CPU or DMA/DTC.

In the following cases, the USBFS sets the FRDY bit to 1 but data cannot be read through the FIFO port because there is no data to be read:

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit = 1.

In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

BCLR bit (CPU Buffer Clear)

Set the BCLR bit to 1 to clear the FIFO buffer on the CPU for the selected pipe.

When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USBFS clears only one plane of the FIFO buffer even when both planes are read-enabled.

When the DCP is the selected pipe, setting the BCLR bit to 1 allows the USBFS to clear the FIFO buffer regardless of whether the CPU or SIE has access rights. To clear the buffer when the SIE has access rights, set the DCPCTR.PID[1:0] bits to 00b (NAK response) before setting the BCLR bit to 1.

When the selected pipe is transmitting, if 1 is written to the BVAL flag and the BCLR bit simultaneously, the USBFS clears the data that is already written, enabling transmission of a zero-length packet. When the selected pipe is not the DCP, only write 1 to the BCLR bit while the FRDY flag in the FIFO port control register is 1 (set by the USBFS).

BVAL flag (Buffer Memory Valid Flag)

Set the BVAL flag to 1 when data is completely written to the FIFO buffer on the CPU for the pipe selected in CURPIPE[3:0].

When the selected pipe is transmitting, set this flag to 1 in the following cases:

- To transmit a short packet, set this flag to 1 after data is written
- To transmit a zero-length packet, set this flag to 1 before data is written to the FIFO buffer.

The USBFS then switches the FIFO buffer from the CPU to the SIE, enabling transmission.

When data of the maximum packet size is written for the pipe in continuous transfer mode, the USBFS sets the BVAL flag to 1 and switches the FIFO buffer from the CPU to the SIE, enabling transmission.

Only write 1 to the BVAL flag while the FRDY bit is 1 (set by the USBFS). When the selected pipe is receiving, do not set the BVAL flag to 1.

28.2.7 Interrupt Enable Register 0 (INTENB0)

Address(es): USBFS.INTENB0 4009 0030h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BRDYE	Buffer Ready Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b9	NRDYE	Buffer Not Ready Response Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b10	BEMPE	Buffer Empty Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b11	CTRE	Control Transfer Stage Transition Interrupt Enable*1	0: Disable interrupt request 1: Enable interrupt request.	R/W
b12	DVSE	Device State Transition Interrupt Enable*1	0: Disable interrupt request 1: Enable interrupt request.	R/W
b13	SOFE	Frame Number Update Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W
b14	RSME	Resume Interrupt Enable*1	0: Disable interrupt request 1: Enable interrupt request.	R/W
b15	VBSE	VBUS Interrupt Enable	0: Disable interrupt request 1: Enable interrupt request.	R/W

Note 1. The RSME, DVSE, and CTRE bits can only be set to 1 in device controller mode. Do not set these bits to 1 in host controller mode.

When a status flag in the INTSTS0 register is set to 1 and the associated interrupt request enable bit setting in the INTENB0 register is 1, the USBFS issues a USBFS interrupt request. Regardless of the INTENB0 register setting, the status flag in the INTSTS0 register is set to 1 in response to a state change that satisfies the associated condition. When an interrupt request enable bit in the INTENB0 register is switched from 0 to 1 while the associated status flag in the INTSTS0 register is set to 1, a USBFS interrupt is requested.

28.2.8 Interrupt Enable Register 1 (INTENB1)

Address(es): USBFS.INTENB1 4009 0032h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	—	—	—	EOFERRE	SIGNE	SACKE	—	—	—	PDDETINTE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PDDETINTE0	PDDETINT0 Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACKE	Setup Transaction Normal Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b5	SIGNE	Setup Transaction Error Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b6	EOFERRE	EOF Error Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCHE	Connection Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b12	DTCHE	Disconnection Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHGE	USB Bus Change Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b15	OVRCRE	Overcurrent Input Change Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled.	R/W

Note: The bits in INTENB1 can only be set to 1 in host controller mode. Do not set these bits to 1 in device controller mode.

INTENB1 specifies the interrupt masks in host controller mode and for the setup transaction.

When a status flag in the INTSTS1 register is set to 1 and the associated interrupt request enable bit setting in the INTENB1 register is 1, the USBFS issues a USBFS interrupt request.

Regardless of the INTENB1 register setting, the status flag in the INTSTS1 register is set to 1 in response to a state change that satisfies the associated condition.

When an interrupt request enable bit in the INTENB1 register is switched from 0 to 1 while the associated status flag in the INTSTS1 register is set to 1, a USBFS interrupt is requested.

Do not enable interrupts in device controller mode.

28.2.9 BRDY Interrupt Enable Register (BRDYENB)

Address(es): USBFS.BRDYENB 4009 0036h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BRDYE	PIPE8BRDYE	PIPE7BRDYE	PIPE6BRDYE	PIPE5BRDYE	PIPE4BRDYE	PIPE3BRDYE	PIPE2BRDYE	PIPE1BRDYE	PIPE0BRDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PIPE0BRDYE	BRDY Interrupt Enable for PIPE0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1BRDYE	BRDY Interrupt Enable for PIPE1	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b2	PIPE2BRDYE	BRDY Interrupt Enable for PIPE2	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b3	PIPE3BRDYE	BRDY Interrupt Enable for PIPE3	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b4	PIPE4BRDYE	BRDY Interrupt Enable for PIPE4	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b5	PIPE5BRDYE	BRDY Interrupt Enable for PIPE5	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b6	PIPE6BRDYE	BRDY Interrupt Enable for PIPE6	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b7	PIPE7BRDYE	BRDY Interrupt Enable for PIPE7	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b8	PIPE8BRDYE	BRDY Interrupt Enable for PIPE8	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b9	PIPE9BRDYE	BRDY Interrupt Enable for PIPE9	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BRDYENB register enables or disables the INTSTS0.BRDY bit to be set to 1 when the BRDY interrupt is detected for each pipe.

When a status flag in the BRDYSTS register is set to 1 and the associated PIPEnBRDYE bit (n = 0 to 9) setting in the BRDYENB register is 1, the INTSTS0.BRDY flag is set to 1. In this case, if the BRDYE bit in INTENB0 is 1, the USBFS generates a BRDY interrupt request. While at least one PIPEnBRDY bit indicates 1, the USBFS generates the BRDY interrupt request when the associated interrupt request enable bit in the BRDYENB register is changed from 0 to 1 by the software.

28.2.10 NRDY Interrupt Enable Register (NRDYENB)

Address(es): USBFS.NRDYENB 4009 0038h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9NRDYE	PIPE8NRDYE	PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	PIPE3NRDYE	PIPE2NRDYE	PIPE1NRDYE	PIPE0NRDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PIPE0NRDYE	NRDY Interrupt Enable for PIPE0	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b1	PIPE1NRDYE	NRDY Interrupt Enable for PIPE1	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b2	PIPE2NRDYE	NRDY Interrupt Enable for PIPE2	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b3	PIPE3NRDYE	NRDY Interrupt Enable for PIPE3	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b4	PIPE4NRDYE	NRDY Interrupt Enable for PIPE4	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b5	PIPE5NRDYE	NRDY Interrupt Enable for PIPE5	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b6	PIPE6NRDYE	NRDY Interrupt Enable for PIPE6	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b7	PIPE7NRDYE	NRDY Interrupt Enable for PIPE7	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b8	PIPE8NRDYE	NRDY Interrupt Enable for PIPE8	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b9	PIPE9NRDYE	NRDY Interrupt Enable for PIPE9	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

NRDYENB enables or disables the INTSTS0.NRDY bit to be set to 1 when the NRDY interrupt is detected for each pipe.

When a status flag in the NRDYSTS register is set to 1 and the associated PIPEnNRDYE (n = 0 to 9) bit setting in the NRDYENB register is 1, the INTSTS0.NRDY flag is set to 1. In this case, if the NRDYE bit in INTENB0 is 1, the USBFS generates a NRDY interrupt request. While at least one PIPEnNRDY bit indicates 1, the USBFS generates the NRDY interrupt request when the associated interrupt request enable bit in the NRDYENB register is changed from 0 to 1 by the software.

28.2.11 BEMP Interrupt Enable Register (BEMPENB)

Address(es): USBFS.BEMPENB 4009 003Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BEMPE	PIPE8BEMPE	PIPE7BEMPE	PIPE6BEMPE	PIPE5BEMPE	PIPE4BEMPE	PIPE3BEMPE	PIPE2BEMPE	PIPE1BEMPE	PIPE0BEMPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PIPE0BEMPE	BEMP Interrupt Enable for PIPE0	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b1	PIPE1BEMPE	BEMP Interrupt Enable for PIPE1	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b2	PIPE2BEMPE	BEMP Interrupt Enable for PIPE2	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b3	PIPE3BEMPE	BEMP Interrupt Enable for PIPE3	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b4	PIPE4BEMPE	BEMP Interrupt Enable for PIPE4	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b5	PIPE5BEMPE	BEMP Interrupt Enable for PIPE5	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b6	PIPE6BEMPE	BEMP Interrupt Enable for PIPE6	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b7	PIPE7BEMPE	BEMP Interrupt Enable for PIPE7	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b8	PIPE8BEMPE	BEMP Interrupt Enable for PIPE8	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b9	PIPE9BEMPE	BEMP Interrupt Enable for PIPE9	0: Interrupt output disabled 1: Interrupt output enabled.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BEMPENB register enables or disables the INTSTS0.BEMP bit to be set to 1 when the BEMP interrupt is detected for each pipe.

When a status flag in the BEMPSTS register is set to 1 and the associated PIPEnBEMPE (n = 0 to 9) bit setting in the BEMPENB register is 1, the INTSTS0.BEMP flag is set to 1. In this case, if the BEMPE bit in INTENB0 is 1, the USBFS generates a BEMP interrupt request. While at least one PIPEnBEMP bit indicates 1, the USBFS generates the BEMP interrupt request when the associated interrupt request enable bit in the BEMPENB register is changed from 0 to 1 by the software.

28.2.12 SOF Output Configuration Register (SOFCFG)

Address(es): USBFS.SOFCFG 4009 003Ch

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

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	EDGESTS	Edge Interrupt Output Status Monitor*1	Indicates 1 during the edge processing of an edge interrupt output signal.	R
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	BRDYM	BRDY Interrupt Status Clear Timing	0: BRDY flag cleared by software 1: BRDY flag cleared by the USBFS through a data read from the FIFO buffer or data write to the FIFO buffer.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	TRNENSEL	Transaction-Enabled Time Select*1	0: Not low-speed communication 1: Low-speed communication.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Confirm that the EDGESTS and TRNENSEL bits are 0 before stopping the clock supply to the USBFS.

EDGESTS bit (Edge Interrupt Output Status Monitor*1)

The EDGESTS bit indicates 1 during the edge processing of an edge interrupt output signal. Confirm that this bit is 0 before stopping the clock supply to the USBFS.

BRDYM bit (BRDY Interrupt Status Clear Timing)

The BRDYM bit specifies how the BRDY interrupt status flags for the pipes are cleared.

TRNENSEL bit (Transaction-Enabled Time Select*1)

When the USB port is in use for full- or low-speed communications, the TRNENSEL bit specifies the timing with which the USBFS issues tokens in a frame (transaction-enabled time).

Set this bit to 1 when a low-speed device is connected. The bit is only valid in host controller mode. Set this bit to 0 in device controller mode.

28.2.13 Interrupt Status Register 0 (INTSTS0)

Address(es): USBFS.INTSTS0 4009 0040h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Value after reset:	0	0	0	0/1*1	0	0	0	0	0*2	0*3	0*3	0/1*3	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CTSQ[2:0]	Control Transfer Stage	b2 b0 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error.	R

Bit	Symbol	Bit name	Description	R/W
b3	VALID	USB Request Reception	0: Setup packet is not received 1: Setup packet is received.	R/W*4
b6 to b4	DVSQ[2:0]	Device State	b6 b4 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state 1 x x: Suspended state.	R
b7	VBSTS	VBUS Input Status	0: USB_VBUS pin is low 1: USB_VBUS pin is high.	R
b8	BRDY	Buffer Ready Interrupt Status	0: BRDY interrupts are not generated 1: BRDY interrupts are generated.	R
b9	NRDY	Buffer Not Ready Interrupt Status	0: NRDY interrupts are not generated 1: NRDY interrupts are generated.	R
b10	BEMP	Buffer Empty Interrupt Status	0: BEMP interrupts are not generated 1: BEMP interrupts are generated.	R
b11	CTRT	Control Transfer Stage Transition Interrupt Status*5	0: Control transfer stage transition interrupts are not generated 1: Control transfer stage transition interrupts are generated.	R/W*4
b12	DVST	Device State Transition Interrupt Status*5	0: Device state transition interrupts are not generated 1: Device state transition interrupts are generated.	R/W*4
b13	SOFR	Frame Number Refresh Interrupt Status	0: SOF interrupts are not generated 1: SOF interrupts are generated.	R/W*4
b14	RESM	Resume Interrupt Status*5,*6	0: Resume interrupts are not generated 1: Resume interrupts are generated.	R/W*4
b15	VBINT	VBUS Interrupt Status*6	0: VBUS interrupts are not generated 1: VBUS interrupts are generated.	R/W*4

x: Don't care

Note 1. The value is 0 when the MCU is reset and 1 after a USB bus reset.

Note 2. The value is 1 when the USB_VBUS pin is high and 0 when the USB_VBUS pin is low.

Note 3. The value is 000b when the MCU is reset and 001b after a USB bus reset.

Note 4. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID bit, write 0 only to the bits to be cleared. Write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Note 5. The status of the RESM, DVST, and CTRT bits is changed only in device controller mode. Set the associated interrupt enable bits to 0 (disabled) in host controller mode.

Note 6. The USBFS detects a change in the status indicated by the VBINT and RESM bits even while the clock supply is stopped (SCKE bit = 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply before clearing the status by the software

CTSQ[2:0] bits (Control Transfer Stage)

In host controller mode, the read value is invalid.

VALID bit (USB Request Reception)

In host controller mode, the read value is invalid.

DVSQ[2:0] bits (Device State)

The DVSQ[2:0] bits are initialized by a USB bus reset. In host controller mode, the read value is invalid.

BRDY bit (Buffer Ready Interrupt Status)

This bit indicates the BRDY interrupt status.

The USBFS sets the BRDY bit to 1 when it detects a BRDY interrupt status (PIPEnBRDY = 1, n = 0 to 9) on at least one pipe for which BRDY interrupts are enabled (BRDYENB.PIPEnBRDYE = 1).

For the conditions that cause the PIPEnBRDY status to be asserted, see section 28.3.3.1, BRDY Interrupt.

The USBFS sets the BRDY bit to 0 when software writes 0 to all the PIPEnBRDY bits associated with the PIPEnBRDYE bits that are set to 1. Writing 0 to the BRDY bit in the software does not clear the bit.

NRDY bit (Buffer Not Ready Interrupt Status)

The USBFS sets the NRDY bit to 1 when at least one PIPEnNRDY bit (n = 0 to 9) is set to 1 among the PIPENRDY bits corresponding to the PIPEnNRDYE bits (n = 0 to 9) which has been set to 1 (when the USBFS detects the NRDY interrupt status in at least one pipe among the pipes for which software enables the NRDY interrupt output).

For the conditions that cause the PIPEnNRDY status to be asserted, see section 28.3.3.2, NRDY Interrupt.

The USBFS sets the NRDY bit to 0 when software writes 0 to all the PIPEnNRDY bits associated with the PIPEnNRDYE bits that are set to 1. Writing 0 to the NRDY bit in the software does not clear the bit.

BEMP bit (Buffer Empty Interrupt Status)

This bit indicates the BEMP interrupt status.

The USBFS sets the BEMP bit to 1 when it detects a BEMP interrupt status (PIPEnBEMP = 1, n = 0 to 9) on at least one pipe for which BEMP interrupts are enabled (BEMPENB.PIPEnBEMPE = 1).

For the conditions that cause the PIPEnBEMP status to be asserted, see section 28.3.3.3, BEMP Interrupt.

The USBFS sets the BEMP bit to 0 when software writes 0 to all of the PIPEnBEMP bits associated with the PIPEnBEMPE bits that are set to 1. Writing 0 to the BEMP bit in the software does not clear the bit.

CTRT bit (Control Transfer Stage Transition Interrupt Status)

In device controller mode, the USBFS updates the value of the CTSQ[2:0] bits and sets the CTRT bit to 1 on detecting a transition in the control transfer stage. When a control transfer stage transition interrupt is generated, clear the CTRT bit before the USBFS detects the next control transfer stage transition.

Values read from the CTRT bit in host controller mode are invalid.

DVST bit (Device State Transition Interrupt Status)

In device controller mode, the USBFS updates the value of the DVSQ[2:0] bits and sets the DVST bit to 1 on detecting a change in the device state. When a device state transition interrupt is generated, clear the DVST bit before the USBFS detects the next device state transition.

Values read from the DVST bit in host controller mode are invalid.

SOFR bit (Frame Number Refresh Interrupt Status)

In host controller mode, the USBFS sets the SOFR bit to 1 on updating the frame number when the DVSTCTR0.UACT bit has been set to 1 by software. An SOFR interrupt is detected every 1 ms.

In device controller mode, the USBFS sets the SOFR bit to 1 on updating the frame number. A frame number refresh interrupt is detected every 1 ms.

The USBFS can detect an SOFR interrupt through the internal interpolation function even when a corrupted SOF packet is received from the USB host.

RESM bit (Resume Interrupt Status)

In device controller mode, the USBFS sets the RESM bit to 1 on detecting the falling edge of the signal on the USB_DP pin in the suspended state (DVSQ[2:0] = 1xxb). Values read from the RESM bit in host controller mode are invalid.

VBINT bit (VBUS Interrupt Status)

The USBFS sets the VBINT bit to 1 on detecting a level change (high to low or low to high) in the USB_VBUS pin input value. The USBFS sets the VBSTS bit to indicate the USB_VBUS pin input value. When a VBUS interrupt is generated, eliminate transient elements by reading the VBSTS flag at least three times through software processing and check that the values read are the same.

28.2.14 Interrupt Status Register 1 (INTSTS1)

Address(es): USBFS.INTSTS1 4009 0042h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCR	BCHG	—	DTCH	ATTCH	—	—	—	—	EOFERR	SIGN	SACK	—	—	—	PDDETINT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PDDETINT0	PDDET0 Detection Interrupt Status	0: PDDET0 detection interrupts are not generated 1: PDDET0 detection interrupts are generated.	R/W *1
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACK	Setup Transaction Normal Response Interrupt Status	0: SACK interrupts are not generated 1: SACK interrupts are generated.	R/W *1
b5	SIGN	Setup Transaction Error Interrupt Status	0: SIGN interrupts are not generated 1: SIGN interrupts are generated.	R/W *1
b6	EOFERR	EOF Error Detection Interrupt Status	0: EOFERR interrupts are not generated 1: EOFERR interrupts are generated.	R/W *1
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCH	ATTCH Interrupt Status	0: ATTCH interrupts are not generated 1: ATTCH interrupts are generated.	R/W *1
b12	DTCH	USB Disconnection Detection Interrupt Status	0: DTCH interrupts are not generated. 1: DTCH interrupts are generated.	R/W *1
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHG	USB Bus Change Interrupt Status*2	0: BCHG interrupts are not generated 1: BCHG interrupts are generated.	R/W *1
b15	OVRCR	Over Current Input Change Interrupt Status*2	0: OVRCR interrupts are not generated 1: OVRCR interrupts are generated.	R/W *1

Note 1. To clear the bits in INTSTS1, write 0 only to the bits to be cleared. Write 1 to the other bits.

Note 2. The USBFS detects a change in the status in the OVRCR or BCHG bit even when the clock supply is stopped (SYSCFG.SCKE = 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply (SYSCFG.SCKE = 1) before clearing the status by the software. No other interrupts can be detected while the clock supply is stopped (SYSCFG.SCKE bit = 0).

INTSTS1 is used to confirm the status of each interrupt in host controller mode. Only enable the status change interrupts indicated in the bits in INTSTS1 in host controller mode.

PDDETINT0 bit (PDDET0 Detection Interrupt Status)

The PDDETINT0 bit indicates the status of the portable device detection interrupt in host controller mode. This bit is set to 1 when the USBFS detects a level change (high to low or low to high) in the input value to the VDPDET pin of the USB physical layer transceiver (PHY). The USBFS sets the PDDETSTS0 bit to indicate the VDPDET input value. When the PDDETINT interrupt is generated, eliminate transient elements by reading the PDDETSTS0 bit at least three times through software processing and check that the values read are the same.

SACK bit (Setup Transaction Normal Response Interrupt Status)

The SACK bit indicates the status of the setup transaction normal response interrupt in host controller mode.

The USBFS detects the SACK interrupt and sets this bit to 1 when an ACK response is returned from the peripheral device during the setup transactions issued by the USBFS. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

Values read from the SACK bit in device controller mode are invalid.

SIGN bit (Setup Transaction Error Interrupt Status)

The SIGN bit indicates the status of the setup transaction error interrupt in host controller mode.

The USBFS detects the SIGN interrupt and sets this bit to 1 when an ACK response is not returned from the peripheral device three consecutive times during the setup transactions issued by the USBFS. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions:

- Timeout is detected by the USBFS when the peripheral device has returned no response
- A corrupted ACK packet is received
- A handshake other than ACK (NAK, NYET, or STALL) is received.

Values read from the SIGN bit in device controller mode are invalid.

EOFERR bit (EOF Error Detection Interrupt Status)

The EOFERR bit indicates the status of the EOFERR interrupt in host controller mode.

The USBFS detects the EOFERR interrupt and sets this bit to 1 on detecting that communication did not complete at the EOF2 timing defined in the USB 2.0 specification. If the associated interrupt enable bit is set to 1 by the software, the USBFS generates the interrupt.

After detecting the EOFERR interrupt, the USBFS controls the hardware as follows, regardless of the associated interrupt enable bit setting:

- Sets the DVSTCTR0.UACT bit for the port in which the EOFERR interrupt was detected to 0
- Puts the port in which the EOFERR interrupt occurred into the idle state.

The software must terminate all pipes in which communications are currently being carried out and re-enumerate the USB port.

Values read from the EOFERR flag in device controller mode are invalid.

ATTCH bit (ATTCH Interrupt Status)

The ATTCH bit indicates the status of USB attach detection interrupts in host controller mode.

The USBFS detects the ATTCH interrupt and sets this bit to 1 on detecting a J- or K-state on the full- or low-speed signal level for 2.5 µs. If the associated interrupt enable bit is set to 1 by the software, the USBFS generates the interrupt.

The USBFS detects the ATTCH interrupt on any of the following conditions:

- K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 µs
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 µs.

Values read from the ATTCH bit in device controller mode are invalid.

DTCH bit (USB Disconnection Detection Interrupt Status)

The DTCH bit indicates the status of USB disconnection detection interrupts in host controller mode.

The USBFS detects the DTCH interrupt and sets this bit to 1 on detecting a USB bus detach event. If the associated interrupt enable bit is set to 1 by the software, the USBFS generates the interrupt.

The USBFS detects bus detach events based on the USB 2.0 specification.

After detecting the DTCH interrupt, the USBFS controls hardware as follows, regardless of the associated interrupt enable bit setting:

- Sets the DVSTCTR0.UACT bit for the port in which the DTCH interrupt was detected to 0
- Puts the port in which the DTCH interrupt is generated into the idle state.

The software must terminate all pipes in which communications are currently being carried out and invoke the wait state for attaching to the USB port (waiting for ATTCH interrupt generation).

Values read from the DTCH flag in device controller mode are invalid.

BCHG bit (USB Bus Change Interrupt Status*2)

The BCHG bit indicates the status of USB bus change interrupts in host controller mode.

The USBFS detects the BCHG interrupt and sets this bit to 1 when a change in the full- or low-speed signal level occurs on the USB port. This includes any change from J-state, K-state, or SE0 to J-state, K-state, or SE0. If the associated interrupt enable bit is set to 1 by the software, the USBFS generates the interrupt.

The USBFS sets the LNST[1:0] bits to indicate the current input state of the USB port. When a BCHG interrupt is generated, eliminate transient elements by repeat reading the LNST[1:0] bits by software until the same value is read at least three times.

Change in the USB bus state can be detected while the internal clock is stopped.

Values read from the BCHG flag in device controller mode are invalid.

OVRCR bit (Over Current Input Change Interrupt Status*2)

The OVRCR bit indicates the status of the USB_OVRCURA and USB_OVRCURB input pin change interrupt.

The USBFS detects the OVRCR interrupt and sets this bit to 1 when a change (high to low or low to high) occurs in at least one of the input values to the USB_OVRCURA and USB_OVRCURB pins. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

28.2.15 BRDY Interrupt Status Register (BRDYSTS)

Address(es): USBFS.BRDYSTS 4009 0046h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BRDY	PIPE8BRDY	PIPE7BRDY	PIPE6BRDY	PIPE5BRDY	PIPE4BRDY	PIPE3BRDY	PIPE2BRDY	PIPE1BRDY	PIPE0BRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PIPE0BRDY	BRDY Interrupt Status for PIPE0*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b1	PIPE1BRDY	BRDY Interrupt Status for PIPE1*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b2	PIPE2BRDY	BRDY Interrupt Status for PIPE2*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b3	PIPE3BRDY	BRDY Interrupt Status for PIPE3*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b4	PIPE4BRDY	BRDY Interrupt Status for PIPE4*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b5	PIPE5BRDY	BRDY Interrupt Status for PIPE5*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b6	PIPE6BRDY	BRDY Interrupt Status for PIPE6*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b7	PIPE7BRDY	BRDY Interrupt Status for PIPE7*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b8	PIPE8BRDY	BRDY Interrupt Status for PIPE8*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b9	PIPE9BRDY	BRDY Interrupt Status for PIPE9*2	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated by the bits in BRDYSTS, write 0 only to the bits to be cleared. Write 1 to the other bits.

Note 2. When the SOFCFG.BRDYM bit is set to 0, clear the BRDY interrupts before accessing the FIFO.

28.2.16 NRDY Interrupt Status Register (NRDYSTS)

Address(es): USBFS.NRDYSTS 4009 0048h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9NRDY	PIPE8NRDY	PIPE7NRDY	PIPE6NRDY	PIPE5NRDY	PIPE4NRDY	PIPE3NRDY	PIPE2NRDY	PIPE1NRDY	PIPE0NRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PIPE0NRDY	NRDY Interrupt Status for PIPE0	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b1	PIPE1NRDY	NRDY Interrupt Status for PIPE1	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b2	PIPE2NRDY	NRDY Interrupt Status for PIPE2	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b3	PIPE3NRDY	NRDY Interrupt Status for PIPE3	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b4	PIPE4NRDY	NRDY Interrupt Status for PIPE4	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b5	PIPE5NRDY	NRDY Interrupt Status for PIPE5	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b6	PIPE6NRDY	NRDY Interrupt Status for PIPE6	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b7	PIPE7NRDY	NRDY Interrupt Status for PIPE7	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b8	PIPE8NRDY	NRDY Interrupt Status for PIPE8	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b9	PIPE9NRDY	NRDY Interrupt Status for PIPE9	0: Interrupts are not generated 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated by the bits in NRDYSTS, write 0 only to the bits to be cleared. Write 1 to the other bits.

28.2.17 BEMP Interrupt Status Register (BEMPSTS)

Address(es): USBFS.BEMPSTS 4009 004Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BEMP	PIPE8BEMP	PIPE7BEMP	PIPE6BEMP	PIPE5BEMP	PIPE4BEMP	PIPE3BEMP	PIPE2BEMP	PIPE1BEMP	PIPE0BEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	PIPE0BEMP	BEMP Interrupt Status for PIPE0	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b1	PIPE1BEMP	BEMP Interrupt Status for PIPE1	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b2	PIPE2BEMP	BEMP Interrupt Status for PIPE2	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b3	PIPE3BEMP	BEMP Interrupt Status for PIPE3	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b4	PIPE4BEMP	BEMP Interrupt Status for PIPE4	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1

Bit	Symbol	Bit name	Description	R/W
b5	PIPE5BEMP	BEMP Interrupt Status for PIPE5	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b6	PIPE6BEMP	BEMP Interrupt Status for PIPE6	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b7	PIPE7BEMP	BEMP Interrupt Status for PIPE7	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b8	PIPE8BEMP	BEMP Interrupt Status for PIPE8	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b9	PIPE9BEMP	BEMP Interrupt Status for PIPE9	0: Interrupts are not generated 1: Interrupts are generated.	R/W*1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated by the bits in BEMPSTS, write 0 only to the bits to be cleared. Write 1 to the other bits.

28.2.18 Frame Number Register (FRMNUM)

Address(es): USBFS.FRMNUM 4009 004Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRN	CRCE	—	—	—	FRNM[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b10 to b0	FRNM[10:0]	Frame Number	Latest frame number	R
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	CRCE	Receive Data Error	0: No error 1: An error occurred	R/W*1
b15	OVRN	Overrun/Underrun Detection Status	0: No error 1: An error occurred	R/W*1

Note 1. To clear the status, write 0 only to the bits to be cleared. Write 1 to the other bits.

FRNM[10:0] bits (Frame Number)

The USBFS sets these bits to indicate the latest frame number, which is updated every 1 ms, when an SOF packet is issued or received.

CRCE bit (Receive Data Error)

The CRCE bit is set to 1 when a CRC error or bit stuffing error occurs during isochronous transfer. On detecting a CRC error in host controller mode, the USBFS generates an internal NRDY interrupt.

To clear the CRCE bit, write 0 to it while writing 1 to the other bits in the FRMNUM register.

OVRN bit (Overrun/Underrun Detection Status)

The OVRN bit is set to 1 when an overrun or underrun error occurs during isochronous transfer. To clear the bit, write 0 to it while writing 1 to the other bits in the FRMNUM register.

In host controller mode, the OVRN bit is set to 1 on any of the following conditions:

- For a transmitting isochronous pipe, the time to issue an OUT token comes before all of the transmit data is written to the FIFO buffer.
- For a receiving isochronous pipe, the time to issue an IN token comes when no FIFO buffer planes are empty.

In device controller mode, the OVRN bit is set to 1 on any of the following conditions:

- For a transmitting isochronous pipe, the IN token is received before all of the transmit data is written to the FIFO buffer.
- For a receiving isochronous pipe, the OUT token is received when no FIFO buffer planes are empty.

28.2.19 USB Request Type Register (USBREQ)

Address(es): USBFS.USBREQ 4009 0054h

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

Bit	Symbol	Bit name	Description	R/W
b7 to b0	BMREQUESTTYPE[7:0]	Request Type	USB request bmRequestType value.	R/W *1
b15 to b8	BREQUEST[7:0]	Request	USB request bRequest value.	R/W *1

Note 1. In device controller mode, these bits are readable, but writing to them has no effect. In host controller mode, these bits are readable and writable.

USBREQ stores setup requests for control transfers.

In device controller mode, the USBREQ stores the received bRequest and bmRequestType values. In host controller mode, it sets the bRequest and bmRequestType values to be transmitted.

USBREQ is initialized by a USB bus reset.

BMREQUESTTYPE[7:0] bits (Request Type)

The BMREQUESTTYPE[7:0] bits hold the bmRequestType value of USB requests.

- In host controller mode:
  Set these bits to the value of the USB request data in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the value of the USB request data in reception setup transactions. Writing to the bits has no effect.

BREQUEST[7:0] bits (Request)

These bits store bRequest value of the USB request.

- In host controller mode:
  Set these bits to the value of the USB request data in setup transmission transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the value of the USB request data in reception setup transactions. Writing to the bits has no effect.

28.2.20 USB Request Value Register (USBVAL)

Address(es): USBFS.USBVAL 4009 0056h

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

Bit	Symbol	Bit name	Description	R/W
b15 to b0	WVALUE[15:0]	Value	These bits store the USB request wValue value.	R/W *1

Note 1. In device controller mode, these bits are readable, but writing to them has no effect. In host controller mode, these bits are both readable and writable.

In device controller mode, USBVAL stores the received wValue value. In host controller mode, it is set to the wValue value to be transmitted is set.

USBVAL is initialized by a USB bus reset.

WVALUE[15:0] bits (Value)

These bits store wValue value of the USB request.

- In host controller mode:
  Set these bits to the value of the wValue field in USB requests of transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the wValue value of USB requests in reception setup transactions. Writing to the bits has no effect.

28.2.21 USB Request Index Register (USBINDX)

Address(es): USBFS.USBINDX 4009 0058h

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

Bit	Symbol	Bit name	Description	R/W
b15 to b0	WINDEX[15:0]	Index	These bits store the USB request wIndex value.	R/W *1

Note 1. In device controller mode, these bits are readable, but writing to them has no effect. In host controller mode, these bits are both readable and writable.

USBINDX stores setup requests for control transfers. In device controller mode, it stores the received wIndex value. In host controller mode, it is set to the wIndex value to be transmitted. USBINDX is initialized by a USB bus reset.

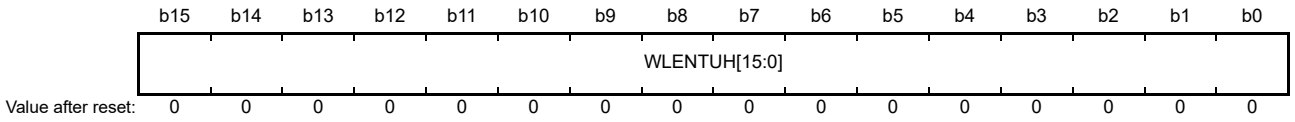
WINDEX[15:0] bits (Index)

These bits hold the value of a USB request.

- In host controller mode:
  Set these bits to the wIndex value in USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the wIndex value in USB requests received in reception setup transactions. Writing to the bits has no effect.

28.2.22 USB Request Length Register (USBLENG)

Address(es): USBFS.USBLENG 4009 005Ah



Bit	Symbol	Bit name	Description	R/W
b15 to b0	WLENTUH[15:0]	Length	These bits store the USB request wLength value.	R/W*1

Note 1. In device controller mode, these bits are readable, but writing to them has no effect. In host controller mode, these bits are both readable and writable.

USBLENG stores setup requests for control transfers.

When the device controller is selected, the value of wLength that is received is stored. In host controller mode, the value of wLength to be transmitted is set.

USBLENG is initialized by a USB bus reset.

WLENTUH[15:0] bits (Length)

These bits hold the wLength value of a USB request.

- In host controller mode:
  Set these bits to the wLength value in USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:
  These bits indicate the wLength value in USB requests received in reception setup transactions. Writing to the bits has no effect.

28.2.23 DCP Configuration Register (DCPCFG)

Address(es): USBFS.DCPCFG 4009 005Ch

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

Bit	Symbol	Bit name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DIR	Transfer Direction*1	0: Data receiving direction 1: Data transmitting direction.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Pipe continued at the end of transfer 1: Pipe disabled at the end of transfer.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only set this bit while the PID is NAK. Before setting this bit, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

DIR bit (Transfer Direction)

In host controller mode, the DIR bit sets the transfer direction of the data stage and status stage for control transfers. In device controller mode, set the DIR bit to 0.

SHTNAK bit (Pipe Disabled at End of Transfer)

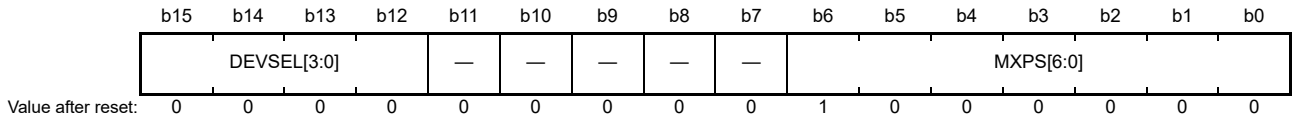
The SHTNAK bit specifies whether to change PID to NAK on transfer end when the selected pipe is receiving. It is only valid when the selected pipe is receiving.

When the SHTNAK bit is 1, the USBFS changes the DCPCTR.PID[1:0] bits for the DCP to NAK on determining that the transfer has ended. The USBFS determines transfer end on the following condition:

- A short packet, including a zero-length packet, is successfully received.

28.2.24 DCP Maximum Packet Size Register (DCPMAXP)

Address(es): USBFS.DCPMAXP 4009 005Eh



Bit	Symbol	Bit name	Description	R/W
b6 to b0	MXPS[6:0]	Maximum Packet Size*1	These bits set the maximum amount of data (maximum packet size) in payloads for the DCP. b6 b0 0 0 0 1 0 0 0: 8 bytes 0 0 1 0 0 0 0: 16 bytes 0 0 1 1 0 0 0: 24 bytes 0 1 0 0 0 0 0: 32 bytes 0 1 0 1 0 0 0: 40 bytes 0 1 1 0 0 0 0: 48 bytes 0 1 1 1 0 0 0: 56 bytes 1 0 0 0 0 0 0: 64 bytes 1 0 0 1 0 0 0: 72 bytes 1 0 1 0 0 0 0: 80 bytes 1 0 1 1 0 0 0: 88 bytes 1 1 0 0 0 0 0: 96 bytes 1 1 0 1 0 0 0: 104 bytes 1 1 1 0 0 0 0: 112 bytes 1 1 1 1 0 0 0: 120 bytes Other settings are prohibited.	R/W
b11 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*2	b15 b12 0 0 0 0: Address 0000 0 0 0 1: Address 0001 0 0 1 0: Address 0010 0 0 1 1: Address 0011 0 1 0 0: Address 0100 0 1 0 1: Address 0101 Other settings are prohibited.	R/W

Note 1. Only set the MXPS[6:0] bits while PID is NAK. Before setting these bits, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary. After the MXPS[6:0] bits are set and the DCP is set to the CURPIPE[3:0] bits in a port select register, clear the buffer by setting the BCLR bit the port control register to 1.

Note 2. Only set the DEVSEL[3:0] bits while PID is NAK and the DCPCTR.SUREQ bit is 0. Before setting these bits, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

MXPS[6:0] bits (Maximum Packet Size)

The MXPS[6:0] bits specify the maximum amount of data (maximum packet size) in payloads for the DCP. The initial value of the bits is 40h (64 bytes). Set the bits to a USB 2.0-compliant value. Do not write to the FIFO buffer or set PID = BUF while MXPS[6:0] is set to 0.

DEVSEL[3:0] bits (Device Select)

In host controller mode, these bits specify the address of the target peripheral device for a control transfer. Set up the device address in the associated DEVADDn (n = 0 to 5) register first, and then set these bits to the corresponding value. To set the DEVSEL[3:0] bits to 0010b, for example, first set the address in the DEVADD2 register.

In device controller mode, set these bits to 0000b.

28.2.25 DCP Control Register (DCPCTR)

Address(es): USBFS.DCPCTR 4009 0060h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	SUREQ	—	—	SUREQ CLR	—	—	SQCLR	SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response.	R/W
b2	CCPL	Control Transfer End Enable	0: Invalid 1: Control transfer completion enabled.	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy	0: DCP not used for the transaction 1: DCP in used for the transaction.	R
b6	SQMON	Sequence Toggle Bit Monitor	0: DATA0 1: DATA1.	R
b7	SQSET	Sequence Toggle Bit Set*2	Sets the sequence toggle bit in DCP transfers: 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1.	R/W*1
b8	SQCLR	Sequence Toggle Bit Clear*2	Clears the sequence toggle bit in DCP transfers: 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0. This bit is read as 0.	R/W*1
b10, b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	SUREQCLR	SUREQ Bit Clear	Clears the SUREQ bit in host controller mode: 0: Invalid (writing 0 has no effect) 1: Clear SUREQ to 0. This bit is read as 0.	R/W
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	SUREQ	Setup Token Transmission	Sets up token transmission in host controller mode: 0: Invalid (writing 0 has no effect) 1: Transmit setup packet.	R/W
b15	BSTS	Buffer Status	0: Buffer access disabled 1: Buffer access enabled.	R

Note 1. This bit is read as 0.

Note 2. Write 1 to the SQSET and SQCLR bits while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

PID[1:0] bits (Response PID)

The PID[1:0] bits control the USB response type during control transfers.

In host controller mode, to change the PID[1:0] setting from NAK to BUF:

- When the transmitting direction is set:
  a. Write all of the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and PID is NAK.
  b. Set PID[1:0] bits to 01b (BUF).
  The USBFS then executes the OUT transaction.

- When the receiving direction is set:
  a. Check that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and PID is NAK.
- Set PID[1:0] bits to 01b (BUF).
  The USBFS then executes the IN transaction.

The USBFS changes the PID[1:0] setting as follows:

- When the PID[1:0] bits are set to BUF (01b) by the software and the USBFS has received data exceeding MaxPacketSize, the USBFS sets the PID[1:0] to STALL (11b).
- When a reception error, such as a CRC error, is detected three times consecutively, the USBFS sets the PID[1:0] bits to NAK (00b).
- On receiving the STALL handshake, the USBFS sets PID[1:0] to STALL (11b).

In device controller mode, the USBFS changes the PID[1:0] setting as follows:

- On receiving a setup packet, the USBFS sets PID[1:0] to NAK (00b). The USBFS then sets the INTSTS0.VALID flag to 1, and the PID[1:0] setting cannot be changed until the software clears the VALID flag to 0.
- When the PID[1:0] bits are set to BUF (01b) by the software and the USBFS has received data exceeding MaxPacketSize, the USBFS sets PID[1:0] to STALL (11b).
- On detecting a control transfer sequence error, the USBFS sets PID[1:0] to STALL (1xb).
- On detecting a USBFS bus reset, the USBFS sets PID[1:0] to NAK.

The USBFS does not check the PID[1:0] setting while processing a SET_ADDRESS request.

The PID[1:0] bits are initialized by a USB bus reset.

CCPL bit (Control Transfer End Enable)

In device controller mode, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed. When the bit is set to 1 by the software while the associated PID[1:0] bits are set to BUF, the USBFS completes the control transfer status stage.

During control read transfers, the USBFS transmits the ACK handshake in response to the OUT transaction from the USB host. During control write or no-data control transfers, it transmits the zero-length packet in response to the IN transaction from the USB host. On detecting a SET_ADDRESS request, the USBFS operates in auto response mode from the setup stage up to status stage completion regardless of the CCPL bit setting.

The USBFS changes the CCPL bit from 1 to 0 on receiving a new setup packet. The software cannot write 1 to the bit while the INTSTS0.VALID bit is 1. The bit is initialized by a USB bus reset.

In host controller mode, always write 0 to the CCPL bit.

PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether DCP is used for the transaction when USBFS changes the PID[1:0] bits from BUF to NAK. The USBFS changes the PBUSY bit from 0 to 1 on start of a USBFS transaction for the selected pipe. It changes the PBUSY bit from 1 to 0 on completion of one transaction.

After PID is set to NAK by the software, the value in the PBUSY bit indicates whether changes to pipe settings can proceed.

For details, see section 28.3.4.1, Pipe Control Register Switching Procedures.

SQMON bit (Sequence Toggle Bit Monitor)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction during a DCP transfer.

The USBFS toggles the bit on normal completion of the transaction. It does not toggle the bit, however, when a DATA-PID mismatch occurs during a transfer in the receiving direction.

In device controller mode, the USBFS sets the SQMON bit to 1 (specifies DATA1 as the expected value) on successful reception of the setup packet.

In device controller mode, the USBFS does not reference this bit during IN or OUT transactions at the status stage, and it does not toggle the bit on normal completion.

SQSET bit (Sequence Toggle Bit Set)

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SQCLR bit (Sequence Toggle Bit Clear)

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer. It is read as 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SUREQCLR bit (SUREQ Bit Clear)

In host controller mode, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The bit is read as 0.

If transfer stops while the SUREQ bit is set to 1 in a setup transaction, set the SUREQCLR bit to 1 by the software. This is not necessary at the end of a normal setup transaction, because the USBFS automatically clears the SUREQ bit to 0.

Only control the SUREQ bit through the SUREQCLR bit while the DVSTCTR0.UACT bit is 0. When UACT is 0, communication is halted or no transfer is occurring because a bus disconnection was detected.

In device controller mode, always write 0 to this bit.

SUREQ bit (Setup Token Transmission)

In host controller mode, setting the SUREQ bit to 1 triggers the USBFS to transmit the setup packet. After completing the setup transaction process, the USBFS generates either the SACK or SIGN interrupt and clears the SUREQ bit to 0. The USBFS also clears the SUREQ bit to 0 when the software sets the SUREQCLR bit to 1.

Before setting the SUREQ bit to 1, set the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the wanted USB request in the setup transaction. Also check that the PID[1:0] bits for the DCP are set to NAK. After setting the SUREQ bit to 1, do not change the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is complete (SUREQ bit = 1). Write 1 to the SUREQ bit only when transmitting the setup token. Otherwise, write 0.

In device controller mode, always write 0 to this bit.

BSTS bit (Buffer Status)

This bit indicates the status of access to the DCP FIFO buffer. The meaning of this bit varies as follows depending on the CFIFOSEL.ISEL setting:

- When ISEL = 0, the bit indicates whether receive data can be read from the buffer
- When ISEL = 1, the bit indicates whether transmit data can be written to the buffer.

28.2.26 Pipe Window Select Register (PIPESEL)

Address(es): USBFS.PIPESEL 4009 0064h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	PIPESEL[3:0]	Pipe Window Select	b3 b0 0 0 0 0: No pipe selected 0 0 0 1: PIPE1 0 0 1 0: PIPE2 0 0 1 1: PIPE3 0 1 0 0: PIPE4 0 1 0 1: PIPE5 0 1 1 0: PIPE6 0 1 1 1: PIPE7 1 0 0 0: PIPE8 1 0 0 1: PIPE9 Other settings are prohibited.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set pipes 1 to 9 using the PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN registers (n = 0 to 9).

After selecting the pipe in the PIPESEL register, pipe functions must be set in the associated PIPECFG, PIPEMAXP, and PIPEPERI. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set independently of the pipe selection in this register.

PIPESEL[3:0] bits (Pipe Window Select)

The PIPESEL[3:0] bits select the pipe number associated with the PIPECFG, PIPEMAXP, and PIPEPERI registers used for data writing and reading. Selecting a pipe number in the PIPESEL[3:0] bits allows writing to and reading from PIPECFG, PIPEMAXP, and PIPEPERI associated with the selected pipe number.

When PIPESEL[3:0] = 0000b, 0 is read from all of the bits in PIPECFG, PIPEMAXP, and PIPEPERI. Writing to these bits is invalid.

28.2.27 Pipe Configuration Register (PIPECFG)

Address(es): USBFS.PIPECFG 4009 0068h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TYPE[1:0]		—	—	—	BFRE	DBLB	—	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	EPNUM[3:0]	Endpoint Number*1	Specifies the endpoint number for the selected pipe. Setting 0000b indicates the pipe is not used.	R/W
b4	DIR	Transfer Direction*2,*3	0: Receiving direction 1: Transmitting direction.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Continue pipe operation after transfer ends 1: Disable pipe operation after transfer ends.	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9	DBLB	Double Buffer Mode*2,*3	0: Single buffer 1: Double buffer.	R/W
b10	BFRE	BRDY Interrupt Operation Specification*2,*3	0: BRDY interrupt on transmitting or receiving data 1: BRDY interrupt on completion of reading data.	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	TYPE[1:0]	Transfer Type*1	• Pipes 1 and 2 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Isochronous transfer. • Pipes 3 to 5 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Setting prohibited. • Pipes 6 to 9 b15 b14 0 0: Pipe not used 0 1: Setting prohibited 1 0: Interrupt transfer 1 1: Setting prohibited.	R/W

Note 1. Only set the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while PID is NAK. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 2. Only set the BFRE, DBLB, and DIR bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 3. To modify the BFRE, DBLB, or DIR bits after completing USB communication on the selected pipe, in addition to the constraints described in note 2, write 1 and 0 to the PIPEnCTR.ACLRM bit continuously by the software and clear the FIFO buffer assigned to the pipe.

PIPECFG specifies the transfer type, FIFO buffer access direction, and endpoint numbers for pipes 1 to 9. It also selects single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.

EPNUM[3:0] bits (Endpoint Number)

The EPNUM[3:0] bits specify the endpoint number for the selected pipe. Setting 0000b indicates the pipe is not used.

Set these bits so that the combination of the DIR and EPNUM[3:0] settings is different from those for other pipes. The EPNUM[3:0] bits can be set to 0000b for all pipes.

DIR bit (Transfer Direction)

The DIR bit specifies the transfer direction for the selected pipe.

When the software sets this bit to 0, the USBFS uses the selected pipe for receiving. When the software sets this bit to 1, the USBFS uses the selected pipe for transmitting.

SHTNAK bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to change the PIPEnCTR.PID[1:0] bits to 00b (NAK) at the end of transfer when the selected pipe is set in the receiving direction. The bit is valid for pipes 1 to 5 in the receiving direction.

When the software sets this bit to 1 for a receiving pipe, the USBFS changes the associated PIPEnCTR.PID[1:0] bits to 00b (NAK) on determining the transfer end. The USBFS determines that the transfer has ended on the following conditions:

- A short packet (including a zero-length packet) is successfully received
- The transaction counter is used and the number of packets specified for the transaction counter are successfully received.

DBLB bit (Double Buffer Mode)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe. The bit is valid when pipes 1 to 5 are selected.

BFRE bit (BRDY Interrupt Operation Specification)

The BFRE bit specifies the BRDY interrupt generation timing from the USBFS to the CPU for the selected pipe.

When the software sets the BFRE bit 1 and the selected pipe is in the receiving direction, the USBFS detects the transfer completion and generates the BRDY interrupt on reading the packet.

When a BRDY interrupt is generated with this setting, the software must write 1 to the BCLR bit in the port control register. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.

When the BFRE bit is set to 1 by the software and the selected pipe is in the transmitting direction, the USBFS does not generate the BRDY interrupt. For details, see section 28.3.3.1, BRDY Interrupt.

TYPE[1:0] bits (Transfer Type)

The TYPE[1:0] bits specify the transfer type for the pipe selected in the PIPESEL.PIPESEL[3:0] bits. Before setting PID to BUF and starting USB communication on the selected pipe, set the TYPE[1:0] bits to a value other than 00b.

28.2.28 Pipe Maximum Packet Size Register (PIPEMAXP)

Address(es): USBFS.PIPEMAXP 4009 006Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEVSEL[3:0]				—	—	—	MXPS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0/1 *1	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	MXPS[8:0]	Maximum Packet Size*2	• Pipes 1 and 2: 1 byte (001h) to 256 bytes (100h) • Pipes 3 to 5: 8 bytes (008h), 16 bytes (010h) 32 bytes (020h), 64 bytes (040h) (Bits [8:7] and [2:0] not supported.) • Pipes 6 to 9: 1 byte (001h) to 64 bytes (040h) (Bits [8:7] not supported.)	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*3	b3 b0 0 0 0 0: Address 0000 0 0 0 1: Address 0001 0 0 1 0: Address 0010 0 0 1 1: Address 0011 0 1 0 0: Address 0100 0 1 0 1: Address 0101. Other settings are prohibited.	R/W

Note 1. The value of the MXPS[8:0] bits is 000h when no pipe is selected in the PIPESEL.PIPESEL[3:0] bits and 040h when a pipe is selected.

Note 2. Only set the MXPS[8:0] bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 3. Only set the DEVSEL[3:0] bits while PID is NAK. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PIPEMAXP specifies the maximum packet size for pipes 1 to 9.

MXPS[8:0] bits (Maximum Packet Size)

The MXPS[8:0] bits specify the maximum data payload (maximum packet size) for the selected pipe.

Set these bits to the appropriate value for each transfer type based on the USB 2.0 specification. When MXPS[8:0] = 0, do not write to the FIFO buffer or set PID to BUF. These writes have no effect.

DEVSEL[3:0] bits (Device Select)

In host controller mode, the DEVSEL[3:0] bits specify the address of the target device for USB communication. Set up the device address in the associated DEVADDn (n = 0 to 5) register first, and then set these bits to the corresponding value. To set the DEVSEL[3:0] bits to 0010b, for example, first set the address in the DEVADD2 register.

In device controller mode, set these bits to 0000b.

28.2.29 Pipe Cycle Control Register (PIPEPERI)

Address(es): USBFS.PIPEPERI 4009 006Eh

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

Bit	Symbol	Bit name	Description	R/W
b2 to b0	IITV[2:0]*1	Interval Error Detection Interval	Specifies the interval error detection timing for the selected pipe as nth power of 2 of the frame timing	R/W
b11 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	IFIS	Isochronous IN Buffer Flush	0: The buffer not flushed 1: The buffer is flushed.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only set the IITV[2:0] bits while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PIPEPERI selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfers, and sets the interval error detection interval for pipes 1 to 9.

IITV[2:0] bits (Interval Error Detection Interval)

To change the IITV[2:0] bits to another value after they are set and USB communication is performed, set the PIPEnCTR.PID[1:0] bits to 00b (NAK) and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer.

The IITV[2:0] bits are not provided for pipes 3 to 5. Write 000b to bit positions of the IITV[2:0] bits associated with pipes 3 to 5.

IFIS bit (Isochronous IN Buffer Flush)

The IFIS bit specifies whether to flush the buffer when the pipe selected in the PIPESEL.PIPESEL[3:0] bits is used for isochronous IN transfers.

In device controller mode when the selected pipe is for isochronous IN transfers, the USBFS automatically clears the FIFO buffer if the USBFS fails to receive the IN token from the USB host within the interval set in the IITV[2:0] bits in terms of frames.

When double buffering is specified (PIPECFG.DBLB = 1), the USBFS only clears the data in the previously used plane.

The USBFS clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USBFS expected to receive the IN token. Even if the SOF packet is corrupted, the FIFO buffer is cleared at the time the SOF packet is expected to be received by using the internal interpolation function.

When the host controller function is selected, set this bit is 0. Set this bit to 0 when the selected pipe is not for isochronous transfer.

28.2.30 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9)

PIPEnCTR (n = 1 to 5)

Address(es): USBFS.PIPE1CTR 4009 0070h, USBFS.PIPE2CTR 4009 0072h, USBFS.PIPE3CTR 4009 0074h, USBFS.PIPE4CTR 4009 0076h, USBFS.PIPE5CTR 4009 0078h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depends on the buffer state) 1 0: STALL response 1 1: STALL response.	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy	0: Pipe n not in use for the transaction 1: Pipe n in use for the transaction.	R
b6	SQMON	Sequence Toggle Bit Confirmation	0: DATA0 1: DATA1.	R
b7	SQSET	Sequence Toggle Bit Set*2	Sets the sequence toggle bit for pipe n: 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1. This bit is read as 0.	R/W*1
b8	SQCLR	Sequence Toggle Bit Clear*2	Clears the sequence toggle bit for pipe n: 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0. This bit is read as 0.	R/W*1
b9	ACLRM	Auto Buffer Clear Mode*3	0: Disabled 1: Enabled (all buffers initialized).	R/W
b10	ATREPM	Auto Response Mode*2	0: Auto response disabled 1: Auto response enabled.	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	INBUFM	Transmit Buffer Monitor	0: No data to be transmitted is in the FIFO buffer 1: Data to be transmitted is in the FIFO buffer.	R
b15	BSTS	Buffer Status	0: Buffer access by the CPU disabled 1: Buffer access by the CPU enabled.	R

Note 1. Only 0 can be read.

Note 2. Only set the ATREPM bit or write 1 to the SQCLR or SQSET bit while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 3. Only set the ACLRM bit while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting this bit, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PIPEnCTR can be set for any pipe selection in the PIPESEL register.

PID[1:0] bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the selected pipe.

The default PID[1:0] setting is NAK. Change the PID[1:0] setting to BUF to use the associated pipe for USB transfer. Table 28.7 and Table 28.8 show the basic operations of the USBFS (when there are no errors in the communication packets) based on the PID[1:0] bit setting. After changing the PID[1:0] setting from BUF to NAK through the software during USB communication on the selected pipe, check that the PBUSY bit is 1 to see if USB transfer on the pipe has actually entered the NAK state. If the USBFS changes the PID[1:0] bits to NAK, checking the PBUSY bit through the software is not necessary.

The USBFS changes the PIPEnCTR.PID[1:0] setting in the following cases:

- The USBFS sets PID to NAK on recognizing completion of the transfer when the selected pipe is in the receiving direction and the PIPECFG.SHTNAK bit for the selected pipe is set to 1 by the software
- The USBFS sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the selected pipe
- The USBFS sets PID to NAK on detecting a USB bus reset in device controller mode
- The USBFS sets PID to NAK on detecting a reception error, such as a CRC error, three consecutive times in host controller mode
- The USBFS sets PID to STALL (11b) on receiving the STALL handshake in host controller mode.

To specify the response type, set the PID[1:0] bits as follows.

- To transition from NAK (00b) to STALL, set 10b
- To transition from BUF (01b) to STALL, set 11b
- To transition from STALL (11b) to NAK, set 10b and then 00b
- To transition from STALL to BUF, transition to NAK and then BUF.

Table 28.7 Operation of the USBFS based on the PID[1:0] setting in host controller mode

PID[1:0] value	Transfer type	Transfer direction (DIR bit)	USBFS operation
00b (NAK)	Does not depend on the setting.	Does not depend on the setting.	Does not issue tokens.
01b (BUF)	Bulk or interrupt	Does not depend on the setting.	Issues tokens when the DVSTCTR0.UACT bit is 1 and the FIFO buffer associated with the selected pipe is ready for transmission and reception. Does not issue tokens when the DVSTCTR0.UACT bit is 0 or the FIFO buffer associated with the selected pipe is not ready for transmission or reception.
	Isochronous	Does not depend on the setting.	Issues tokens regardless of the status of the FIFO buffer associated with the selected pipe.
10b (STALL) or 11b (STALL)	Does not depend on the setting.	Does not depend on the setting.	Does not issue tokens.

Table 28.8 Operation of the USBFS based on the PID[1:0] setting in device controller mode (1 of 2)

PID[1:0] value	Transfer type	Transfer direction (DIR bit)	USBFS operation
00b (NAK)	Bulk or interrupt	Does not depend on the setting.	Returns NAK in response to the token from the USB host.
	Isochronous	Does not depend on the setting.	Returns nothing in response to the token from the USB host.
01b (BUF)	Bulk	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception.
	Interrupt	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception.
	Bulk or interrupt	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the FIFO buffer associated with the selected pipe is ready for transmission. Otherwise, returns NAK.
	Isochronous	Receiving direction (DIR = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, discards the data.

Table 28.8 Operation of the USBFS based on the PID[1:0] setting in device controller mode (2 of 2)

PID[1:0] value	Transfer type	Transfer direction (DIR bit)	USBFS operation
01b (BUF)	Isochronous	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the associated FIFO buffer is ready for transmission. Otherwise, transmits a zero-length packet.
10b (STALL) or 11b (STALL)	Bulk or interrupt	Does not depend on the setting.	Returns STALL in response to the token from the USB host.
	Isochronous	Does not depend on the setting.	Returns nothing in response to the token from the USB host.

PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether the selected pipe is being used for the current transaction.

The USBFS changes the PBUSY bit from 0 to 1 on start of the USB transaction for the selected pipe, and changes the PBUSY bit from 1 to 0 on completion of one transaction.

Reading the PBUSY bit by the software after PID is set to NAK allows you to check whether changing the pipe setting is possible. For details, see section 28.3.4.1, Pipe Control Register Switching Procedures.

SQMON bit (Sequence Toggle Bit Confirmation)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction of the selected pipe.

When the selected pipe is not the isochronous transfer type, the USBFS toggles the SQMON bit on successful completion of the transaction. However, the USBFS does not toggle the SQMON flag when a DATA-PID mismatch occurs during transfer in the receiving direction.

SQSET bit (Sequence Toggle Bit Set)

Setting the SQSET bit to 1 through the software causes the USBFS to set DATA1 as the expected value of the sequence toggle bit for the next transaction on the selected pipe. The USBFS clears the SQSET bit to 0.

SQCLR bit (Sequence Toggle Bit Clear)

Setting the SQCLR bit to 1 through the software causes the USBFS to clear the expected value of the sequence toggle bit for the next transaction on the selected pipe to DATA0. The USBFS clears the SQCLR bit to 0.

ACLRM bit (Auto Buffer Clear Mode)

The ACLRM bit enables or disables auto buffer clear mode for the selected pipe. To completely clear the data in the FIFO buffer allocated to the selected pipe, write 1 and then 0 to the ACLRM bit continuously.

Table 28.9 shows the data cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which this processing is required.

Table 28.9 Data cleared by the USBFS when ACLRM = 1

No.	Data cleared by setting the ACLRM bit	Situations requiring data clear
1	All data in the FIFO buffer allocated to the selected pipe (two FIFO buffers in double buffer mode)	When initializing the selected pipe
2	Interval count value when the selected pipe is the isochronous transfer type	When resetting the interval count value
3	Internal flags related to the PIPECFG.BFRE bit	When changing the PIPECFG.BFRE setting
4	FIFO buffer toggle control	When changing the PIPECFG.DBLB setting
5	Internal flags related to the transaction count	When forcing the transaction count function to terminate

ATREPM bit (Auto Response Mode)

The ATREPM bit enables or disables auto response mode for the selected pipe.

This bit can be set to 1 in device controller mode when the selected pipe is the bulk transfer type. When the bit is set to 1, the USBFS responds to the token from the USB host as follows:

- When the selected pipe is set for Bulk IN transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 1):
  a. When the ATREPM bit = 1 and PID = BUF, the USBFS transmits a zero-length packet in response to the IN token.
  b. The USBFS updates (allows toggling of) the sequence toggle bit (DATA-PID) each time the USBFS receives ACK from the USB host. In a single transaction, the IN token is received, a zero-length packet is transmitted, and then ACK is received. The USBFS does not generate the BRDY or BEMP interrupt.
- When the selected pipe is set for Bulk OUT transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 0):

  When the ATREPM bit = 1 and PID = BUF, the USBFS returns NAK in response to the OUT token and generates an NRDY interrupt.

For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode. When the selected pipe uses isochronous transfer, always set this bit to 0.

In host controller mode, always set the ATREPM bit to 0.

INBUFM bit (Transmit Buffer Monitor)

The INBUMFM bit indicates the FIFO buffer status for the selected pipe in the transmitting direction.

When the selected pipe is set in the transmitting direction (PIPECFG.DIR = 1), the USBFS sets this bit to 1 when the CPU or DMA/DTC completes writing data to at least one FIFO buffer plane.

The USBFS sets this bit to 0 when the USBFS completes transmitting the data from the FIFO buffer plane to which all the data is written. In double buffer mode (PIPECFG.DBLB = 1), the USBFS sets the INBUFM bit to 0 when the USBFS completes transmitting the data from the two FIFO buffer planes before the CPU or DMA/DTC completes writing data to one FIFO buffer plane.

The INBUFM bit indicates the same value as the BSTS bit when the selected pipe is in the receiving direction (PIPECFG.DIR = 0).

BSTS bit (Buffer Status)

The BSTS bit indicates the FIFO buffer status for the selected pipe.

The meaning of the BSTS bit depends on the PIPECFG.DIR, PIPECFG.BFRE, and DnFIFOSEL.DCLRM settings, as shown in Table 28.10.

Table 28.10 BSTS bit operation

DIR value	BFRE value	DCLRM value	BSTS bit function
0	0	0	Sets to 1 when receive data can be read from the FIFO buffer, and is set to 0 on completion of data read.
		1	Setting prohibited
	1	0	Sets to 1 when receive data can be read from the FIFO buffer, and is set to 0 when the software sets the BCLR bit in the port control register to 1 after the data read is complete.
		1	Sets to 1 when receive data can be read from the FIFO buffer, and is set to 0 on completion of data read.
1	0	0	Sets to 1 when transmit data can be written to the FIFO buffer, and is set to 0 on completion of data write.
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

PIPEnCTR (n = 6 to 9)

Address(es): USBFS.PIPE6CTR 4009 007Ah, USBFS.PIPE7CTR 4009 007Ch, USBFS.PIPE8CTR 4009 007Eh, USBFS.PIPE9CTR 4009 0080h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	—	—	—	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depends on the buffer state) 1 0: STALL response 1 1: STALL response.	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy	0: Pipe *n* not in use for the transaction. 1: Pipe *n* in use for the transaction.	R
b6	SQMON	Sequence Toggle Bit Confirmation	0: DATA0 1: DATA1.	R
b7	SQSET	Sequence Toggle Bit Set*2	Sets the sequence toggle bit for pipe n: 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1. This bit is read as 0.	R/W *1
b8	SQCLR	Sequence Toggle Bit Clear*2	Clears the sequence toggle bit for pipe n: 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0. This bit is read as 0.	R/W *1
b9	ACLRM	Auto Buffer Clear Mode*2,*3	0: Disabled 1: Enabled (all buffers are initialized).	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	BSTS	Buffer Status	0: Buffer access disabled 1: Buffer access enabled.	R

Note 1. Only 0 can be read. Only 1 can be written.

Note 2. Only write 1 to the SQCLR or SQSET bit while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 3. Only set the ACLRM bit while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting this bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PID[1:0] bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the selected pipe.

The default PID[1:0] setting is NAK. Change the PID[1:0] setting to BUF to use the associated pipe for USB transfer. Table 28.7 and Table 28.7 show the basic operation (when there are no errors in the transmitted and received packets) of the USB depending on the setting of the PID[1:0] bits.

After changing the PID[1:0] setting from BUF to NAK through the software during USB communication on the selected pipe, check that the PBUSY bit is 1 to see if USB transfer on the selected pipe has actually entered the NAK state. If the USBFS changes the PID[1:0] bits to NAK, checking the PBUSY bit through the software is not necessary.

The USBFS changes the PIPEnCTR.PID[1:0] setting in the following cases:

- The USBFS sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the selected pipe
- The USBFS sets PID to NAK on detecting a USB bus reset in device controller mode
- The USBFS sets PID to NAK on detecting a reception error, such as a CRC error, three consecutive times in host controller mode
- The USBFS sets PID to STALL (11b) on receiving the STALL handshake in host controller mode.

To specify each response type, set the PID[1:0] bits as follows:

- To transition from NAK (00b) to STALL, set 10b
- To transition from BUF (01b) to STALL, set 11b
- To transition from STALL (11b) to NAK, set 10b and then 00b
- To transition from STALL to BUF, set 00b (NAK) and then 01b (BUF).

PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether the selected pipe is being used for the current transaction.

The PBUSY bit indicates whether the selected pipe is being used for the current transaction.

The USBFS changes the PBUSY bit from 0 to 1 on start of the USB transaction for the selected pipe, and changes the PBUSY bit from 1 to 0 on completion of one transaction.

Reading the PBUSY bit by the software after PID is set to NAK allows you to check whether changing the pipe setting is possible.

SQMON bit (Sequence Toggle Bit Confirmation)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction of the selected pipe.

The USBFS toggles the SQMON flag on successful completion of the transaction. However, the USBFS does not toggle the SQMON bit when a DATA-PID mismatch occurs during transfer in the receiving direction.

SQSET bit (Sequence Toggle Bit Set)

Setting the SQSET bit to 1 through the software causes the USBFS to set DATA1 as the expected value of the sequence toggle bit for the next transaction on the selected pipe. The USBFS sets the SQSET bit to 0.

SQCLR bit (Sequence Toggle Bit Clear)

Setting the SQCLR bit to 1 through the software causes the USBFS to clear the expected value of the sequence toggle bit for the next transaction on the selected pipe to DATA0. The USBFS sets the SQCLR bit to 0.

ACLRM bit (Auto Buffer Clear Mode)

The ACLRM bit enables or disables auto buffer clear mode for the selected pipe. To completely clear the data in the FIFO buffer allocated to the selected pipe, write 1 and then 0 to the ACLRM bit continuously.

Table 28.11 shows the data cleared by writing 1 and 0 continuously to the ACLRM bit and the cases in which this processing is required.

Table 28.11 Data cleared by USBFS when ACLRM = 1

No.	Data cleared by setting the ACLRM bit	Situations requiring data clear
1	All data in the FIFO buffer allocated to the selected pipe	When initializing the selected pipe
2	The interval count value when the selected pipe is for interrupt transfer and the host controller is selected	When resetting the interval count value
3	Internal flags related to the PIPECFG.BFRE bit	When changing the PIPECFG.BFRE setting
4	Internal flags related to the transaction count	When forcing the transaction count function to terminate

BSTS bit (Buffer Status)

The BSTS bit indicates the FIFO buffer status for the selected pipe.

The meaning of the BSTS bit depends on the PIPECFG.DIR, PIPECFG.BFRE, and DnFIFOSEL.DCLRM settings, as shown in Table 28.10.

28.2.31 PIPEn Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5)

Address(es): USBFS.PIPE1TRE 4009 0090h, USBFS.PIPE2TRE 4009 0094h, USBFS.PIPE3TRE 4009 0098h, USBFS.PIPE4TRE 4009 009Ch, USBFS.PIPE5TRE 4009 00A0h

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

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TRCLR	Transaction Counter Clear	0: Invalid (writing 0 has no effect) 1: Clear the current counter value.	R/W
b9	TRENB	Transaction Counter Enable	0: Transaction counter disabled 1: Transaction counter enabled.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set each bit in PIPEnTRE while PID is NAK. Before setting these bits after changing the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

TRCLR bit (Transaction Counter Clear)

The USBFS bit clears the current value of the transaction counter associated with the selected pipe and then sets the TRCLR bit to 0.

TRENB bit (Transaction Counter Enable)

The TRENB bit enables or disables the transaction counter.

For receiving pipes, setting the TRENB bit to 1 after setting the total number of the packets to be received in the PIPEnTRN.TRNCNT[15:0] bits through the software allows the USBFS to control hardware as follows on having received the number of packets equal to the TRNCNT[15:0] setting.

- When the PIPECFG.SHTNAK bit is 1, the USBFS changes the PID bits to NAK for the associated pipe on having received the number of packets equal to the TRNCNT[15:0] setting.
- When the PIPECFG.BFRE bit is 1, the USBFS asserts the BRDY interrupt on having received the number of packets equal to the TRNCNT[15:0] setting and then reading the last received data.

For transmitting pipes, set the TRENB bit to 0.

When the transaction counter is not used, set this bit to 0. When the transaction counter is used, set the TRNCNT[15:0] bits before setting this bit to 1. Set this bit to 1 before receiving the first packet to be counted by the transaction counter.

28.2.32 PIPEn Transaction Counter Register (PIPEnTRN) (n = 1 to 5)

Address(es): USBFS.PIPE1TRN 4009 0092h, USBFS.PIPE2TRN 4009 0096h, USBFS.PIPE3TRN 4009 009Ah, USBFS.PIPE4TRN 4009 009Eh, USBFS.PIPE5TRN 4009 00A2h

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

Bit	Symbol	Bit name	Description	R/W
b15 to b0	TRNCNT[15:0]	Transaction Counter	• When written to: Specifies the total of packets (number of transactions) to be received by the selected pipe. • When read from: When PIPEnTRE.TRENB is 0, indicates the specified number of transactions. When PIPEnTRE.TRENB is 1, indicates the number of currently counted transactions.	R/W

The PIPEnTRN registers retain their current setting during a USB bus reset.

TRNCNT[15:0] bits (Transaction Counter)

The USBFS increments the value of the TRNCNT[15:0] bits by 1 when all of the following conditions are satisfied on receiving the packet:

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value ≠ current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the PIPEMAXP.MXPS[8:0] setting.

The USBFS clears the value of the TRNCNT[15:0] bits to 0 when any of the following conditions are satisfied:

All of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value = current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the PIPEMAXP.MXPS[8:0] setting.

All of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- The USBFS received a short packet.

All of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- The PIPEnTRE.TRCLR bit is set to 1 by software.

For transmitting pipes, set the TRNCNT[15:0] bits to 0. When the transaction counter is not used, set the TRNCNT[15:0] bits to 0.

Setting the number of transactions to be transferred to the TRNCNT[15:0] bits is only enabled when the PIPEnTRE.TRENB bit is 0. To set the number of transactions to be transferred, set the TRCLR bit to 1 to clear the current counter value before setting the PIPEnTRE.TRENB bit to 1.

28.2.33 Device Address n Configuration Register (DEVADDn) (n = 0 to 5)

Address(es): USBFS.DEVADD0 4009 00D0h, USBFS.DEVADD1 4009 00D2h, USBFS.DEVADD2 4009 00D4h, USBFS.DEVADD3 4009 00D6h, USBFS.DEVADD4 4009 00D8h, USBFS.DEVADD5 4009 00DAh

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

Bit	Symbol	Bit name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7, b6	USBSPD[1:0]	Transfer Speed of Communication Target Device	b7 b6 0 0: DEVADDn not used 0 1: Low-speed 1 0: Full-speed 1 1: Setting prohibited.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The DEVADDn register specifies the transfer speed of the peripheral device that is the communication target for pipes 0 to 9. In host controller mode, set all DEVADDn bits before starting communication to any pipes. Only change the bits in DEVADDn when no valid pipes are using the bit settings. A valid pipe is defined as one that satisfies both of the following conditions:

- DEVADDn is selected in the DEVSEL[3:0] bits
- The PID[1:0] bits are set to BUF for the selected pipe, or the selected pipe is the DCP with the DCPCTR.SUREQ bit set to 1.

In device controller mode, set all bits in this register to 0.

USBSPD[1:0] bits (Transfer Speed of Communication Target Device)

The USBSPD[1:0] bits specify the USB transfer speed of the target peripheral device. In host controller mode, the USBFS generates packets based on the USBSPD[1:0] setting. In device controller mode, set these bits to 00b.

28.2.34 USB Module Control Register (USBMC)

Address(es): USBFS.USBMC 4009 00CCh

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

Bit	Symbol	Bit name	Description	R/W
b0	VDDUSBE	USB Reference Power Supply Circuit On/Off Control	0: USB reference power supply circuit off 1: USB reference power supply circuit on.	R/W
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	VDCEN	USB Regulator On/Off Control	0: USB regulator off 1: USB regulator on.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

VDDUSBE bit (USB Reference Power Supply Circuit On/Off Control)

The USB reference power supply circuit generates the reference voltage for battery charging. Set this bit to 1 when using the battery charging function.

VDCEN bit (USB Regulator On/Off Control)

This bit is used to control the USB regulator circuit. Set this bit to 1 when using the USB regulator circuit.

28.2.35 BC Control Register 0 (USBBCCTRL0)

Address(es): USBFS.USBBCCTRL0 4009 00B0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PDDETSTS0	CHGDETSTS0	BATCHGE0	—	VDMSRCE0	IDPSINKE0	VDPSRCE0	IDMSINKE0	IDPSRCE0	RPDME0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RPDME0	D- Pin Pull-Down Control	0: Pull-down off 1: Pull-down on.	R/W
b1	IDPSRCE0	D+ Pin IDPSRC Output Control	0: Stop 1: 10 µA output.	R/W
b2	IDMSINKE0	D- Pin 0.6 V Input Detection (Comparator and Sink) Control	0: Detection off 1: Detection on (comparator and sink current on).	R/W
b3	VDPSRCE0	D+ Pin VDPSRC (0.6 V) Output Control	0: Stop 1: 0.6 V output.	R/W
b4	IDPSINKE0	D+ Pin 0.6 V Input Detection (Comparator and Sink) Control	0: Detection off 1: Detection on (comparator and sink current on).	R/W
b5	VDMSRCE0	D- Pin VDMSRC (0.6 V) Output Control	0: Stop 1: 0.6 V output.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	BATCHGE0	BC (Battery Charger) Function General Enable Control	0: Disabled 1: Enabled.	R/W
b8	CHGDETSTS0	D- Pin 0.6 V Input Detection Status*1	0: Not detected 1: Detected.	R
b9	PDDETSTS0	D+ Pin 0.6 V Input Detection Status*2	0: Not detected 1: Detected.	R
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Valid when IDMSINKE0 = 1.
Note 2. Valid when IDPSINKE0 = 1.

RPDME0 bit (D- Pin Pull-Down Control)

When using the battery charging function, set this bit to 1 to control the pull-down resistor of the D- pin.

IDPSRCE0 bit (D+ Pin IDPSRC Output Control)

With this bit set to 1 in device controller mode, the current output is enabled on detection of the data connection pin and the D+ pin is pulled up.

IDMSINKE0 bit (D- Pin 0.6 V Input Detection (Comparator and Sink) Control)

With this bit set to 1 in device controller mode, the USBFS detects whether VDMSRC (0.6 V) that is output from the host to D- on primary detection is connected, or whether VDPSRC (0.6 V) that is output from the function to D+ is connected to the function of D- through the host.

VDPSRCE0 bit (D+ Pin VDPSRC (0.6 V) Output Control)

With this bit set to 1 in device controller mode, output is enabled on primary detection and VDPSRC (0.6 V) is applied to D+.

IDPSINKE0 bit (D+ Pin 0.6 V Input Detection (Comparator and Sink) Control)

With this bit set to 1 in device controller mode, the USBFS detects whether VDMSRC (0.6 V) that is output from the function to D- is connected to the function of D+ (DCP) through the host. In host controller mode, the USBFS detects whether VDPSRC (0.6 V) that is output from the device to D+ when primary detection is connected.

VDMSRCE0 bit (D- Pin VDMSRC (0.6 V) Output Control)

With this bit set to 1 in device controller mode, output is enabled on secondary detection and VDMSRC (0.6 V) is applied to D-. In host controller mode, output is enabled on primary detection and VDMSRC (0.6 V) is applied to D-.

CHGDETSTS0 flag (D- Pin 0.6 V Input Detection Status)

In host controller mode, this flag is set to 1 if the USBFS detects whether VDMSRC (0.6 V) that is output from the host to D- during primary detection is connected, or whether VDPSRC (0.6 V) that is output from the function to D+ is connected to the function of D- through the host.

PDDETSTS0 flag (D+ Pin 0.6 V Input Detection Status)

In device controller mode, this flag is set to 1 if the USBFS detects whether VDMSRC (0.6 V) that is output from the function to D- during secondary detection is connected to the function of D+ (DCP) through the host.

In host controller mode, this bit is set to 1 if the USBFS detects whether VDPSRC (0.6 V) that is output from the function to D+ during primary detection is connected.

28.3 Operation

28.3.1 System Control

This section describes register settings required for initializing the USBFS and controlling power consumption.

28.3.1.1 Setting Data to USB-related Register

Setting the SYSCFG.USBE bit to 1 after starting the clock supply (SYSCFG.SCKE bit = 1) enables and starts USBFS operation.

28.3.1.2 Selecting the Controller Function

Use the SYSCFG.DCFM bit to select one of the USBFS functions. The DCFM bit must be changed in the initial settings immediately after a reset or in the D+ pull-up-disabled state (SYSCFG.DPRPU bit = 0) and D+ and D- pull-down-disabled state (SYSCFG.DRPD bit = 0).

28.3.1.3 Controlling the USB Data Bus Using Resistors

The USBFS provides pull-up and pull-down resistors for the D+ and D- lines. Pull these lines up or down by setting the SYSCFG.DPRPU, DMRPU, and DRPD bits.

In device controller mode, confirm that connection to the USB host is made, and then set the SYSCFG.DPRPU bit to 1 and pull up the D+ line (in full-speed communication), or set the SYSCFG.DMRPU bit to 1 and pull up the D-line (in low-speed communication).

When the SYSCFG.DPRPU (during full-speed) or the SYSCFG.DMRPU (during low-speed) bit is set to 0 during communication with a PC, the USBFS disables the pull-up resistor of the USB data line, thereby notifying the USB host of disconnection.

In host controller mode, set the SYSCFG.DRPD bit to 1 to pull down the D+ and D- lines.

Table 28.12 USB data bus resistor control

SYSCFG register settings					
DRPD bit	**DPRPU Bit**	**DMRPU Bit**	**D-**	**D+**	**Function**
0	0	0	Open	Open	When resistors not used
0	1	0	Open	Pull-up	When operating as the device controller at full-speed
0	0	1	Pull-up	Open	When operating as the device controller at low-speed
1	0	0	Pull-down	Pull-down	When operating as a host controller
Other settings			-	-	Setting prohibited

28.3.1.4 Example of USB Power Supply Connection

Figure 28.2 shows an example of power supply connection when the USB regulator is not used. Figure 28.3 and Figure 28.4 show examples of power supply connection when the USB regulator is used.

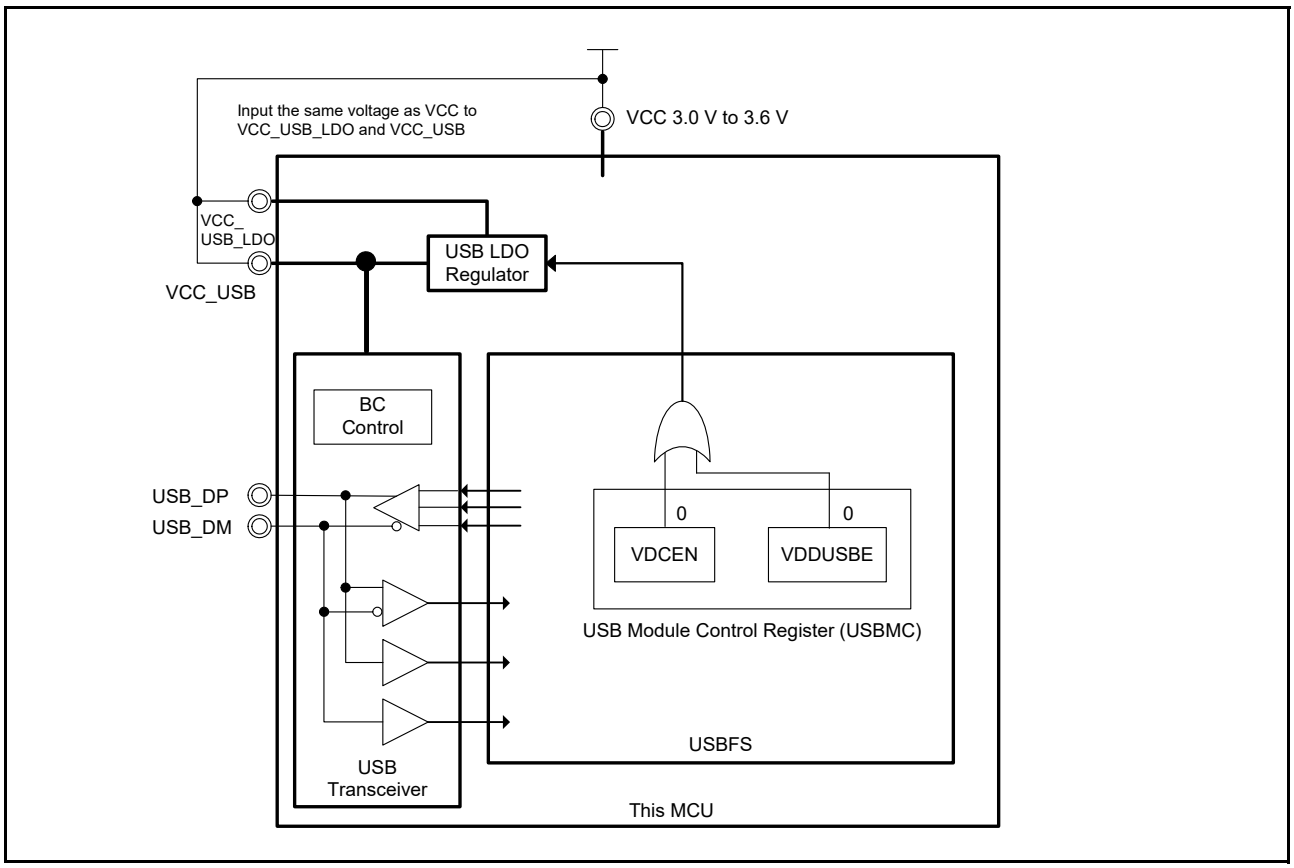
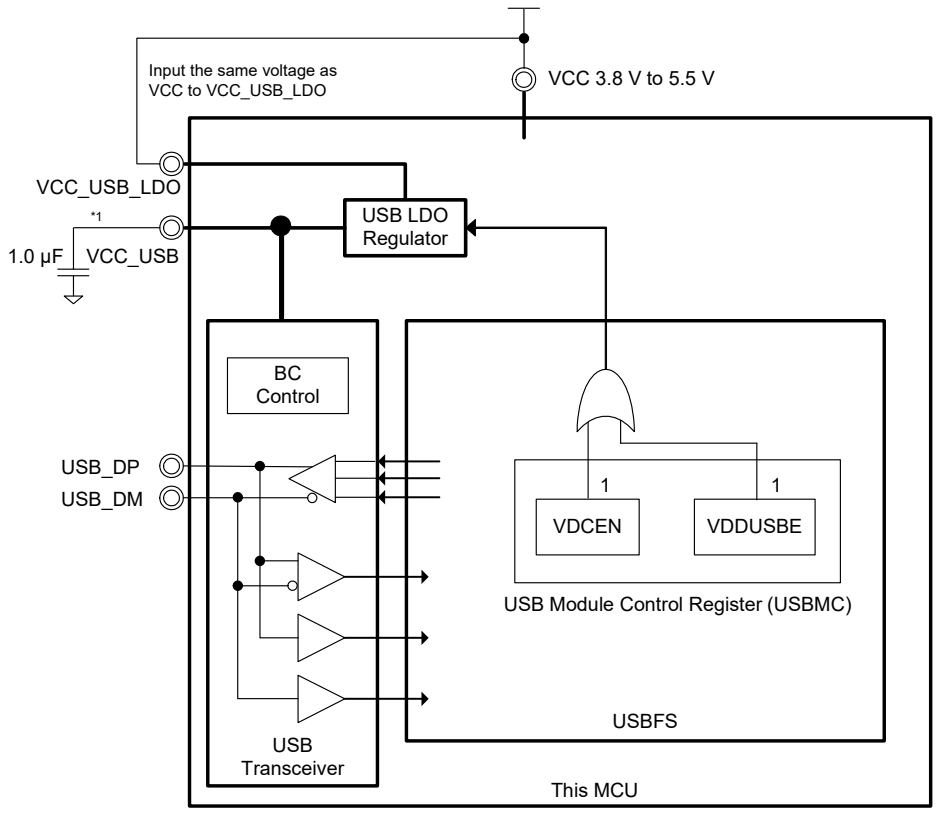
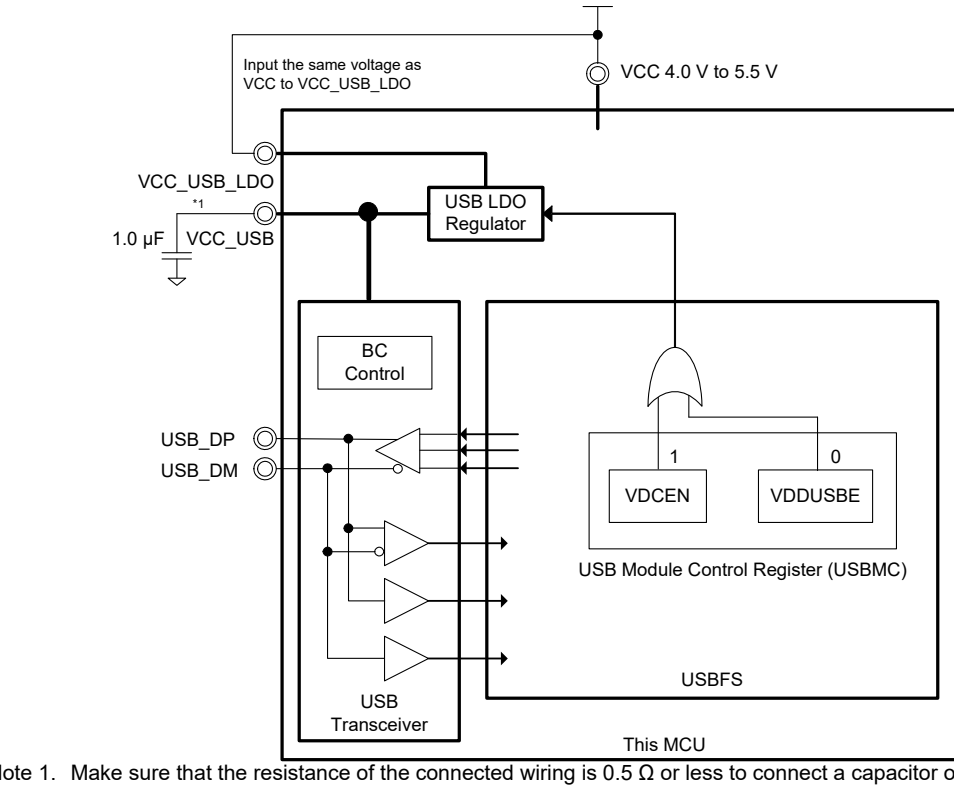


Figure 28.2 Example of power supply connection when the USB LDO regulator is not used



Note 1. Make sure that the resistance of the connected wiring is 0.5 Ω or less to connect a capacitor of ESR ≤ 1 Ω.

Figure 28.3 Example of power supply connection when the USB LDO regulator is used (BC used)



Note 1. Make sure that the resistance of the connected wiring is 0.5 Ω or less to connect a capacitor of ESR ≤ 1 Ω.

Figure 28.4 Example of power supply connection when the USB LDO regulator is used (BC not used)

28.3.1.5 Example of USB External Connection Circuits

The host recognizes a USB device when one of the data lines is pulled up. The MCU can use switching of the internal pull-up resistor for this. Also, bus-powered devices do not require external regulators because the MCU incorporates a power supply in the USB-PHY.

Figure 28.5 and Figure 28.6 show examples of external circuits for USB connection.

Figure 28.5 shows an example of OTG connection of the USB connector in the self-powered state.

The USBFS controls the pull-up resistor of the D+ line and the pull-down resistor of D+ and D- lines. Select pull-up and pull-down for the lines in the SYSCFG.DPRPU and SYSCFG.DRPD bits. In device controller mode, the pull-up resistor of USB data line is disabled if SYSCFG.DPRPU bit is set to 0 while communicating with the USB host. The USBFS can use this to notify the USB host of a device disconnect.

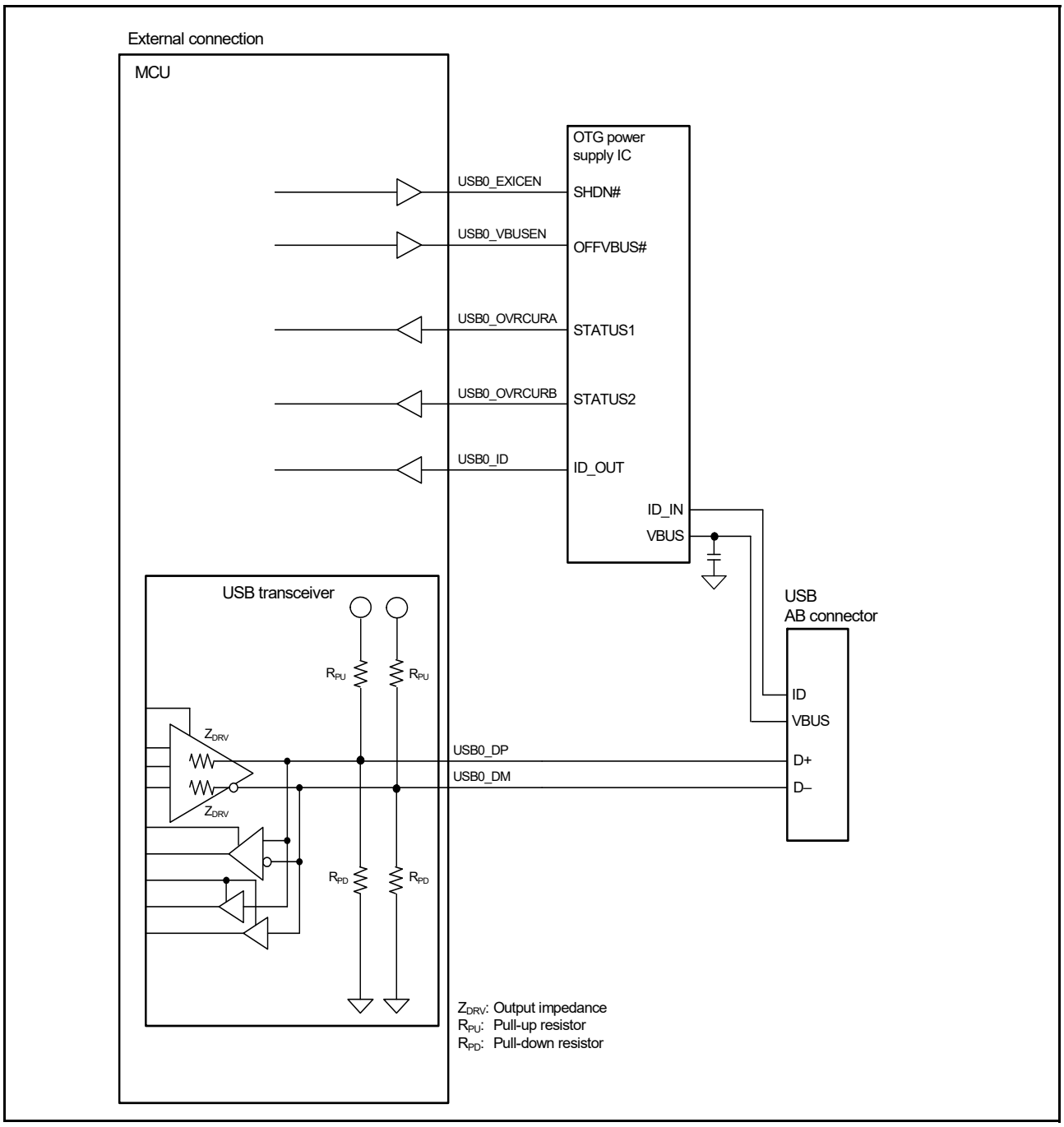


Figure 28.5 Example OTG connection in self-powered state

Figure 28.6 shows an example of functional connection of the USB connector in the self-powered state.

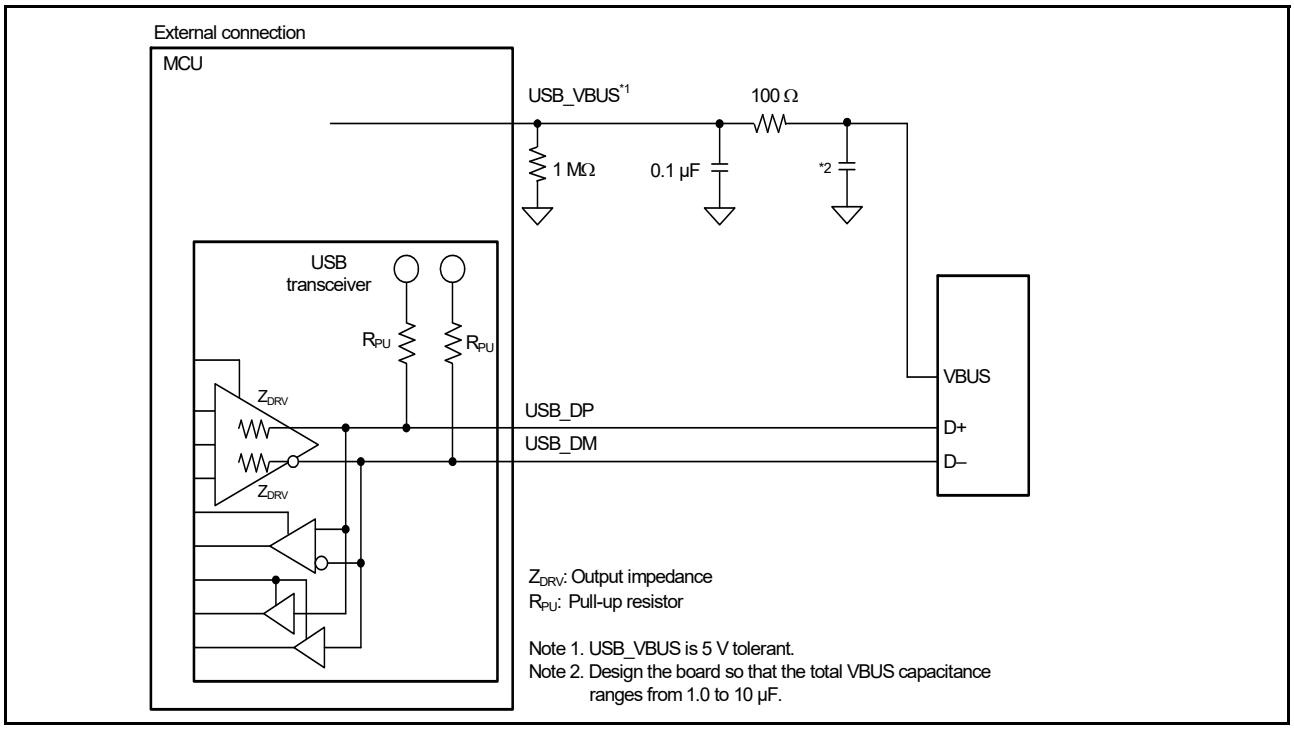


Figure 28.6 Example device connection in self-powered state

Figure 28.7 shows an example of host connection of the USB connector.

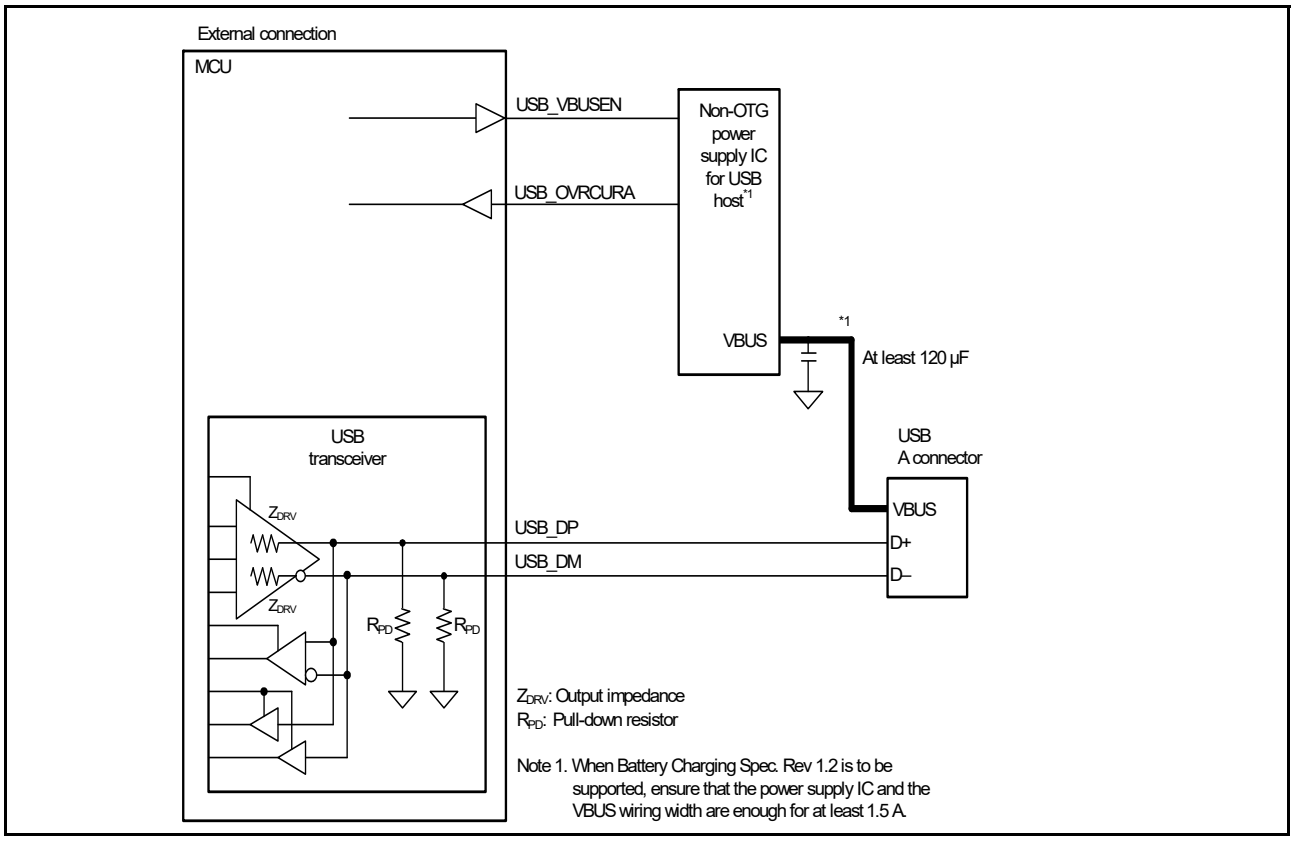


Figure 28.7 Example host connection

Figure 28.8 shows an example of functional connection of the USB connector in bus-powered state.

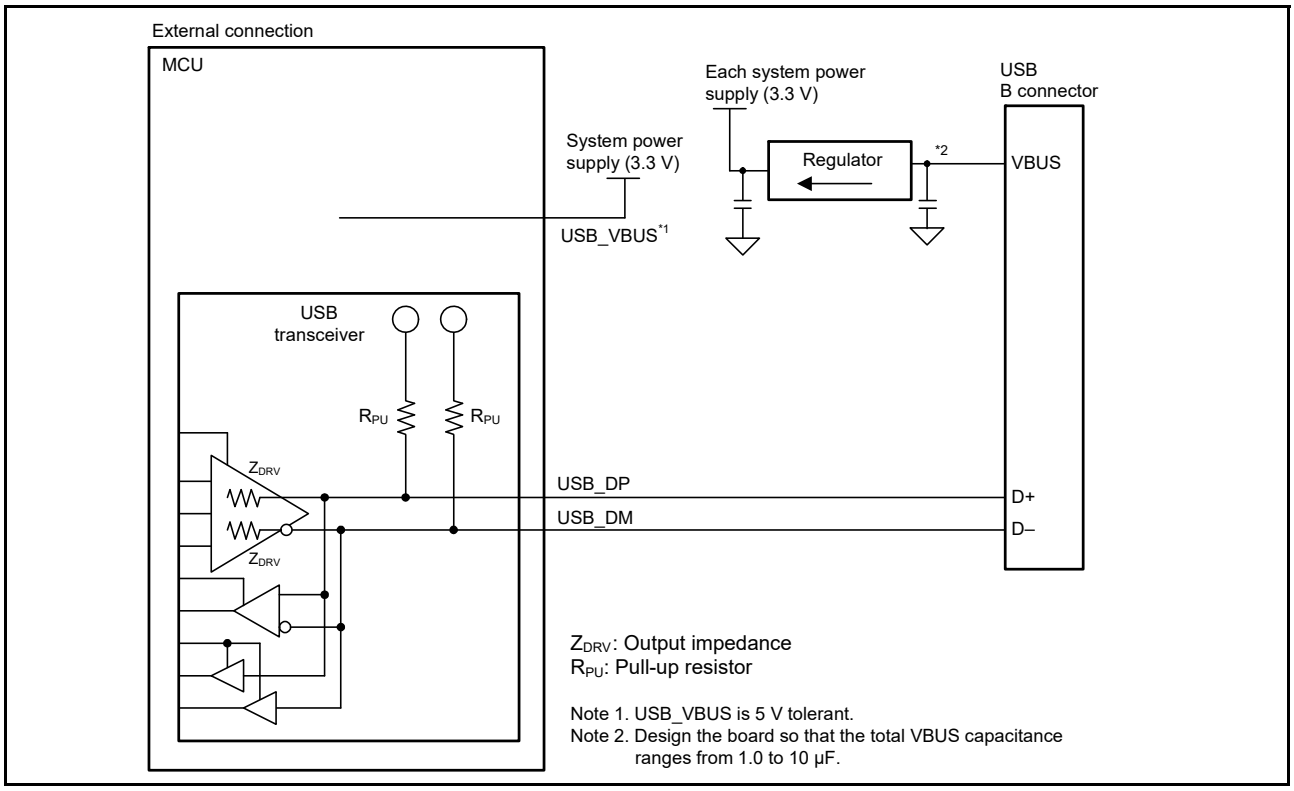


Figure 28.8 Example device connection in bus-powered state

The examples of external circuits given in this section are simplified circuits, and their operation in every system is not guaranteed.

Figure 28.9 shows an example of functional connection of the USB connector in bus-powered state 2.

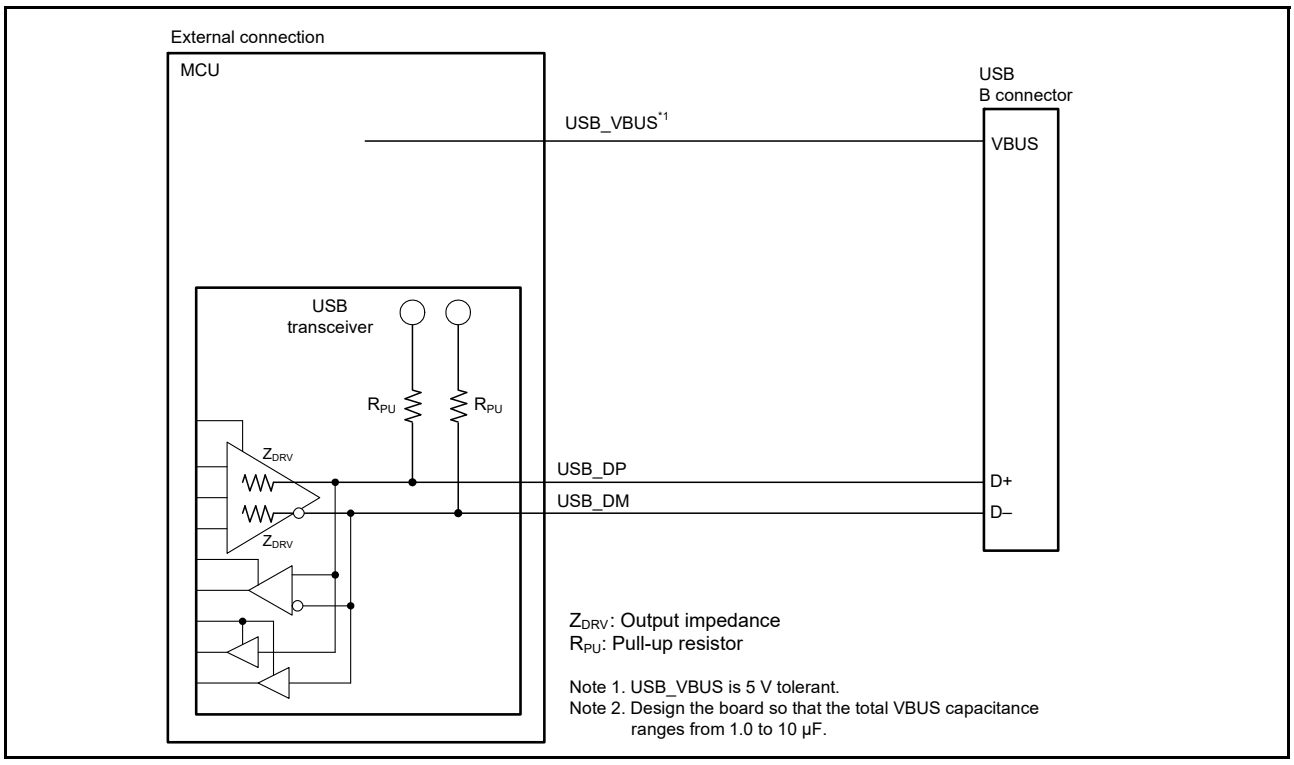


Figure 28.9 Example device connection in bus-powered state 2

Figure 28.10 shows an example of functional connection of the USB connector with Battery Charging Rev 1.2 supported.

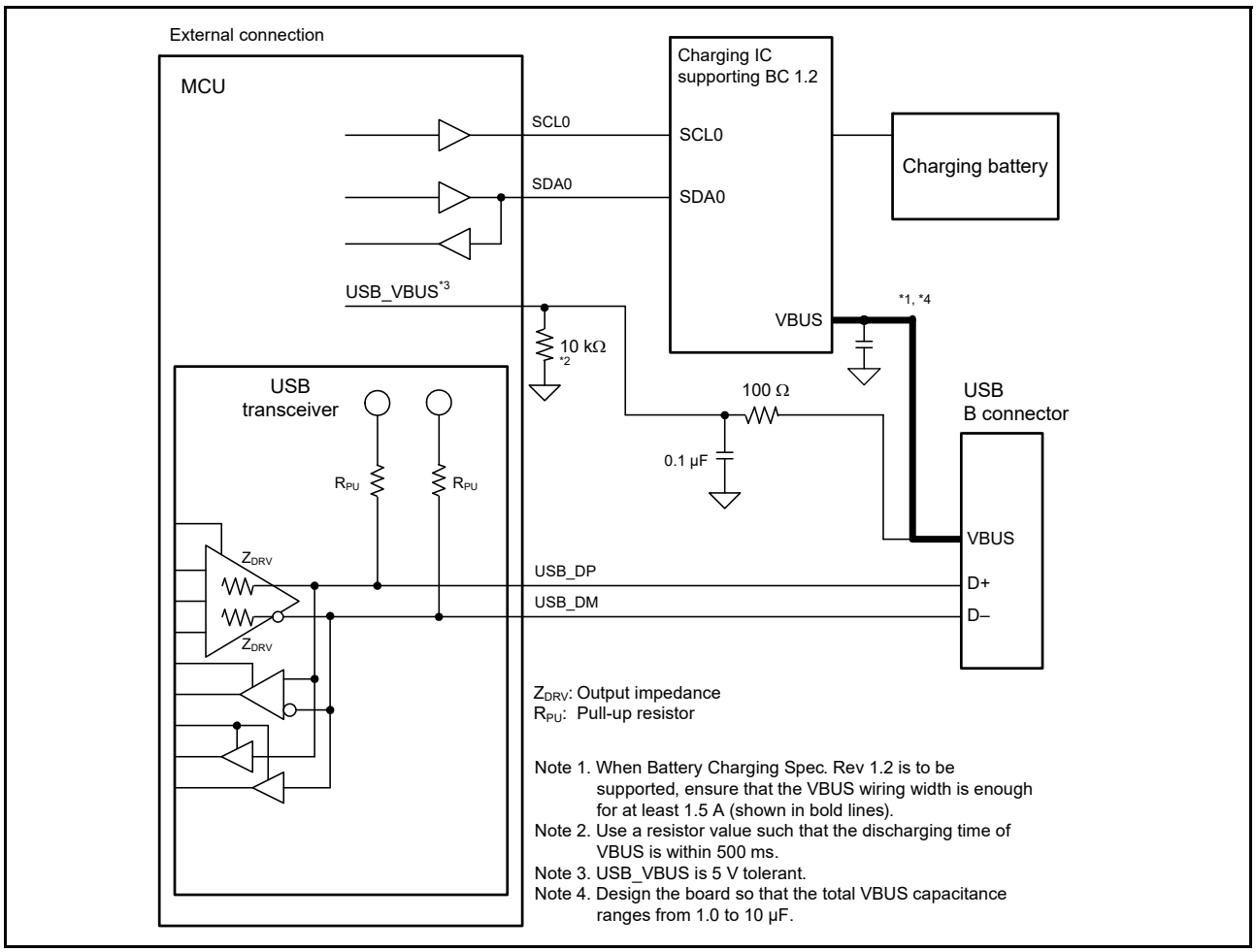


Figure 28.10 Example of functional connection with Battery Charging Rev 1.2 supported

28.3.2 Interrupts

Table 28.13 lists the interrupt sources in the USBFS. When an interrupt generation condition is satisfied and the interrupt output is enabled using the associated interrupt enable register, a USBFS interrupt request is issued to the Interrupt Controller Unit (ICU) and an USBFS interrupt is generated.

Table 28.13 Interrupt sources (1 of 2)

Bit to be set	Name	Interrupt source	Applicable controller function	Status flag
VBINT	VBUS interrupt	• A change in the state of the USB_VBUS input pin was detected (low to high or high to low)	Host or device*1	INTSTS0.VBSTS
RESM	Resume interrupt	• A change in the state of the USB bus was detected in the suspended state (J-state to K-state or J-state to SE0).	Device	-
SOFR	Frame number update interrupt	In host controller mode: • An SOF packet with a different frame number was transmitted. In device controller mode: • An SOF packet with a different frame number was received.	Host/device	-

Table 28.13 Interrupt sources (2 of 2)

Bit to be set	Name	Interrupt source	Applicable controller function	Status flag
DVST	Device state transition interrupt	One of the following device state transitions was detected: • USB bus reset detected • Suspend state detected • SET_ADDRESS request received • SET_CONFIGURATION request received.	Device	INTSTS0.DVSQ[2:0]
CTRT	Control transfer stage transition interrupt	A control transfer stage transition was detected because of one of the following: • Setup stage completed • Control write transfer status stage transition occurred • Control read transfer status stage transition occurred • Control transfer completed • Control transfer sequence error occurred.	Device	INTSTS0.CTSQ[2:0]
BEMP	Buffer empty interrupt	• The buffer is empty after all FIFO buffer data was transmitted • A packet larger than the maximum packet size was received.	Host/device	BEMPSTS.PIPEnBEMP
NRDY	Buffer not ready interrupt	In host controller mode: • A STALL response was received from the peripheral device in response to the issued token • The response from the peripheral device in response to the issued token was not received successfully (no response three times consecutively or packet reception error three times consecutively) • An overrun or underrun error occurred during isochronous transfer In device controller mode: • NAK was returned for an IN or OUT token while the PID[1:0] bits were set to 01b (BUF) • A CRC error or bit stuffing error occurred during data reception in isochronous transfer • An overrun or underrun occurred during data reception in isochronous transfer.	Host/device	NRDYSTS.PIPEnNRDY
BRDY	Buffer ready interrupt	• The buffer is ready (readable or writable state)	Host/device	BRDYSTS.PIPEnBRDY
OVRCR	Overcurrent input change interrupt	• USB_OVRCURA or USB_OVRCURB input pin state change was detected (low to high or high to low)	Host	INTSTS1.OVRCR
BCHG	Bus change interrupt	• USB bus state change was detected	Host/device	SYSSTS0.LNST[1:0]
DTCH	Disconnection detection during full-speed operation	• Peripheral device disconnect was detected in full-speed operation	Host	DVSTCTR0.RHST[2:0]
ATTCH	Device connection detection	• J-state or K-state was detected on the USB bus for 2.5 µs continuously This interrupt can be used to check whether peripheral devices are connected.	Host	-
EOFERR	EOF error detection	• An EOF error was detected for a peripheral device	Host	-
SACK	Normal setup operation	• A setup transaction normal response (ACK) was received	Host	-
SIGN	Setup error	• A setup transaction error (no response or ACK packet corruption) was detected three consecutive times	Host	-
PDDEINT0	Portable device detection interrupt	• A connection of the portable device was detected	Host	INTSTS1.PDDETINT0

Note 1. Although this interrupt can be generated in host controller mode, it is not usually used in this mode.

Figure 28.11 shows the circuits related to the USBFS interrupts.

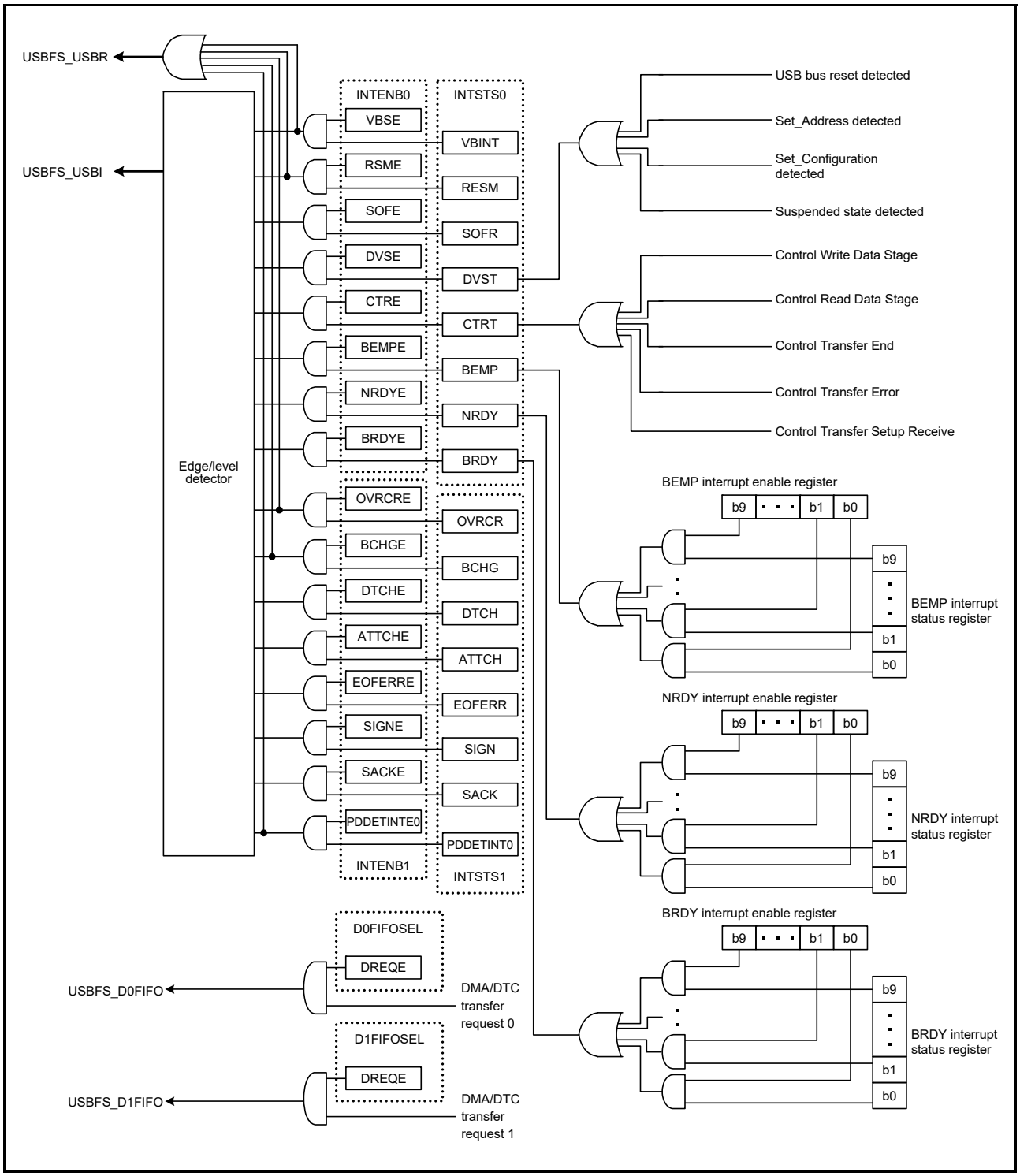


Figure 28.11 USBFS interrupt-related circuits

Table 28.14 shows the interrupts generated by the USBFS.

Table 28.14 USBFS interrupts

Interrupt name	Interrupt status flag	DTC activation	DMAC activation	Priority
D0FIFO	DMA transfer request 0	Possible	Possible	High
D1FIFO	DMA transfer request 1	Possible	Possible	
USBFS_USBI	VBUS interrupt, resume interrupt, frame number update interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, overcurrent input change interrupt, bus change interrupt, disconnection detection during full-speed operation, device connection detection, EOF error detection, normal setup operation, setup error, and portable device detection interrupt	Not possible	Not possible	Low
USBFS_USBR	VBUS interrupt, resume interrupt, overcurrent input change interrupt, bus change interrupt, and portable device detection interrupt	Not possible	Not possible	-

28.3.3 Interrupt Descriptions

28.3.3.1 BRDY Interrupt

The BRDY interrupt is generated in both host and device controller modes. This section describes the conditions in which the USBFS sets the associated bit in BRDYSTS to 1. Under these conditions, the USBFS generates a BRDY interrupt if the software has set 1 to the bit in BRDYENB associated with the given pipe and 1 to the INTENB0.BRDYE bit.

The conditions for generating and clearing the BRDY interrupt depend on the SOFCFG.BRDYM and PIPECFG.BFRE settings for each pipe as follows:

(1) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.

On any of the following conditions, the USBFS generates an internal BRDY interrupt request trigger and sets 1 to the BRDYSTS.PIPEnBRDY bit associated with the selected pipe.

(a) For Transmitting Pipes

- When the DIR bit is changed from 0 to 1 by the software.
- When packet transmission is complete for a pipe while write-access from the CPU to the FIFO buffer for the pipe is disabled (when the BSTS bit is read as 0).
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode.
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is complete.
- When the hardware flushes the buffer of the pipe for isochronous transfers.
- When 1 is written to the PIPEnCTR.ACLRM bit, which causes the FIFO buffer to transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP, that is, during data transmission for control transfers.

(b) For Receiving Pipes

- When packet reception is successfully complete, enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the given pipe is disabled (when the BSTS bit is read as 0). No request trigger is generated for transactions in which a DATA-PID mismatch has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode. No request trigger is generated until completion of reading data from the currently-read FIFO buffer, even if reception by the other FIFO buffer is complete.

In device controller mode, the BRDY interrupt is not generated in the status stage of control transfers. The PIPEnBRDY interrupt status of the selected pipe can be set to 0 by writing 0 to the associated PIPEnBRDY bit through the software. In this case, the other PIPEnBRDY bits should be set to 1.

Clear the BRDY status before accessing the FIFO buffer.

(2) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 1

With these settings, the USBFS generates a BRDY interrupt on completion of reading all data for a single transfer using the receiving pipe, and sets 1 to the bit in BRDYSTS associated with the pipe.

On any of the following conditions, the USBFS determines that the last data for a single transfer was received.

- When a short packet including a zero-length packet is received
- When the PIPEn transaction counter register (PIPEnTRN) is used and the number of packets specified in the PIPEnTRN.TRNCNT[15:0] bits are completely received.

When the data is completely read after any of the above conditions is satisfied, the USBFS determines that all data for a single transfer is completely read.

When a zero-length packet is received while the FIFO buffer is empty, the USBFS determines that all data for a single transfer is completely read when the FRDY bit in the FIFO port control register is 1 and the DTLN[8:0] bits are 0. In this case, to start the next transfer, write 1 to the BCLR bit in the associated port control register through the software. With these settings, the USBFS does not detect a BRDY interrupt for the transmitting pipe.

The PIPEnBRDY interrupt status of a pipe can be set to 0 by writing 0 to the associated BRDYSTS.PIPEnBRDY bit through the software. In this case, the other PIPEnBRDY bits should be set to 1.

In this mode, do not change the PIPECFG.BFRE bit setting until all data for a single transfer is processed. When it is necessary to change the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pipe must be cleared using the PIPEnCTR.ACLRM bit.

(3) When SOFCFG.BRDYM = 1 and PIPECFG.BFRE = 0

With these settings, the BRDYSTS.PIPEnBRDY values are linked to the BSTS bit setting for each pipe. In other words, the BRDY interrupt status bits (PIPEnBRDY) are set to 1 or 0 by the USB depending on the FIFO buffer status.

(a) For Transmitting Pipes

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready. The BRDY interrupt is not generated for the DCP in the transmitting direction even when it is ready for write access.

(b) For Receiving Pipes

The BRDY interrupt status bits set to 1 when the FIFO buffer is ready for read access, and set to 0 when all data has been read (not ready for read access).

When a zero-length packet is received while the FIFO buffer is empty, the associated bit is set to 1 and the BRDY interrupt is continuously generated until the software writes 1 to BCLR. With this setting, the PIPEnBRDY bit cannot be set to 0 by the software.

When the SOFCFG.BRDYM bit is set to 1, set the PIPECFG.BFRE bit for all pipes to 0.

Figure 28.12 shows the timing of BRDY interrupt generation.

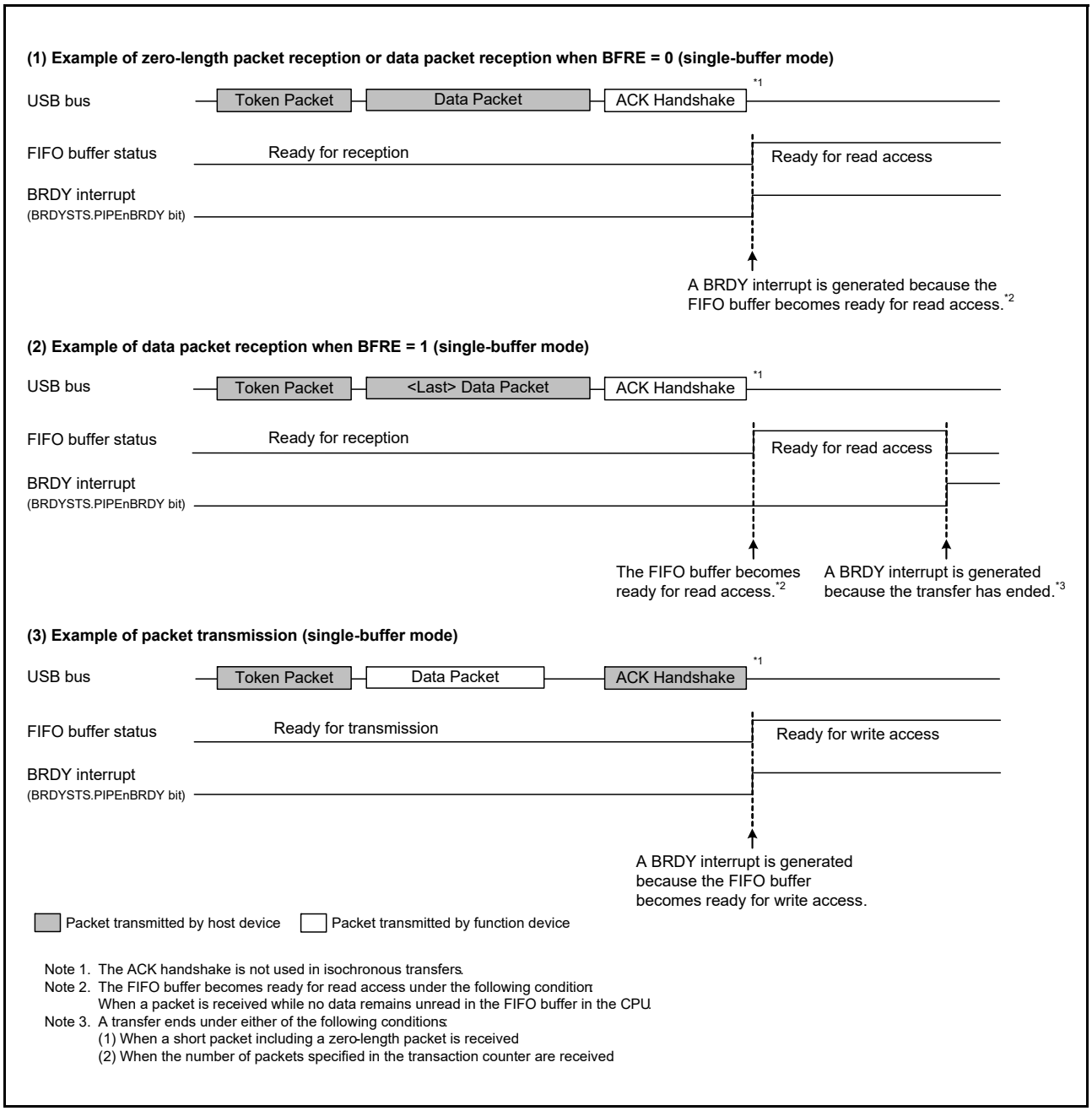


Figure 28.12 Timing of BRDY interrupt generation

The condition for clearing the INTSTS0.BRDY bit depends on the SOFCFG.BRDYM bit setting as shown in Table 28.15.

Table 28.15 Condition for clearing BRDY bit

BRDYM bit	Condition for clearing BRDY bit
0	When all bits in BRDYSTS are set to 0 by software
1	When the BSTS bits for all pipes become 0

28.3.3.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID bits are set to BUF by the software, the USBFS sets the associated PIPEnNRDY bit in NRDYSTS to 1. If the associated bit in NRDYENB is set to 1 by the software, the USBFS sets the INTSTS0.NRDY bit to 1 and generates a USBFS interrupt.

This section describes the conditions in which the USBFS generates the internal NRDY interrupt request for a given pipe.

The internal NRDY interrupt request is not generated during setup transaction execution in host controller mode. During setup transactions in host controller mode, the SACK or SIGN interrupt is detected.

The internal NRDY interrupt request is not generated during status stage execution of the control transfer in device controller mode.

(1) In Host Controller Mode

(a) For Transmitting Pipes

On any of the following conditions, the USBFS detects an NRDY interrupt:

- For isochronous transfer pipes, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer. In this case, the USBFS transmits a zero-length packet following the OUT token and sets the associated NRDYSTS.PIPEnNRDY bit and the FRMNUM.OVRN bit to 1.
- During communications other than setup transactions on pipes not used for isochronous transfers, when any combination of the following two cases occur three consecutive times:
  - No response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device.
  - An error is detected in the packet from the peripheral device. In this case, the USBFS sets the associated PIPEnNRDY bit to 1 and changes the associated PID[1:0] setting for the pipe to NAK.
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device. In this case, the USBFS sets the associated PIPEnNRDY bit to 1 and changes the PID[1:0] setting for the associated pipe to STALL (11b).

(b) For Receiving Pipes

- For isochronous transfer pipes, when the time to issue an IN token comes but there is no space available in the FIFO buffer. In this case, the USBFS discards the received data for the IN token and sets the PIPEnNRDY bit associated with the pipe and the OVRN bit to 1. When a packet error is detected in the received data for the IN token, the USBFS also sets the FRMNUM.CRCE bit to 1.
- For non-isochronous transfer pipes, when any combination of the following two cases occur three consecutive times:
  - No response is returned from the peripheral device for the IN token issued by the USBFS (when timeout is detected before detection of the DATA packet from the peripheral device).
  - An error is detected in the packet from the peripheral device. In this case, the USBFS sets the associated PIPEnNRDY bit to 1 and changes the associated PID[1:0] setting for the pipe to NAK.
- For isochronous transfer pipes, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device. In this case, the USBFS sets the PIPEnNRDY bit associated with the pipe to 1. The PID[1:0] setting for the pipe is not changed.
- For isochronous transfer pipes, when a CRC error or a bit stuffing error is detected in the received data packet. In this case, the USBFS sets the PIPEnNRDY bit associated with the pipe and the CRCE bit to 1.
- When the STALL handshake is received. In this case, the USBFS sets the PIPEnNRDY bit associated with the pipe to 1 and changes the PID[1:0] setting for the associated pipe to STALL.

(2) In Device Controller Mode

(a) For Transmitting Pipes

- When an IN token is received while there is no data to be transmitted in the FIFO buffer. In this case, the USBFS generates a NRDY interrupt request on reception of the IN token and sets the NRDYSTS.PIPEnNRDY bit to 1. For an isochronous transfer pipe in which an interrupt is generated, the USBFS transmits a zero-length packet and sets the FRMNUM.OVRN bit to 1.

(b) For Receiving Pipes

- When an OUT token is received but there is no space available in the FIFO buffer. For an isochronous transfer pipe in which an interrupt is generated, the USBFS generates a NRDY interrupt request on reception of the OUT token and sets the PIPEnNRDY bit to 1 and OVRN bit to 1. For a non-isochronous transfer pipe in which an interrupt is generated, the USBFS generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPEnNRDY bit to 1. The NRDY interrupt request is not generated during retransmission because of a DATA-PID mismatch. In addition, the NRDY interrupt request is not generated if an error occurs in the DATA packet.
- For isochronous transfer pipes, when a token is not received successfully within an interval frame. In this case, the USBFS generates a NRDY interrupt request when the SOF is received, and sets the PIPEnNRDY bit to 1.

Figure 28.13 shows the timing of NRDY interrupt generation when the device controller is selected.

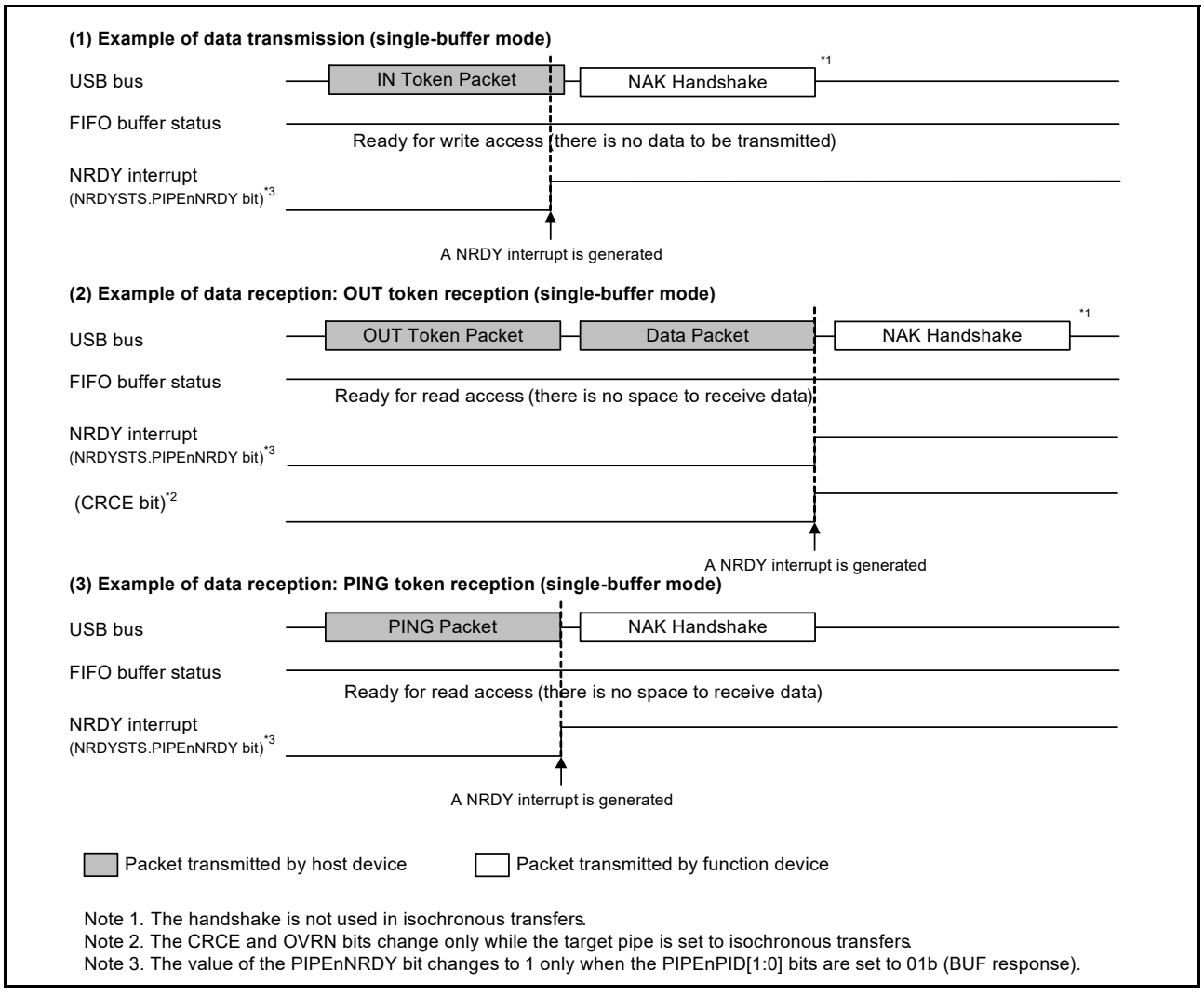


Figure 28.13 Timing of NRDY interrupt generation in device controller mode

28.3.3.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID bits are set to BUF by the software, the USBFS sets the associated BEMPSTS.PIPEnBEMP bit to 1. If the associated bit in BEMPENB is set to 1 by the software, the USBFS sets the INTSTS0.BEMP bit to 1 and generates a USBFS interrupt. This section describes the conditions in which the USBFS generates an internal BEMP interrupt request.

(1) For Transmitting Pipes

When the FIFO buffer of the associated pipe is empty on completion of transmission, including zero-length packet transmission, and in single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for a non-DCP pipe. The internal BEMP interrupt request is not generated on any of the following conditions:

- When the CPU or DMA/DTC has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode.
- When the buffer is cleared (emptied) by a 1 setting to the PIPEnCTR.ACLRM or the BCLR bit in the port control register.
- When an IN transfer (zero-length packet transmission) is performed during the control transfer status stage in device controller mode.

(2) For Receiving Pipes

When a successfully-received data packet size exceeds the specified maximum packet size. In this case, the USBFS generates a BEMP interrupt request, sets the associated BEMPSTS.PIPEnBEMP bit to 1, discards the received data, and changes the associated PID[1:0] setting for the pipe to STALL (11b). The USBFS returns no response in host controller mode, and returns STALL response in device controller mode.

The internal BEMP interrupt request is not generated on any of the following conditions:

- When a CRC error or a bit stuffing error is detected in the received data
- When a setup transaction is performed:
  - Writing 0 to the BEMPSTS.PIPEnBEMP bit clears the status
  - Writing 1 to the BEMPSTS.PIPEnBEMP bit has no effect.

Figure 28.14 shows the timing of BEMP interrupt generation in device controller mode.

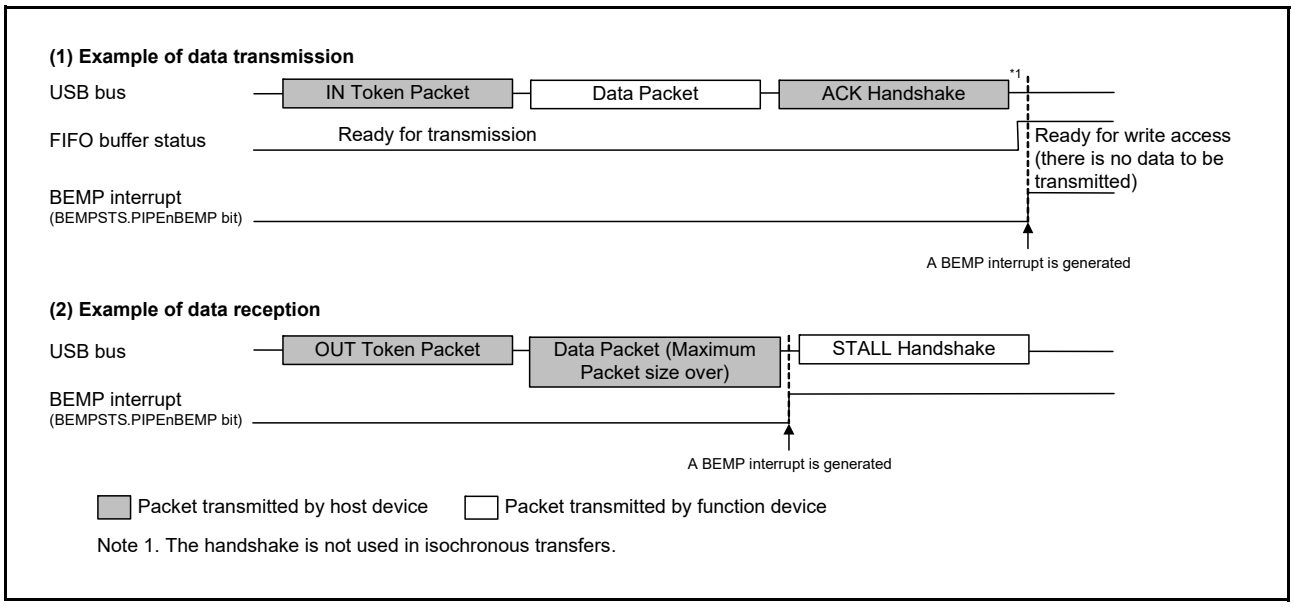


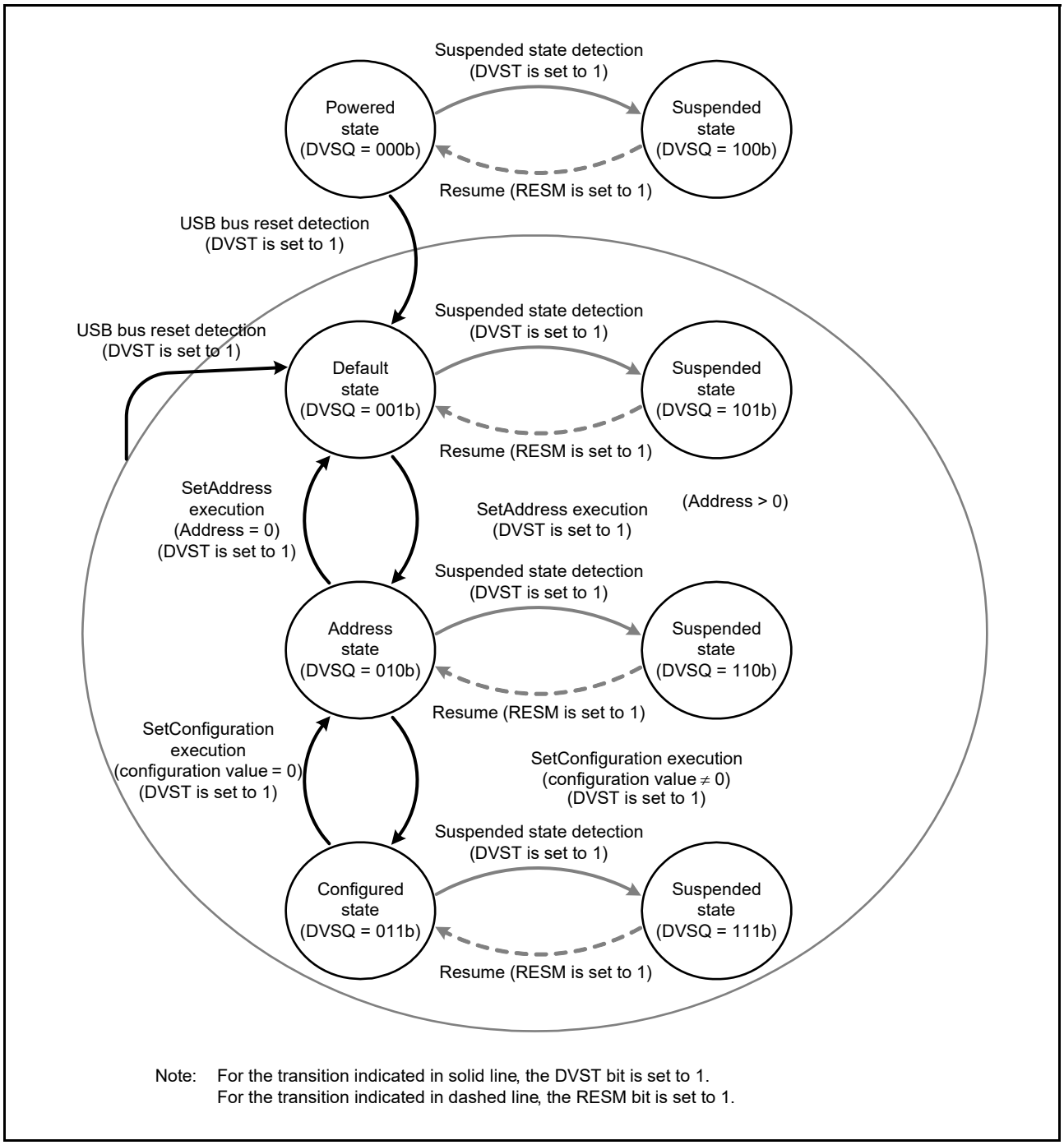
Figure 28.14 Timing of BEMP interrupt generation in device controller mode

28.3.3.4 Device State Transition Interrupt (Device Controller Mode)

Figure 28.15 shows a diagram of the USBFS device state transitions. The USBFS controls device states and generates device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by means of the resume interrupt. Device state transition interrupts can be enabled or disabled independently in INTENB0. Devices whose states have changed can be checked in the INTSTS0.DVSQ[2:0] bits.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

The USBFS controls device states, and device state transition interrupts can be generated, only in device controller mode.



Note: For the transition indicated in solid line, the DVST bit is set to 1.
For the transition indicated in dashed line, the RESM bit is set to 1.

Figure 28.15 Device state transitions

28.3.3.5 Control Transfer Stage Transition Interrupt (Device Controller Mode)

Figure 28.16 shows a diagram of the control transfer stage transitions of the USBFS. The USBFS controls the control transfer sequence and generates control transfer stage transition interrupts. Control transfer stage transition interrupts can be enabled or disabled independently in INTENB0. Transfer stages that have transitioned can be checked in the INTSTS0.CTSQ[2:0] bits.

Control transfer stage transition interrupts are generated only in device controller mode. This section describes control transfer sequence errors. If an error occurs, the DCPCTR.PID[1:0] bits are set to 1xb (STALL response).

(1) Control Read Transfer Errors

- An OUT token is received but no data is transferred in response to the IN token at the data stage
- An IN token is received at the status stage
- A data packet with DATAPID = DATA0 is received at the status stage.

(2) Control Write Transfer Errors

- An IN token is received but no ACK is returned in response to the OUT token at the data stage
- A data packet with DATAPID = DATA0 is received as the first data packet at the data stage
- An OUT token is received at the status stage.

(3) Control Write No Data Transfer Errors

- An OUT token is received at the status stage.

At the control write transfer data stage, if the receive data length exceeds the wLength value of the USB request, it is not recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (INTSTS0.CTRT = 1), the CTSQ[2:0] = 110b value is saved until the CTRT bit is set to 0, clearing the interrupt status. While CTSQ[2:0] = 110b is being saved, no CTRT interrupt for ending the setup stage is generated, even if a new USB request is received. The USBFS saves the setup stage completion status, and it generates a CTRT interrupt after the interrupt status is cleared by the software.

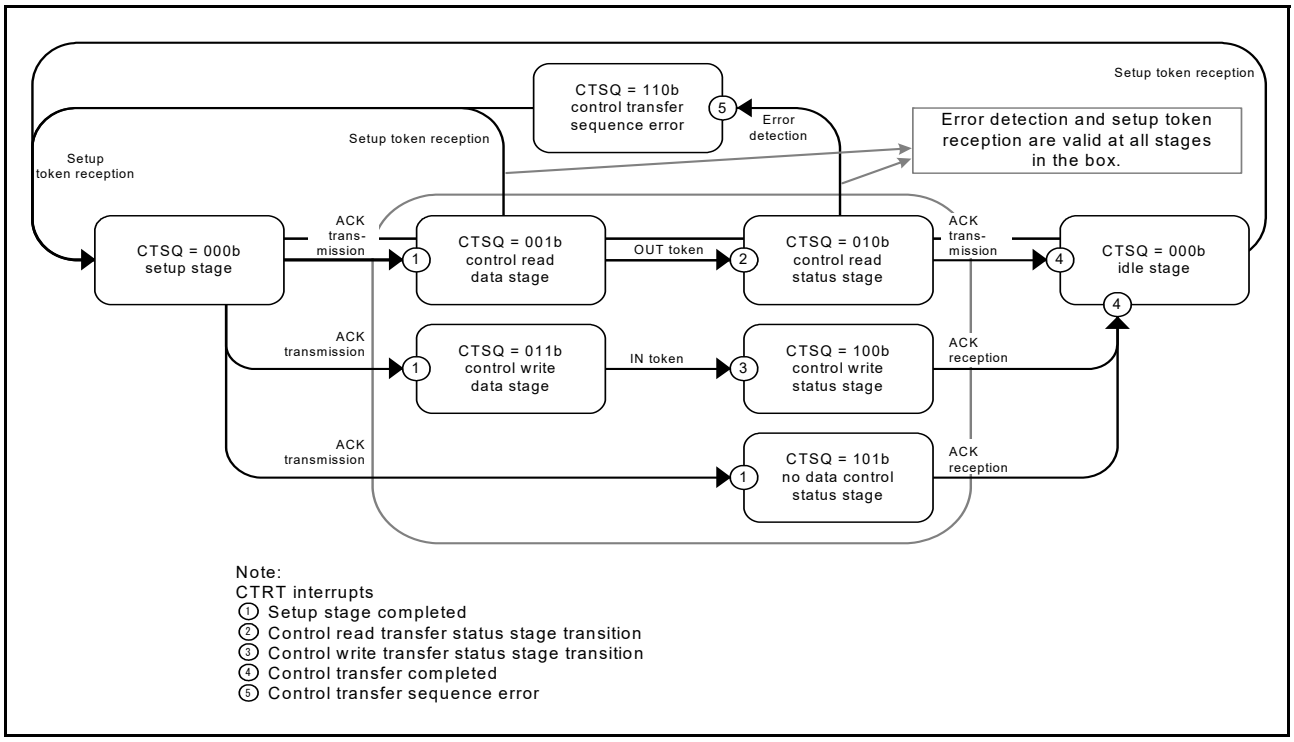


Figure 28.16 Control transfer stage transitions

28.3.3.6 Frame Update Interrupt

In host controller mode, an interrupt is generated when the frame number is updated.

In device controller mode, an SOFR interrupt is generated when the frame number is updated. The USBFS updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

28.3.3.7 VBUS Interrupt

When the USB_VBUS pin level changes, a VBUS interrupt is generated. The level of the USB_VBUS pin can be checked with the INTSTS0.VBSTS bit. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. If the system is activated with the host controller connected, the first VBUS interrupt is not generated, because there is no change in the USB_VBUS pin level.

28.3.3.8 Resume Interrupt

In device controller mode, a resume interrupt is generated when the device state is the suspended state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the suspended state is detected by means of the resume interrupt.

In host controller mode, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

28.3.3.9 OVRCR Interrupt

An OVRCR interrupt is generated when the USB_OVRCURA or USB_OVRCURB pin level has changed. The levels of the USB_OVRCURA and USB_OVRCURB pins can be checked in the SYSSTS0.OVCMON[1:0] bits. The external power supply IC can check whether overcurrent is detected using the OVRCR interrupt.

For OTG connections, the OVRCR interrupt allows you to check whether a change is detected in the VBUS comparator.

28.3.3.10 BCHG Interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether a peripheral device is connected and can also be used to detect a remote wakeup in host controller mode. The BCHG interrupt is generated both host and device controller modes.

28.3.3.11 DTCH Interrupt

A DTCH interrupt is generated when a USB bus disconnect is detected in host controller mode. The USBFS detects bus disconnects in compliance with the USB 2.0 specification.

On interrupt detection, all pipes in which communications are carried out for the relevant port must be terminated by the software. The pipes enter the wait state for a bus connection to the port, waiting for an ATTCH interrupt to generate. Regardless of the value set in the associated interrupt enable bit, the USBFS hardware:

- Sets the DVSTCTR0.UACT bit for the port in which the DTCH interrupt is detected to 0
- Puts the port in which the DTCH interrupt occurred into the idle state.

28.3.3.12 SACK Interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet is received from the peripheral device in host controller mode. The SACK interrupt can be used to confirm that the setup transaction is successfully complete.

28.3.3.13 SIGN Interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet has not been correctly received from the peripheral device three consecutive times in host controller mode. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

28.3.3.14 ATTCH Interrupt

An ATTCH interrupt is generated when J-state or K-state of the full-speed signal level is detected on the USB port for 2.5 µs with in host controller mode. To be more specific, an ATTCH interrupt is detected on any of the following conditions:

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 µs
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 µs.

28.3.3.15 EOFERR Interrupt

An EOFERR interrupt occurs when the USBFS detects that communication is not complete at the EOF2 timing defined in the USB 2.0 specification.

On interrupt detection, all pipes in which communications are being carried out for the relevant port must be terminated by the software, and the port must be re-enumerated. Regardless of the value set in the associated interrupt enable bit, the USBFS hardware:

- Sets the DVSTCTR0.UACT bit for the port in which the EOFERR interrupt is detected to 0
- Puts the port in which the EOFERR interrupt is generated into the idle state.

28.3.3.16 Portable Device Detection Interrupt

A portable device detection interrupt is generated when the USBFS detects a level change (high to low or low to high) in the PDDET output from the USB-PHY. When a portable device detection interrupt is generated, use software to repeat the reading of the PDDETSTS0 bit until the same value is read three or more times to debounce the signal.

28.3.4 Pipe Control

Table 28.16 lists the pipe settings for the USBFS. USB data transfer is performed through logical pipes that the software associates with endpoints. The USBFS provides 10 pipes that are used for data transfer. Set up the pipes based on your system specifications.

Table 28.16 Pipe settings (1 of 2)

Register name	Bit name	Setting	Remarks
DCPCFG PIPECFG	TYPE	Transfer type	Pipes 1 to 9: Settable
	BFRE	BRDY interrupt mode	Pipes 1 to 5: Settable
	DBLB	Double buffer select	Pipes 1 to 5: Settable
	DIR	Transfer direction select	IN or OUT settable
	EPNUM	Endpoint number	Pipes 1 to 9: Settable A value other than 0000b must be set when the pipe is used.
	SHTNAK	Disabled state select for pipe on transfer end	Pipes 1 and 2: Settable only for bulk transfers Pipes 3 to 5: Settable
DCPMAXP PIPEMAXP	DEVSEL	Device select	Referenced only in host controller mode
	MXPS	Maximum packet size	Compliant with the USB 2.0 specification
PIPEPERI	IFIS	Buffer flush	Pipes 1 and 2: Settable only for isochronous transfers Pipes 3 to 9: Setting disabled
	IITV	Interval counter	Pipes 1 and 2: Settable only for isochronous transfers Pipes 3 to 5: Setting disabled Pipes 6 to 9: Settable only in host controller mode

Table 28.16 Pipe settings (2 of 2)

Register name	Bit name	Setting	Remarks
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit
	INBUFM	IN buffer monitor	Available only for pipes 1 to 5
	SUREQ	Setup request	Settable only for the DCP and controlled in host controller mode
	SUREQCLR	SUREQ clear	Settable only for the DCP and controlled in host controller mode
	ATREPM	Auto response mode	Pipes 1 to 5: Settable only in device controller mode
	ACLRM	Auto buffer clear	Pipes 1 to 9: Settable
	SQCLR	Sequence clear	Clears the data toggle bit
	SQSET	Sequence set	Sets the data toggle bit
	SQMON	Sequence monitor	Monitors the data toggle bit
	PBUSY	Pipe busy status	-
	PID	Response PID	See section 28.3.4.6, Response PID
PIPEnTRE	TRENB	Transaction counter enable	Pipes 1 to 5: Settable
	TRCLR	Current transaction counter clear	Pipes 1 to 5: Settable
PIPEnTRN	TRNCNT	Transaction counter	Pipes 1 to 5: Settable

28.3.4.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be changed only when USB communication is prohibited (PID = NAK).

Do not change the following registers and bits when USB communication is enabled (PID = BUF):

- Bits in DCPCFG and DCPMAXP
- SQCLR and SQSET bits in DCPCTR
- Bits in PIPECFG, PIPEMAXP, and PIPEPERI
- ATREPM, ACLRM, SQCLR, and SQSET bits in PIPEnCTR
- Bits in PIPEnTRE and PIPEnTRN.

To set these bits when USB communication is enabled (PID = BUF):

1. A request to change the bits in the pipe control register occurs.
2. Set the PID[1:0] bits associated with the pipe to NAK.
3. Wait until the associated PBUSY bit is set to 0.
4. Set the bits in the pipe control register.

The following bits in the pipe control registers can be changed only when the selected pipe information has not been set in the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Do not set the following registers when the CURPIPE[3:0] bits are set:

- Bits in DCPCFG and DCPMAXP
- Bits in PIPECFG, PIPEMAXP and PIPEPERI.

To change pipe information, you must set the CURPIPE[3:0] bits in the port select registers to a pipe other than the one to be changed. For the DCP, the buffer must be cleared using the BCLR bit in the Port Control Register after the pipe information is changed.

28.3.4.2 Transfer Types

The PIPECFG.TYPE[1:0] bits specify the following transfer types for each pipe:

- DCP: No setting is necessary (fixed at control transfer)
- Pipes 1 and 2: Set to bulk or isochronous transfer
- Pipes 3 to 5: Set to bulk transfer
- Pipes 6 to 9: Set to interrupt transfer.

28.3.4.3 Endpoint Number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to 15.

- DCP: No setting is necessary (fixed at endpoint 0)
- Pipes 1 to 9: Select and set the endpoint numbers from 1 to 15 so that the combination of the PIPECFG.DIR and EPNUM[3:0] bits is unique.

28.3.4.4 Maximum Packet Size Setting

Specify the maximum packet size for each pipe in the DCPMAXP.MXPS[6:0] and PIPEMAXP.MXPS[8:0] bits. The DCP and pipes 1 to 5 can be set to any of the maximum pipe sizes defined in the USB 2.0 specification. For pipes 6 to 9, the maximum packet size is 64 bytes. Set the maximum packet size as follows before starting a transfer (PID = BUF):

- DCP: Set to 8, 16, 32, or 64
- Pipes 1 to 5: Set to 8, 16, 32, or 64 for bulk transfers
- Pipes 1 and 2: Set between 1 and 256 for isochronous transfers
- Pipes 6 to 9: Set between 1 and 64.

28.3.4.5 Transaction Counter for Pipes 1 to 5 in the Receiving Direction

When the specified number of transactions is complete in the data packet receiving direction, the USBFS recognizes that the transfer has ended. Two transaction counters are provided: one is the PIPEnTRN register that specifies the number of transactions to be executed and the other is the current counter that internally counts the number of executed transactions. If the PIPECFG.SHTNAK bit is set to 1, when the current counter value matches the specified number of transactions, the associated PIPEnCTR.PID[1:0] bits are set to NAK and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The data read from PIPEnTRN differs depending on the PIPEnTRE.TRENB setting as follows:

- The TRENB bit = 0: Specified transaction counter value can be read.
- The TRENB bit = 1: Current counter value indicating the internally counted number of executed transactions can be read.

The following constraints apply when working with the TRCLR bit:

- If the transactions are being counted and PID = BUF, the current counter cannot be cleared
- If there is any data left in the buffer, the current counter cannot be cleared.

28.3.4.6 Response PID

Specify the response PID for each pipe in the PID[1:0] bits in DCPCTR and PIPEnCTR. This section describes the USBFS operation with different response PID settings.

(1) Software Response PID Settings in Host Controller Mode

Select the response PID to specify the execution of transactions as follows:

- NAK setting: Using pipes is disabled and no transactions are executed
- BUF setting: Transactions are executed based on the FIFO buffer state:

OUT direction: An OUT token is issued if the FIFO buffer contains transmit data.
IN direction: An IN token is issued if the FIFO buffer is not full and can receive data.

- STALL setting: Using pipes is disabled and no transactions are executed.

Note: Use the DCPCTR.SUREQ bit to execute setup transactions for the DCP.

(2) Software Response PID Settings in Device Controller Mode

Select the response PID to respond as follows to transactions from the host:

- NAK setting: A NAK response is returned to all generated transactions
- BUF setting: A response is returned to transactions based on the FIFO buffer
- STALL setting: A STALL response is returned to all generated transactions.

Note: For setup transactions, an ACK response is always returned, regardless of the PID[1:0] bits setting, and the USB request is stored in the register.

Sections (3) and (4) describe situations in which the USBFS writes to the PID[1:0] bits because of specific transaction results.

(3) Hardware Response PID Settings in Host Controller Mode

- NAK setting: PID = NAK is set in the following cases, and issuing of tokens is automatically stopped:
  - When a non-isochronous transfer is performed and an NRDY interrupt is generated (For details, see section 28.3.3.2, NRDY Interrupt.)
  - If a short packet is received when the PIPECFG.SHTNAK bit is set to 1 for bulk transfers
  - If transaction counting ends when the SHTNAK bit is set to 1 for bulk transfers.
- BUF setting: The USBFS does not write this setting.
- STALL setting: PID = STALL is set in the following cases, and issuing of tokens is automatically stopped:
  - When STALL is received in response to a transmitted token
  - When a received data packet exceeds the maximum packet size.

(4) Hardware Response PID Settings in Device Controller Mode

- NAK setting: PID = NAK is set in the following cases, and a NAK response is returned to transactions:
  - When the setup token is received normally (DCP only)
  - If transaction counting ends or a short packet is received when the PIPECFG.SHTNAK bit is set to 1 for bulk transfers.
- BUF setting: There is no BUF writing by the USBFS.
- STALL setting: PID = STALL is set in the following cases, and a STALL response is returned to transactions:
  - When a received data packet exceeds the maximum packet size
  - When a control transfer sequence error is detected (DCP only).

28.3.4.7 Data PID Sequence Bit

The USBFS automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON bit in DCPCTR and PIPEnCTR. When data is transmitted, the sequence bit toggles on ACK handshake reception. When data is received, the sequence bit toggles on ACK handshake transmission. The SQCLR and the SQSET bits in DCPCTR and PIPEnCTR registers can be used to change the data PID sequence bit.

In device controller mode when control transfers are used, the USBFS automatically sets the sequence bit for stage transitions. DATA1 is returned when the setup stage ends. The sequence bit is not referenced and PID = DATA1 is returned in the status stage. Therefore, no software settings are required. However, in host controller mode when control transfers are used, the sequence bit must be set by the software for the stage transitions.

For ClearFeature requests for transmission or reception, the data PID sequence bit must be set by the software in both host and device controller modes.

28.3.4.8 Response PID = NAK Function

The USBFS provides a function for disabling pipe operation (PID response = NAK) when the final data packet of a transaction is received. The USBFS automatically distinguishes this based on reception of a short packet or the transaction counter. Enable this function by setting the PIPECFG.SHTNAK bit to 1.

When the double buffer mode is being used for the FIFO buffer, using this function enables reception of data packets in transfer units. If pipe operation is disabled, the software must enable the pipe again (PID response = BUF).

The response PID = NAK function can be used only for bulk transfers.

28.3.4.9 Auto Response Mode

For bulk transfer pipes (1 to 5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (PIPECFG.DIR = 0), OUT-NAK mode is invoked, and during an IN transfer (DIR = 1), null auto response mode is invoked.

28.3.4.10 OUT-NAK Mode

For bulk OUT transfer pipes, NAK is returned in response to an OUT token, and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To transition from normal mode to OUT-NAK mode, specify OUT-NAK mode while pipe operation is disabled (PID[1:0] = 00b for NAK response). Next enable pipe operation (PID[1:0] = 01b for BUF response), on which OUT-NAK mode becomes valid. If an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host.

To transition from OUT-NAK mode to normal mode, cancel OUT-NAK mode while pipe operation is disabled (NAK). Next enable pipe operation (BUF). In normal mode, reception of OUT data is enabled.

28.3.4.11 Null Auto Response Mode

For bulk IN transfer pipes, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1.

To transition from normal mode to null auto response mode, specify null auto response mode in while pipe operation is disabled (response PID = NAK). Next enable pipe operation (response PID = BUF), on which null auto response mode becomes valid. Before setting null auto response mode, check that PIPEnCTR.INBUFM = 0, because the mode can be set only when the buffer is empty. If the INBUFM bit is 1, empty the buffer using the PIPEnCTR.ACLRM bit. Do not write data from the FIFO port while a transition to null auto response mode is being made.

To transition from null auto response mode to normal mode, keep pipe operation disabled (response PID = NAK) for the period of the zero-length packet transmission (about 10 µs) before canceling the null auto response mode. In normal mode, data can be written from the FIFO port, so packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

28.3.5 FIFO Buffer Memory

The USBFS provides a FIFO buffer for data transfers, and it manages the memory area used for each pipe. The FIFO buffer has two states depending on whether the access right is assigned to the system (CPU side) or the USBFS (SIE side).

(1) Buffer Status

Table 28.17 and Table 28.18 show the buffer status in the USBFS. The FIFO buffer status can be confirmed using the DCPCTR.BSTS and PIPEnCTR.INBUFM bits. The transfer direction for the FIFO buffer can be specified in either the PIPECFG.DIR or CFIFOSEL.ISEL bit (when DCP is selected).

The INBUFM bit is valid for pipes 0 to 5 in the transmitting direction.

When a transmitting pipe uses double buffering, the software can read the BSTS bit to monitor the FIFO buffer status on the CPU side and the INBUFM bit to monitor the FIFO buffer status on the SIE side. When write access to the FIFO port by the CPU or DMA/DTC is slow and the buffer empty status cannot be determined using the BEMP interrupt, the software can use the INBUFM bit to confirm the end of transmission.

Table 28.17 Buffer status indicated by BSTS bit

ISEL or DIR	BSTS	Buffer memory status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet has been received. Reading from the FIFO port is allowed. When a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	The transmission is not complete Writing to the FIFO port is disabled.
1 (transmitting direction)	1	The transmission is complete CPU write is allowed.

Table 28.18 Buffer status indicated by INBUFM bit

DIR	INBUFM	Buffer memory status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	The transmission is complete There is no waiting data to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

28.3.6 FIFO Buffer Clearing

Table 28.19 shows the methods for clearing the FIFO buffer. The FIFO buffer can be cleared using BCLR in the port control register, DnFIFOSEL.DCLRM, or the PIPEnCTR.ACLRM bit.

Single or double buffering can be selected for pipes 1 to 5 in the PIPECFG.DBLB bit.

Table 28.19 List of buffer clearing methods

FIFO buffer clearing mode	Clearing FIFO buffer on the CPU side	Mode for automatically clearing the FIFO buffer after reading the specified pipe data	Auto buffer clear mode for discarding all received packets
Register used	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid.	1: Mode valid 0: Mode invalid.

(1) Auto Buffer Clear Mode Function

The USBFS discards all received data packets if the PIPEnCTR.ACLRM bit is set to 1. If a correct data packet is received, the ACK response is returned to the host controller. The auto buffer clear mode function can only be set in the FIFO buffer reading direction.

Setting the ACLRM bit set to 1 and then to 0 clears the FIFO buffer of the selected pipe no matter what the access direction. An access cycle of at least 100 ns is required for the internal hardware sequence processing between ACLRM = 1 and ACLRM = 0.

28.3.7 FIFO Port Functions

Table 28.20 shows the settings for the FIFO port functions. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, set the BVAL flag in the port control register to end writing. To send a zero-length packet, use the BCLR bit to clear the buffer, and then set the BVAL flag to end writing.

In reading, reception of new packets is automatically enabled when all data is read. Data cannot be read when a zero-length packet is received (DTLN[8:0] = 0), so the buffer must be cleared with the BCLR bit. The length of the receive data can be confirmed in the DTLN[8:0] bits in the port control register.

Table 28.20 FIFO port function settings

Register name	Bit name	Description
CFIFOSEL, DnFIFOSEL (n = 0, 1)	RCNT	Selects DTLN[8:0] read mode
	REW	FIFO buffer rewind (re-read, rewrite)
	DCLRM	Automatically clears receive data for a specified pipe after the data is read (only for DnFIFO)
	DREQE	Enables DMA/DTC transfers (only for DnFIFO)
	MBW	FIFO port access bit width
	BIGEND	Selects FIFO port endian
	ISEL	FIFO port access direction (only for DCP)
	CURPIPE	Selects the current pipe
CFIFOCTR, DnFIFOCTR (n = 0, 1)	BVAL	Ends writing to the FIFO memory
	BCLR	Clears the FIFO buffer on the CPU side
	DTLN	Checks the length of receive data

(1) FIFO Port Selection

Table 28.21 shows the pipes that can be selected with the different FIFO ports. The pipe to be accessed must be selected in the CURPIPE[3:0] bits in the port select register. After the pipe is selected, the software must check whether the written value can be read correctly from the CURPIPE[3:0] bits. (If the previous pipe number is read, it indicates that the USBFS is modifying the pipe.) Next, the software checks that the FRDY bit in the port control register is 1.

In addition, the software must specify the bus width to be accessed in the MBW bit in the port select register. The FIFO buffer access direction conforms to the PIPECFG.DIR setting. For the DCP only, the ISEL bit in the port select register determines the direction.

Table 28.21 FIFO port access by pipe

Pipe	Access method	Port that can be used
DCP	CPU access	CFIFO port register
PIPE1 to PIPE9	CPU access	CFIFO port register D0FIFO/D1FIFO port register
	DMA/DTC access	D0FIFO/D1FIFO port register

(2) REW Bit

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing for the current pipe again. The REW bit in the port select register is used for this processing.

If a pipe is selected in the CURPIPE[3:0] bits in the port select register with the REW bit set to 1, the pointer used for reading from and writing to the FIFO buffer is reset, and reading or writing can be carried out from the first byte. If a pipe is selected with 0 set for the REW bit, data can be read and written in continuation from the previous selection, without the pointer being reset.

To access the FIFO port, the software must check that the FRDY bit in the port control register is 1 after selecting a pipe.

28.3.8 DMA Transfers (D0FIFO and D1FIFO Ports)

(1) Overview of DMA Transfers

For pipes 1 to 9, the FIFO port can be accessed using the DMAC. When buffer access for a pipe targeted for DMA transfer is enabled, a DMA transfer request is issued.

Select the unit of transfer to the FIFO port in the DnFIFOSEL.MBW bit, and select the pipe targeted for the DMA transfer in the DnFIFOSEL.CURPIPE[3:0] bits. Do not change the selected pipe during the DMA transfer.

(2) DnFIFO Auto Clear Mode (D0FIFO and D1FIFO Port Reading Direction)

If 1 is set in the DnFIFOSEL.DCLRM bit, the USBFS automatically clears the FIFO buffer of the selected pipe when reading of data from the FIFO buffer is complete.

Table 28.22 shows the packet reception and FIFO buffer clearing processing by the software for each of the settings. As shown in the table, the buffer clearing conditions depend on the value set in the PIPECFG.BFRE bit. Using the DnFIFOSEL.DCLRM bit eliminates the need for the buffer to be cleared by the software in any situation that requires buffer clearing. This enables DMA transfers without involving software.

The DnFIFO auto clear mode can only be set in the FIFO buffer reading direction.

Table 28.22 Packet reception and FIFO buffer clearing processing by software

Buffer status when packet is received	Register setting			
	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	No clearing required	No clearing required	No clearing required	No clearing required
Zero-length packet reception	Clearing required	Clearing required	No clearing required	No clearing required
Normal short packet reception	No clearing required	Clearing required	No clearing required	No clearing required
Transaction count end	No clearing required	Clearing is necessary	No clearing required	No clearing required

28.3.9 Control Transfers Using DCP

The default control pipe (DCP) is used for data transfers in the control transfer data stage. The FIFO buffer of the DCP is a 64-byte single buffer with a fixed area for both control reads and control writes. The FIFO buffer can be accessed only through the CFIFO port.

28.3.9.1 Control Transfers in Host Controller Mode

(1) Setup Stage

The USQREQ, USBVAL, USBINDX, and USBLENG registers are used to transmit USB requests for setup transactions. Writing the setup packet data to the registers and then writing 1 to the DCPCTR.SUREQ bit transmits the specified data for the setup transaction. On completion of the transaction, the SUREQ bit is set to 0. Do not change these USB request registers while SUREQ = 1.

When an attached function device is detected, the software must issue the first setup transaction for the device using this sequence with the DCPMAXP.DEVSEL[3:0] bits set to 0 and the DEVADD0.USBSPD[1:0] bits set appropriately.

When an attached function device is shifted to the Address state, the software must issues setup transactions using this sequence with the assigned USB address set in the DEVSEL[3:0] bits and the bits in DEVADDn corresponding to the specified USB address set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0010b, make appropriate settings in DEVADD2. When PIPEMAXP.DEVSEL[3:0] = 0101b, make appropriate settings in DEVADD5.

When the setup transaction data is sent, an interrupt request is generated based on the response from the peripheral device (SIGN or SACK bit in INTSTS1). This interrupt request allows the software to check the setup transaction result.

A DATA0 data packet (USB request) for the setup transaction is always transmitted regardless of the status of the DCPCTR.SQMON bit.

(2) Data Stage

The data stage is used to transfer data using the DCP FIFO buffer. Before accessing the DCP FIFO buffer, specify the access direction in the CFIFOSEL.ISEL bit. Specify the transfer direction in the DCPCFG.DIR bit.
For the first data packet of the data stage, the data PID must be transferred as DATA1. Set data PID = DATA1 in the DCPCTR.SQSET bit and set the PID bits = BUF. Completion of data transfer is detected using the BRDY or BEMP interrupt. For control write transfers, when the number of data bytes to be sent is an integer multiple of the maximum packet size, the software must send a zero-length packet at the end.

(3) Status Stage

The status stage is used for zero-length packet data transfers in the reverse direction of the data stage. As in the data stage, data is transfered using the DCP FIFO buffer. Transactions are executed using the same procedure as the data stage.

Data packets in the status stage must be transmitted and received with the data PID set to DATA1 using the DCPCTR.SQSET bit.

When a zero-length packet is received, check the receive-data length in the CFIFOCTR.DTLN[8:0] flags after a BRDY interrupt is generated, and then clear the FIFO buffer using the BCLR bit.

28.3.9.2 Control Transfers in Device Controller Mode

(1) Setup Stage

The USBFS sends an ACK response to a normal setup packet for the USBFS. The USBFS operates in the setup stage as follows. On receiving a new setup packet, the USBFS sets the following bits:

- Sets the INTSTS0.VALID bit to 1
- Sets the DCPCTR.PID[1:0] bits to NAK
- Sets the DCPCTR.CCPL bit to 0.

When the USBFS receives a data packet following a setup packet, it stores the USB request parameters in USBREQ, USBVAL, USBINDX, and USBLENG.

Before performing the response processing for a control transfer, set the VALID flag to 0. When the VALID bit = 1, PID = BUF cannot be set, and the data stage cannot be terminated.

Using the VALID bit function, the USBFS can suspend the current request processing on receiving a new USB request during a control transfer and return a response to the latest request.

In addition, the USBFS automatically detects the direction bit (bmRequestType bit 8) and the request data length (wLength) in the received USB request. It distinguishes between control read transfers, control write transfers, and no-data control transfers, and it controls stage transitions. For an incorrect sequence, a sequence error occurs in the control transfer stage transition interrupt, and the interrupt is reported to the software. For a diagram of the stage control by the USBFS, see Figure 28.16.

(2) Data Stage

The DCP must be used to execute data transfers for received USB requests. Before accessing the DCP FIFO buffer, specify the access direction in the CFIFOSEL.ISEL bit.

If the transfer data is larger than the size of the DCP FIFO buffer, execute the data transfer using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

(3) Status Stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to BUF.

After this setting is made, the USBFS automatically executes the status stage based on the data transfer direction determined at the setup stage. The specific is executed as follows:

- For control read transfers
  The USBFS receives a zero-length packet from the USB host and transmits an ACK response.
- For control write transfers and no-data control transfers
  The USBFS transmits a zero-length packet and receives an ACK response from the USB host.

(4) Control Transfer Auto Response Function

The USBFS automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.

- bmRequestType is not 00h: Any transfer other than a control write transfer
- wIndex is not 00h: Request error
- wLength is not 00h: Any transfer other than a no-data control transfer
- wValue is larger than 7Fh: Request error
- INTSTS0.DVSQ[2:0] are 011b (Configured state): Control transfer of a device state error.

For all requests other than the SET_ADDRESS request, a response is required from the associated software.

28.3.10 Bulk Transfers (Pipes 1 to 5)

The FIFO buffer usage (single/double buffer setting) is configurable for bulk transfers. The USBFS provides the following functions for bulk transfers:

- BRDY interrupt function (PIPECFG.BFRE bit), see section 28.3.3.1, (2) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 1.
- Transaction count function (PIPEnTRE.TRENB, TRCLR, and PIPEnTRN.TRNCNT[15:0] bits), see section 28.3.4.5, Transaction Counter for Pipes 1 to 5 in the Receiving Direction.
- Response PID = NAK function (PIPECFG.SHTNAK bit), see section 28.3.4.8, Response PID = NAK Function.
- Auto response mode (PIPEnCTR.ATREPM bit), see section 28.3.4.9, Auto Response Mode.

28.3.11 Interrupt Transfers (Pipes 6 to 9)

In device controller mode, the USBFS performs interrupt transfers based on the timing dictated by the host controller.

In host controller mode, the software can set the timing for issuing tokens using the interval counter.

28.3.11.1 Interval Counter for Interrupt Transfers in Host Controller Mode

Specify the transaction interval for interrupt transfers in the PIPEPERI.IITV[2:0] bits. The USBFS issues interrupt transfer tokens based on the interval.

(1) Counter Initialization

The USBFS initializes the interval counter under the following conditions:

- Power-on reset
  This initializes the IITV[2:0] bits.
- FIFO buffer initialization using the PIPEnCTR.ACLRM bit:
  This does not initialize the IITV[2:0] bits, but does initialize the count value. Setting the PIPEnCTR.ACLRM bit to 0 starts counting from the value set in IITV[2:0].

The interval counter is not initialized in the following case:

- USB bus reset or USB suspended:
  The IITV[2:0] bits are not initialized. Setting 1 to the DVSTCTR0.UACT bit starts counting from the value saved before entering the USB bus reset state or USB Suspend state.

(2) Operation When Tokens Cannot Be Transmitted or Received Even on Token Generation

No token is generated in the following cases even at token generation time. In these cases, the USBFS tries to execute the transaction in the next interval.

- When the PID is set to NAK or STALL
- When the FIFO buffer is full at token transmit time in the receiving (IN) direction
- When there is no data to be transmitted in the FIFO buffer at token transmit time in the transmitting (OUT) direction.

28.3.12 Isochronous Transfers (Pipes 1 and 2)

The USBFS provides the following functions for isochronous transfers:

- Notification of isochronous transfer error
- Interval counter (specified in the PIPEPERI.IITV[2:0] bits)
- Isochronous IN transfer data setup control (IDLY function)
- Isochronous IN transfer buffer flush function (specified in the PIPEPERI.IFIS bit).

28.3.12.1 Error Detection in Isochronous Transfers

The USBFS provides a function for detecting the errors described in this section, so that when errors occur in isochronous transfers, they can be controlled by the software. Table 28.23 and Table 28.24 show the priority order for errors detected by the USBFS and the associated interrupts.

(a) PID errors

- The PID value of the received packet is invalid.

(b) CRC errors and bit stuffing errors

- A CRC error is found in a received packet or the bit stuffing is invalid.

(c) Maximum packet size exceeded

- The data size of the received packet exceeds the specified maximum packet size.

(d) Overrun and underrun errors

In host controller mode:

- The FIFO buffer is full at token transmit time in the IN (receiving) direction
- There is no data to be sent in the FIFO buffer at token transmit time in the OUT (transmitting) direction.

In device controller mode:

- There is no data to be sent in the FIFO buffer at token receive time in the IN (transmitting) direction
- The FIFO buffer is full at token receive time in the OUT (receiving) direction.

(e) Interval Errors

In device controller mode, the following cases are treated as an interval error:

- Failure to receive an IN token in the interval frame during an isochronous IN transfer
- Failure to receive an OUT token in the interval frame during an isochronous OUT transfer.

Table 28.23 Error detection for token transmission and reception

Detection priority	Error	Generated interrupt and status
1	PID errors	No interrupts are generated in either host or device controller mode (ignored as a corrupted packet)
2	CRC errors and bit stuffing errors	No interrupts are generated in either host or device controller mode (ignored as a corrupted packet)
3	Overrun and underrun errors	An NRDY interrupt is generated to set the FRMNUM.OVRN bit to 1 in both host and device controller modes. In device controller mode, a zero-length packet is transmitted in response to an IN token. No data packets are received in response to OUT token.
4	Interval errors	An NRDY interrupt is generated in device controller mode. No interrupt is generated in host controller mode.

Table 28.24 Error detection for data packet reception

Detection priority	Error	Generated interrupt and status
1	PID errors	No interrupts are generated (ignored as a corrupted packet)
2	CRC errors and bit stuffing errors	An NRDY interrupt is generated and the FRMNUM.CRCE bit is set to 1 bit in both host and device controller modes
3	Maximum packet size exceeded errors	A BEMP interrupt is generated and the PID[1:0] bits set to STALL in both host and device controller modes

28.3.12.2 DATA-PID

In device controller mode, the USBFS responds as follows to a received PID:

(1) IN Direction

- DATA0: Transmitted as data packet PID
- DATA1: Not transmitted
- DATA2: Not transmitted
- mDATA: Not transmitted.

(2) OUT Direction

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets ignored
- mDATA: Packets ignored.

28.3.12.3 Interval Counter

The isochronous transfer interval can be set in the PIPEPERI.IITV[2:0] bits. In device controller mode, the interval counter enables the functions as shown in Table 28.25. In host controller mode, the USBFS generates the token issuance timing, and the interval counter operation is the same as that for interrupt transfers.

Table 28.25 Interval counter function in device controller mode

Transfer direction	Function	Conditions for detection
IN	Flushes transmit buffer	Failure to receive an IN token successfully in the interval frame during an isochronous IN transfer
OUT	Notifies that a token not being received	Failure to receive an OUT token successfully in the interval frame during an isochronous OUT transfer

The interval count is carried out when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is damaged. The frame interval that can be set is the 2^{IITV} frames.

(1) Counter Initialization in Device Controller Mode

The USBFS initializes the interval counter under the following conditions:

- Power-on reset:
  This initializes the PIPEPERI.IITV[2:0] bits.
- FIFO buffer initialization using the ACLRM bit:
  This does not initialize the IITV[2:0] bits, but does initialize the count value.

After the interval counter is initialized, the interval count starts under either of the following conditions when a packet is transferred successfully:

- An SOF is received after data is transmitted in response to an IN token when PID = BUF
- An SOF is received after data is received in response to an OUT token when PID = BUF.

The interval counter is not initialized in the following conditions:

- When the PID[1:0] bits are set to NAK or STALL
  This does not stop the interval timer. The USBFS attempts the transaction in the next interval.
- When the USB bus is reset or USBFS is suspended
  This does not initialize the IITV[2:0] bits. When an SOF is received, the interval counter starts counting from the value set before SOF was received.

(2) Interval Counting and Transfer Control in Host Controller Mode

The USBFS controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bit settings. Specifically, the USBFS issues a token for a selected pipe once every 2^{IITV} frames.

The USBFS starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits are set to BUF by the software.

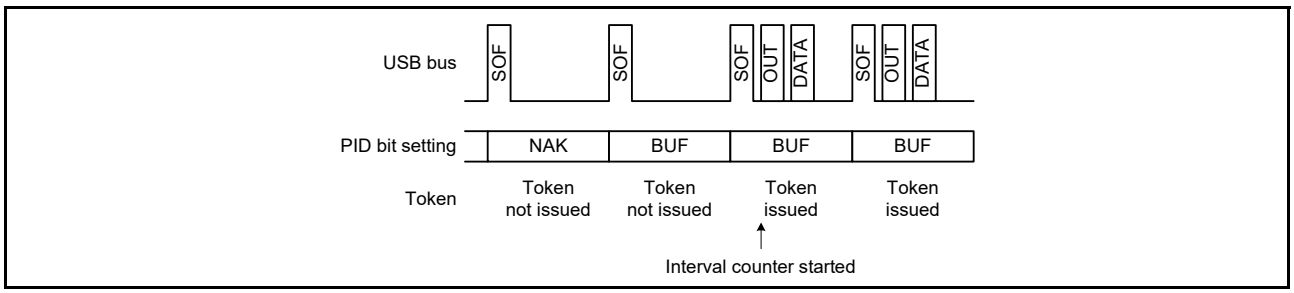


Figure 28.17 Token issuance when IITV = 0

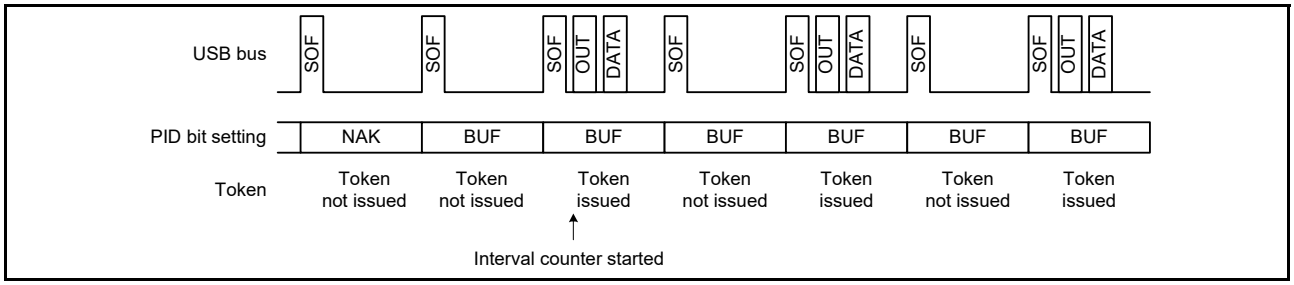


Figure 28.18 Token issuance when IITV = 1

When the selected pipe is set for isochronous transfers, the USB carries out the following operation in addition to controlling the token issuance interval. The USB issues a token even when the NRDY interrupt generation condition is satisfied.

(a) When the Selected Pipe is for Isochronous IN Transfers

The USBFS generates an NRDY interrupt when the USBFS issues an IN token but does not receive a packet successfully from a peripheral device (no response or packet error).

The USBFS sets the FRMNUM.OVRN bit to 1, generating an NRDY interrupt, when the time to issue an IN token comes while the USBFS cannot receive data because the FIFO buffer is full, because the CPU or DMAC/DTC is too slow in reading data from the FIFO buffer.

(b) When the Selected Pipe is for Isochronous OUT Transfers

The USBFS sets the OVRN bit to 1, generating an NRDY interrupt and transmitting a zero-length packet, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer, because the CPU or DMAC/DTC is too slow in writing data to the FIFO buffer.

The token issuance interval is reset on any of the following conditions:

- When the USBFS is reset through a reset pin
  This initializes the IITV[2:0] bits.
- When the PIPEnCTR.ACLRM bit is set to 1 by the software.

(3) Interval Counting and Transfer Control in Device Controller Mode

(a) When the Selected Pipe is for Isochronous OUT Transfers

The USBFS generates an NRDY interrupt when it fails to receive a data packet within the interval set by the PIPEPERI.IITV[2:0] bits.

The USBFS also generates an NRDY interrupt when it fails to receive data because of a CRC error or other errors contained in the data packet or because of FIFO buffer is full.

The NRDY interrupt is generated on SOF packet reception. Even if the SOF packet is corrupted, internal interpolation allows the interrupt to be generated when the SOF packet is received. However, when the IITV bits are set to a value other than 0, the USBFS generates an NRDY interrupt on receiving an SOF packet for every interval after interval counting starts.

When the PID[1:0] bits are set to NAK by the software after starting the interval timer, the USBFS does not generate an NRDY interrupt on receiving an SOF packet.

The timing for starting interval counting depend on the IITV[2:0] setting as follows:

- When the IITV[2:0] bits = 0:
  Interval counting starts at the next frame after the software changes the PID[1:0] bits of the selected pipe to BUF.

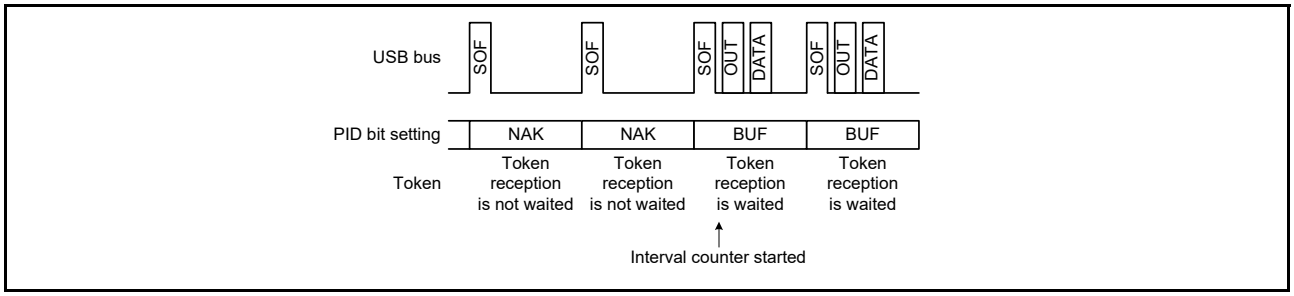


Figure 28.19 Relationship between frames and expected token reception when IITV = 0

- When the IITV ≠ 0: The interval counting starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe are modified to BUF.

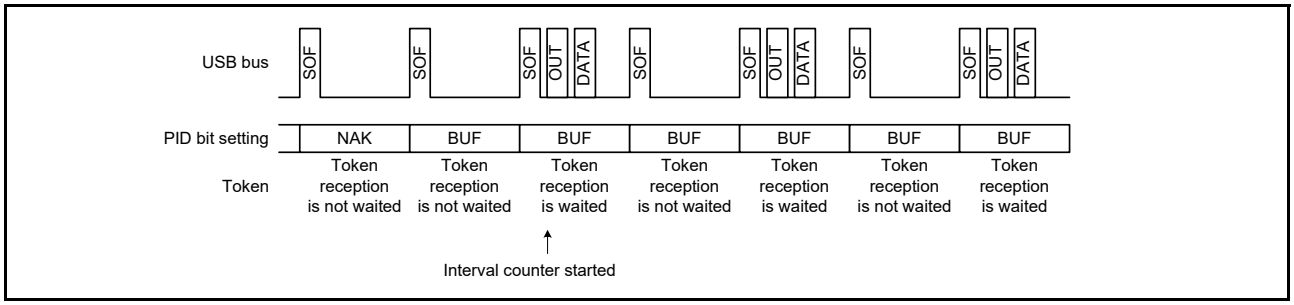


Figure 28.20 Relationship between frames and expected token reception when IITV ≠ 0

(b) When the Selected Pipe is for Isochronous IN Transfers

The PIPEPERI.IFIS bit must be 1 for this use case. When IFIS = 0, the USBFS transmits a data packet in response to a received IN token regardless of the PIPEPERI.IITV[2:0] setting.

When IFIS is 1 and there is data to be transmitted in the FIFO buffer, the USBFS clears the FIFO buffer when it fails to receive an IN token in the frame at the interval set in the IITV[2:0] bits.

The USBFS also clears the FIFO buffer when it fails to receive an IN token successfully because of a bus error, such as a CRC error, contained in the IN token.

The FIFO buffer is cleared on SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the FIFO buffer to be cleared when the SOF packet is received.

The timing to start interval counting depends on the IITV[2:0] setting, as with OUT transfers.

The interval is counted on any of the following conditions in device controller mode:

- When a hardware reset is applied to the USBFS (which also sets the IITV[2:0] bits to 000b)
- When the PIPEnCTR.ACLRM bit is set to 1 by the software
- When the USBFS detects a USB bus reset.

(4) Transmit Data Setup for Isochronous Transfers in Device Controller Mode

With isochronous data transmission using the USBFS in device controller mode, after data is written to the FIFO buffer, a data packet can be transmitted in the first frame after the SOF packet is detected. This isochronous transfer transmit data setup function can identify the frame that started transmission.

When the double buffering is used, transmission is only enabled for the buffer where data writing was completed first, even after the data write to both buffers is complete. Accordingly, even if multiple IN tokens are received, only the one packet of FIFO buffer data is transmitted.

When the FIFO buffer is ready to transmit data when an IN token is received, the data is transferred and a normal response is returned. However, if the FIFO buffer cannot transmit data, a zero-length packet is transmitted and an underrun error occurs.

Figure 28.21 shows an example transmission using the isochronous transfer transmission data setup function when IITV = 0 (every frame) is set.

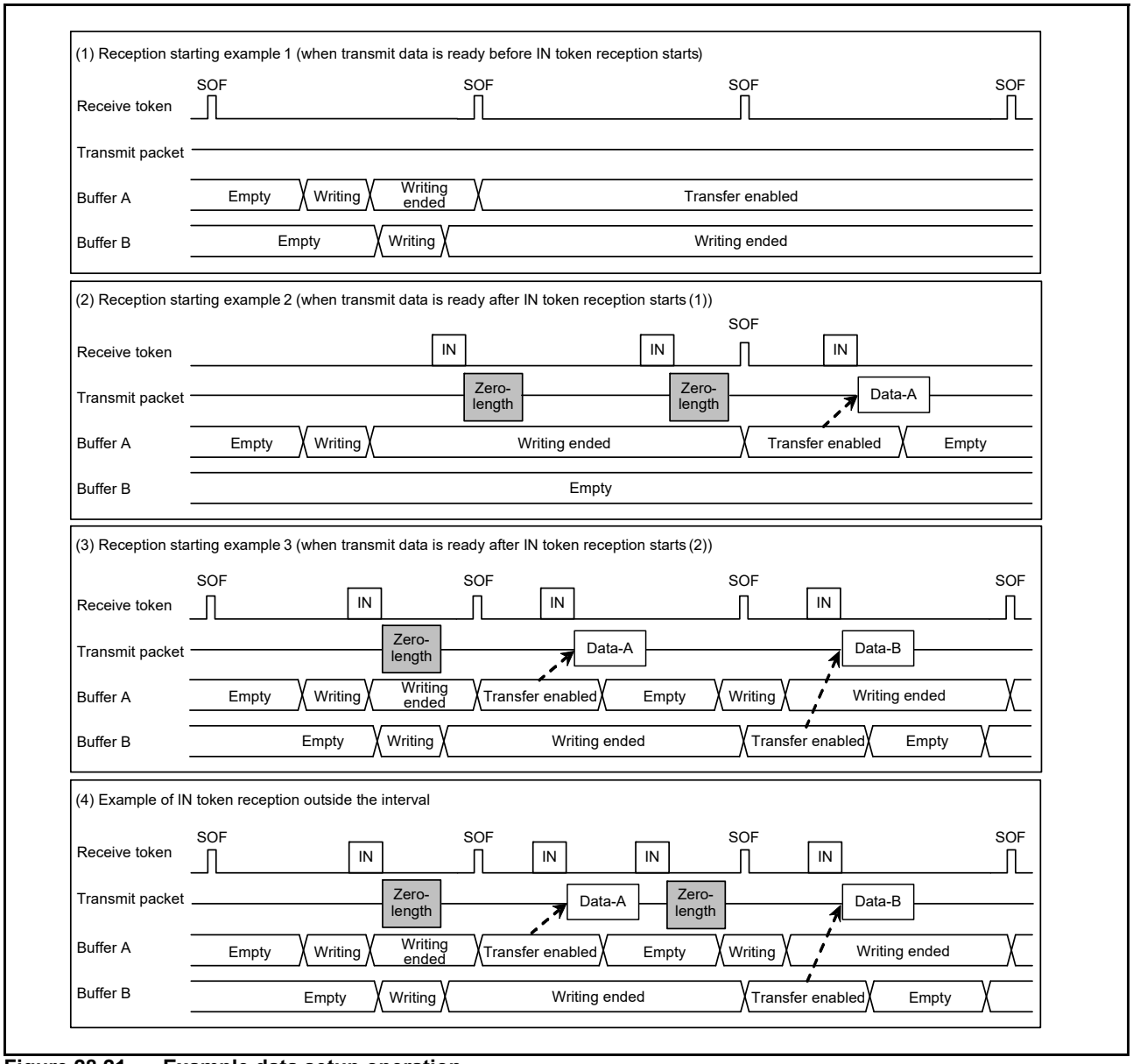


Figure 28.21 Example data setup operation

(5) Transmit Buffer Flush for Isochronous Transfers in Device Controller Mode

In device controller mode during isochronous data transmission, if the USBFS receives an SOF packet for the next frame without receiving an IN token in the interval frame, it operates as if the IN token is corrupt and clears the buffer that is enabled for transmission, putting that buffer in the writing enabled state.

When double buffering is used and writing to both buffers is complete, the cleared FIFO buffer is assumed to be the one where the data was transmitted in the interval frame, and transmission is enabled for the FIFO buffer that was not cleared on SOF packet reception.

The timing of the buffer flush function depends on the PIPEPERI.IITV[2:0] setting as follows:

- When IITV = 0:
  The buffer flush operation starts from the first frame after the pipe is enabled.
- When IITV ≠ 0:
  The buffer flush operation starts after the first normal transaction.

Figure 28.22 shows an example buffer flush. When an unanticipated token is received before the interval frame, the USBFS sends the write data or a zero-length packet as an underrun error, depending on the data setup status.

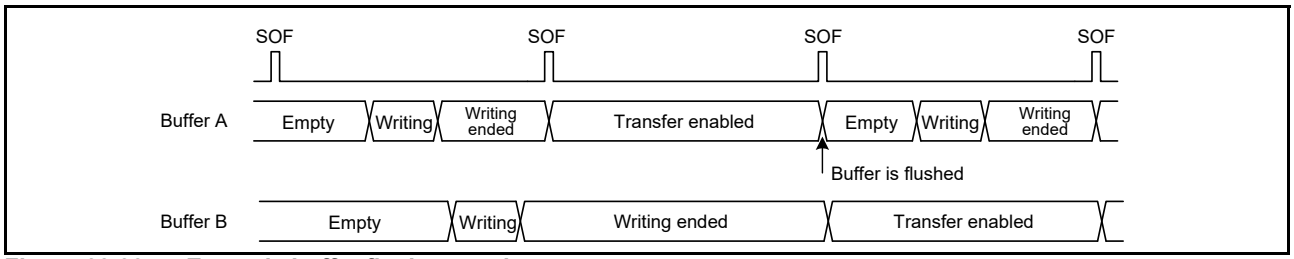


Figure 28.22 Example buffer flush operation

Figure 28.23 shows an example interval error occurrence. There are five types of interval errors, as shown in the figure. An interval error occurs at timing ① in the figure, and the buffer flush function is activated.

If an interval error occurs during an IN transfer, the buffer flush function is activated. If it occurs during an OUT transfer, an NRDY interrupt is generated. Use the FRMNUM.OVRN bit to distinguish between this and NRDY interrupts triggered by received packet errors and overrun errors.

For tokens that are shaded in the figure, responses are returned based on the FIFO buffer status.

- IN direction:
  - If the buffer is ready to transfer data, the data is transferred and a normal response is returned
  - If the buffer is not ready to transfer data, a zero-length packet is transmitted and an underrun error occurs.
- OUT direction:
  - If the buffer is ready to receive data, the data is received and a normal response is returned
  - If the buffer is not ready to receive data, the received data is discarded and an overrun error occurs.

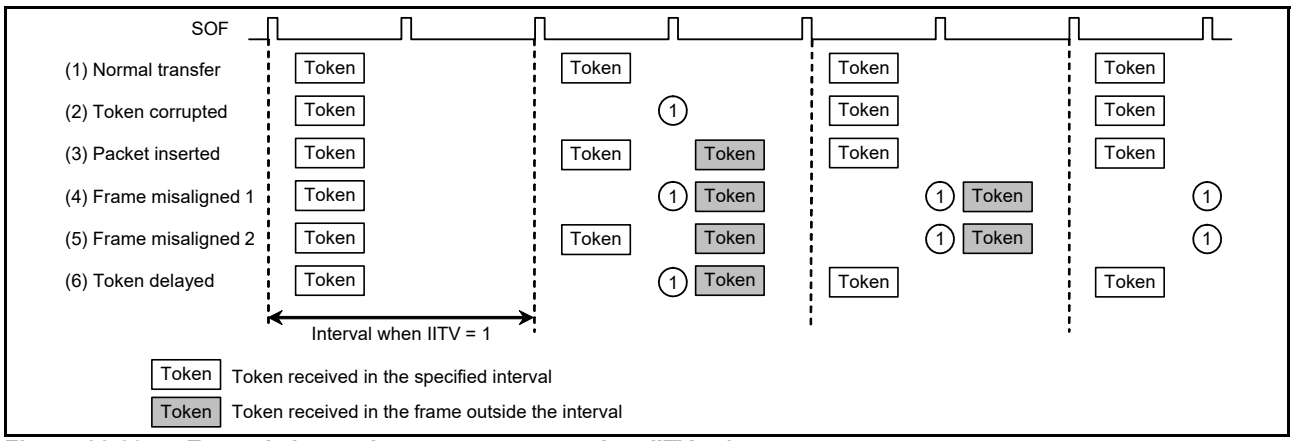


Figure 28.23 Example interval error occurrence when IITV = 1

28.3.13 SOF Interpolation Function

In device controller mode, if packet reception is disabled at intervals of 1 ms because the SOF packet is corrupted or missing, the USBFS interpolates the SOF. SOF interpolation begins when the USBE and SCKE bits in SYSCFG are set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions:

- MCU reset
- USB bus reset
- Suspend state detection.

The SOF interpolation operates as follows:

- The interpolation function is not activated until an SOF packet is received.
- When the first SOF packet is received, interpolation is performed by counting 1 ms on the 48-MHz internal clock
- When the second and subsequent SOF packets are received, interpolation is performed at the previous reception interval
- Interpolation is not performed in the suspended state or on reception of a USB bus reset.

The USBFS supports the following functions controlled by SOF packet reception. These functions operate normally with SOF interpolation if the SOF packet is missing:

- Updating of the frame number
- SOFR interrupt timing
- Isochronous transfer interval count.

If an SOF packet is missing during full-speed operation, the FRMNUM.FRNM[10:0] bits are not updated.

28.3.14 Pipe Schedule

28.3.14.1 Conditions for Generating Transactions

In host controller mode and when the DVSTCTR0.UACT bit is set to 1, the USBFS generates transactions under the conditions shown in Table 28.26.

Table 28.26 Conditions for generating transactions

	Conditions for generation				
Transaction	**DIR**	**PID**	**IITV0**	**Buffer state**	**SUREQ**
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	—*1
	OUT	BUF	Invalid	Transmit data exists	—*1
Interrupt transfer	IN	BUF	Valid	Receive area exists	—*1
	OUT	BUF	Valid	Transmit data exists	—*1
Isochronous transfer	IN	BUF	Valid	*2	—*1
	OUT	BUF	Valid	*3	—*1

Note 1. An em dash (—) in the table indicates that the condition is unrelated to the generating of tokens. "Valid" indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter. "Invalid" indicates that a transaction is generated regardless of the interval counter.

Note 2. This indicates that a transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.

Note 3. This indicates that a transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

28.3.14.2 Transfer Schedule

This section describes the transfer scheduling within a frame of the USBFS. After the USBFS sends an SOF, the transfer is carried out in the following sequence:

1. Execution of periodic transfers:
   A pipe is searched for in the order of pipe 1 → pipe 2 → pipe 6 → pipe 7 → pipe 8 → pipe 9, and then if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.
2. Setup transactions for control transfers:
   The DCP is checked, and if a setup transaction is possible, it is sent.
3. Execution of bulk transfers, control transfer data stages, and control transfer status stages:
   A pipe is searched for in the order of DCP → pipe 1 → pipe 2 → pipe 3 → pipe 4 → pipe 5, and then if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.
   When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

28.3.14.3 Enabling USB Communication

Setting the DVSTCTR0.UACT bit to 1 initiates a SOF transmission, and transaction generation is enabled. Setting the UACT bit to 0 stops SOF transmission and the suspended state is invoked. If the UACT setting is changed from 1 to 0, processing stops after the next SOF is sent.

28.3.15 Battery Charging Detection Processing

It is possible to control the processing for data contact detection (D+ line contact check), primary detection (charger detection), and secondary detection (charger verification), which are defined in the battery charging specification. The following section describes the required operations for an individual function device and a host device.

28.3.15.1 Processing in Device Controller Mode

The following processing is required when operating the USBFS as a portable device for battery charging:

1. Detect when the data lines (D+ and D-) have made contact and start the processing for primary detection.
2. After primary detection starts, wait 40 ms for masking, then check the D- voltage level to confirm the primary detection result.
3. If the charger is detected during primary detection, start secondary detection.
4. After secondary detection starts, wait 40 ms for masking, then check the D+ voltage level to confirm the secondary detection result.

For step 1, after VBUS is detected using the VBINT and VBSTS bits:

1. Wait for 300 to 900 ms, then set the VDPSRCE0 and IDMSINKE0 bits in the USBBCCTRL0 register.
2. Set the IDPSRCE0 bit.
3. After a change from high to low on the D+ line is detected using the LNST bits, clear the IDPSRCE0 bit, and set the VDPSRCE0 and IDMSINKE0 bits simultaneously*1.

For step 2, set the VDPSRCE0 and IDMSINKE0 bits and wait 40 ms, then use the CHGDETSTS0 bit to verify the primary detection result*2.

For step 3, if the CHGDETSTS0 bit is set in step 2, verify that the charger is detected, then clear the VDPSRCE0 and IDMSINKE0 bits and set the VDMSRCE0 and IDPSINKE0 bits.

For step 4, set the VDMSRCE0 and IDPSINKE0 bits and wait for 40 ms, then use the PDDETSTS0 bit to verify the secondary detection result.

Figure 28.24 shows the process flow.

Note 1. The battery charging specification describes two implementation methods for data contact detection (D+/D- line contact check). One of the methods is to detect a change to logic low due to the pull-down resistor of the host device when the D+ and D- lines have made contact with the target while the D+ line is held at logic high by applying a current of 7 to 13 µA on the D+ line. The other method is to wait for 300 to 900 ms after VBUS is detected.

Note 2. During primary detection, when the voltage on the D- line is detected to be 0.25 to 0.4 V or above and 0.8 to 2.0 V or below, the target device is recognized as the host device for battery charging, that is, charging downstream port. When using a PHY in which the CHGDETSTS bit only indicates that the voltage on the D- line is 0.25 to 0.4 V or above, add the processing to check that the voltage on D- line is 0.8 V to 2.0 V or below using the LNST bits, as required.

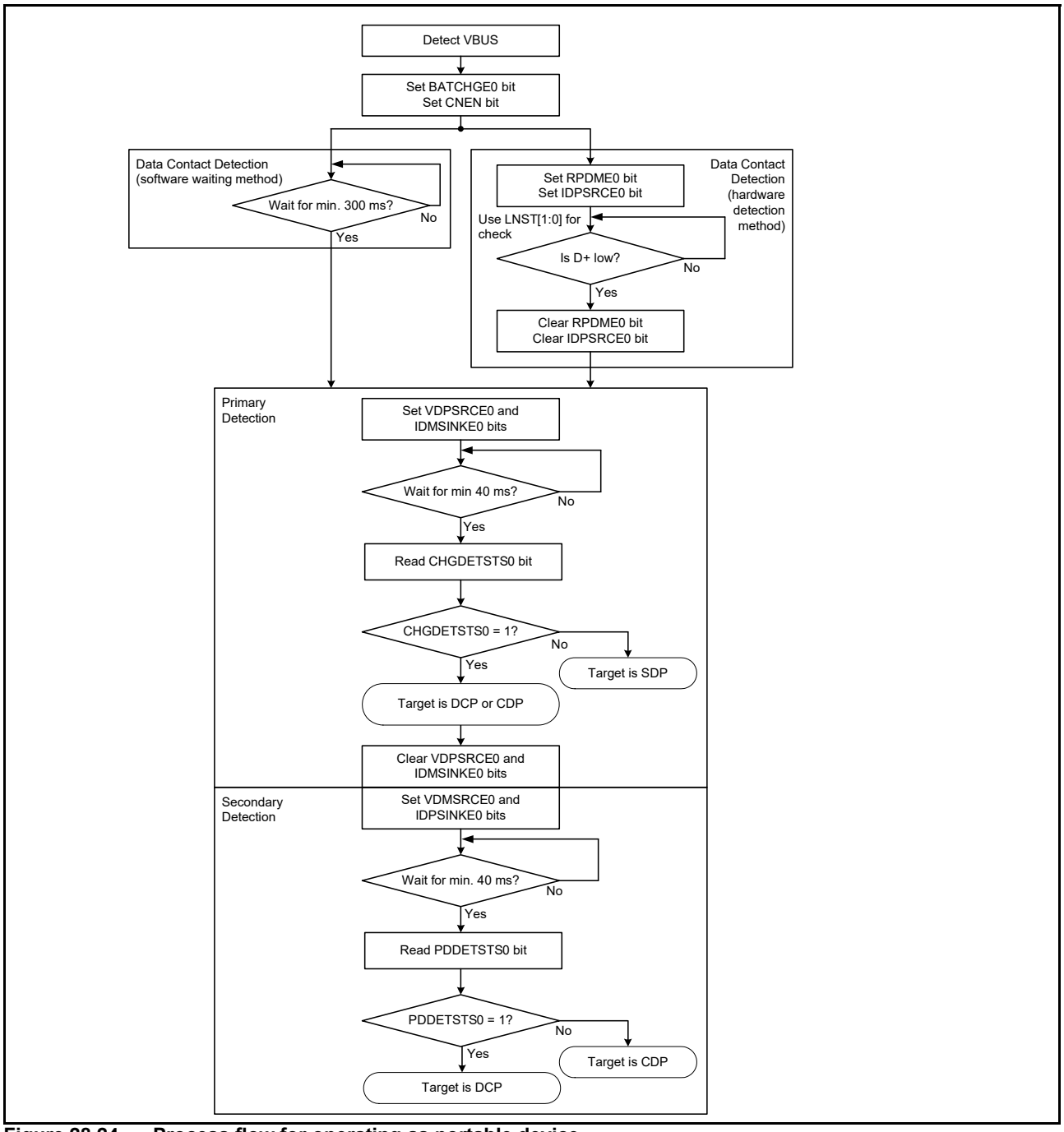


Figure 28.24 Process flow for operating as portable device

28.3.15.2 Processing when Host Controller is Selected

The following processing is required when operating the USBFS as a charging downstream port for battery charging:

1. Start driving the VBUS.
2. Enable the portable device detection circuit.
3. Monitor the portable device detection signal, and start driving the D- line if the detection signal is high.

4. Detect when the portable device detection signal is a low level and stop driving the D- line.

The following processing can also be used in associated with the battery charging specification:

a. After disconnection is detected, start driving the D- line within 200 ms

b. After connection is detected, stop driving the D- line within 10 ms.

The D- line must be driven to allow the portable device to detect the primary detection described in section 28.3.15.1, Processing in Device Controller Mode. Steps 1 to 4 apply when the portable device detection function is provided by hardware. This method is to drive the D- line when the portable device is detected.

Steps a and b apply when the portable device function is not provided or available by hardware. Regardless of detection of the portable device, the D- line is driven in the disconnected state and not in the connected state. In the battery charging specification, either of these methods can be used.

For steps 3 and 4, after a change in the portable device detection signal is detected using the PDDETINT interrupt, the current signal state can be confirmed by reading the PDDETSTS0 bit. Steps a and b can be performed only in a software timer.

Figure 28.25 show the process flow for steps 1 to 4 and the process flow for steps a to b, respectively.

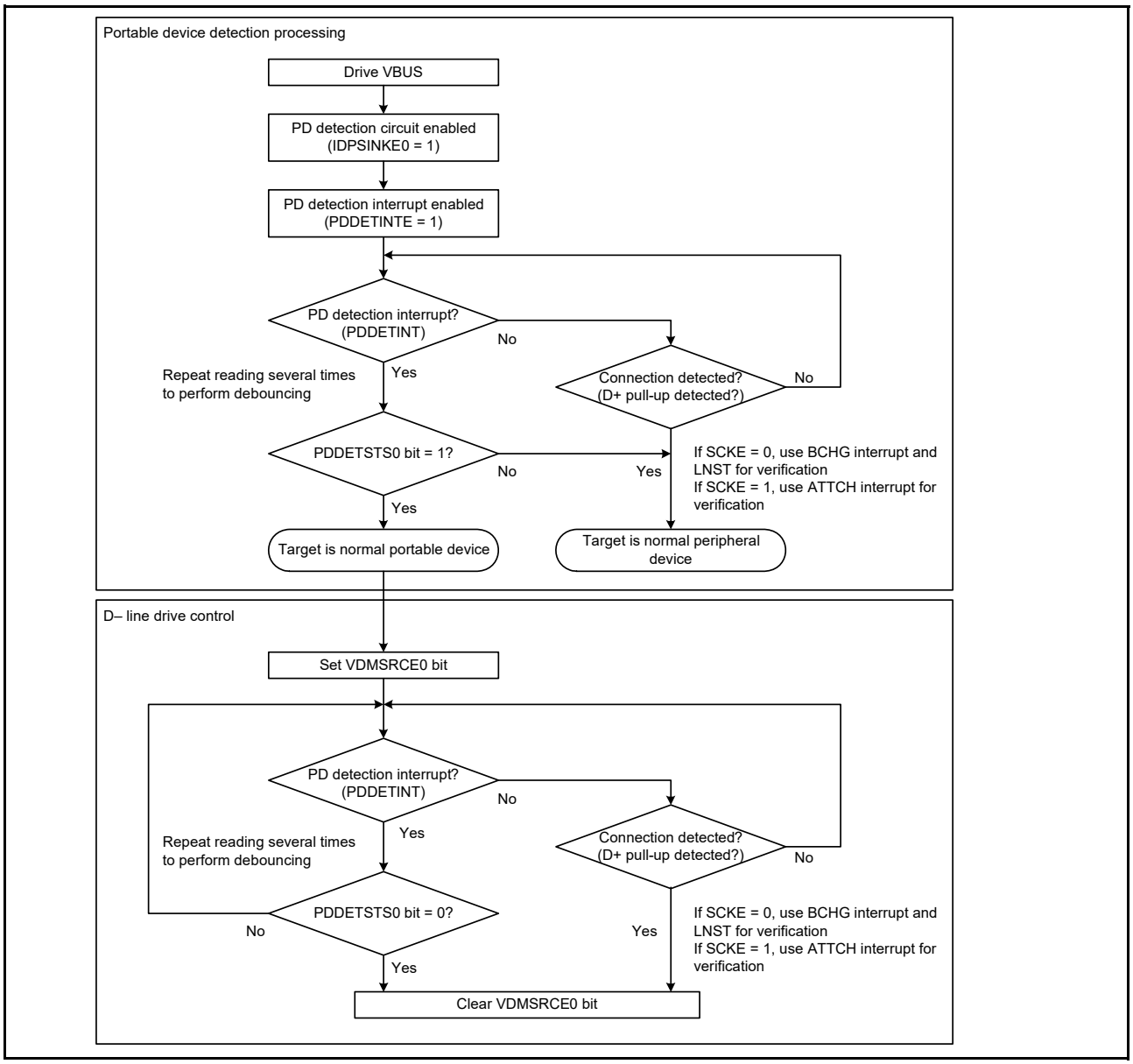


Figure 28.25 Process flow for operating as charging downstream port (steps 1 to 4)

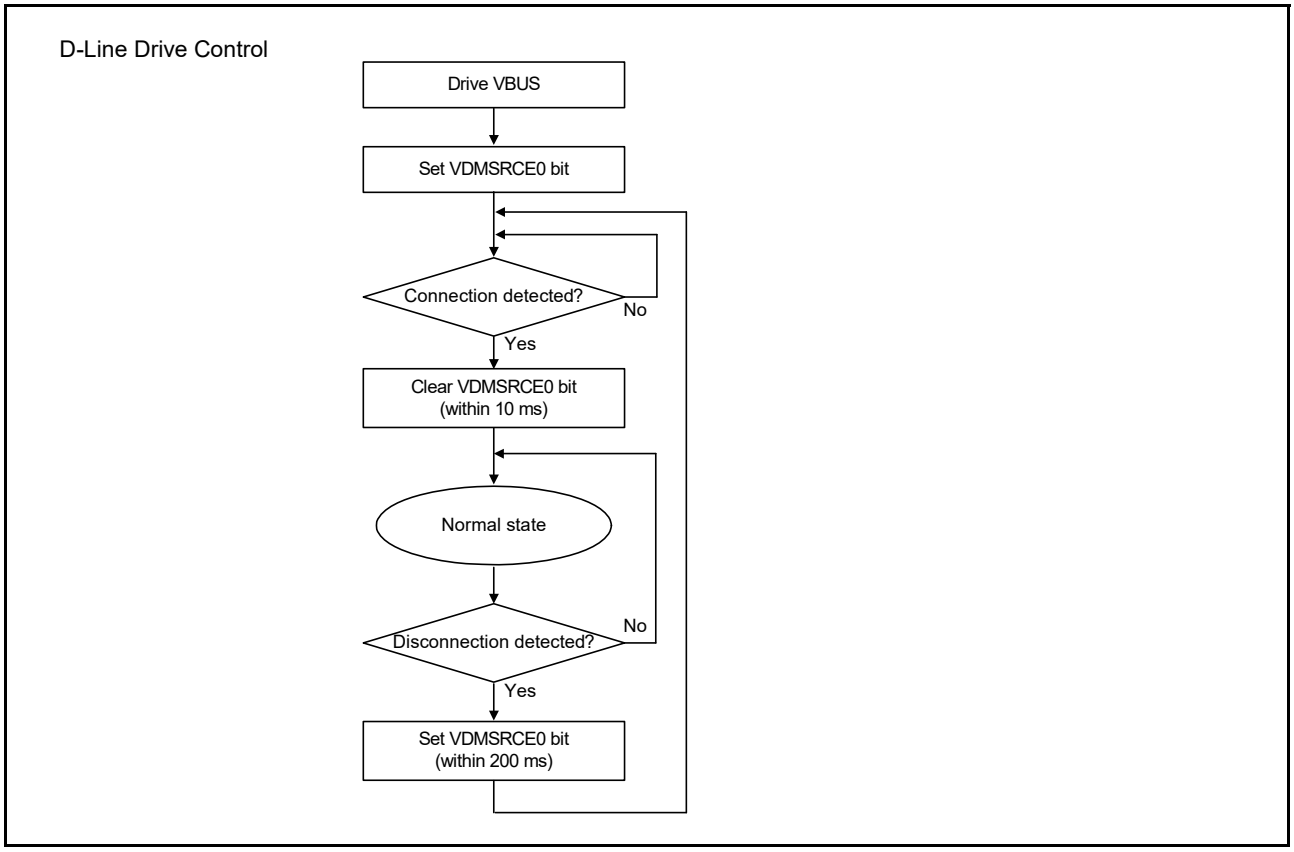


Figure 28.26 Process flow for operating as charging downstream port (steps a to b)

28.4 Usage Notes

28.4.1 Settings for the Module-Stop State

USBFS operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The USBFS is initially stopped after reset. Register access is enabled by releasing the module-stop state. For details, see section 11, Low Power Modes.

28.4.2 Clearing the Interrupt Status Register on Exiting Software Standby Mode

Because the input buffer is always enabled in Software Standby mode, an unexpected interrupt might occur under the following conditions:

- When the interrupt is enabled in Normal mode
- When the interrupt is disabled in Software Standby mode
- When the input level of the pin that cancels Software Standby is changed in Software Standby mode.

These conditions might cause the associated interrupt flag in the Interrupt Status Register to set unexpectedly. After the MCU exits the Software Standby mode, the unexpected interrupt might be sent to the interrupt controller. To avoid this, always clear the INTSTS0 and INTSTS1 registers in the canceling sequence.

28.4.3 Clearing the Interrupt Status Register after Setting Up the Port Function

The input buffer is disabled before the PmnPFS.PSEL and PmnPFS.PMR port is set up, so the internal signal is fixed high or low. The input buffer is enabled after the port is set so that the external pin state is propagated to the MCU. An unexpected interrupt might occur at this time, causing the VBINT and OVRCR bits in INTSTS0 and INTSTS1, or other interrupt status flags to set to 1. To avoid a malfunction, always clear the INTSTS0 and INTSTS1 registers after setting up the port.

29. Serial Communications Interface (SCI)

29.1 Overview

The Serial Communications Interface (SCI) is configurable to five 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. Each SCI channel 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. Table 29.1 lists the SCI specifications, Figure 29.1 shows the block diagram, and Table 29.2 lists the I/O pins by mode.

Note: In this section, PCLK refers to PCLKA.

Table 29.1 SCI specifications (1 of 2)

Parameter		Description
Serial communication modes		• Asynchronous • Clock synchronous • Smart card interface • Simple IIC • Simple SPI.
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
I/O pins		See Table 29.2.
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, and 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)
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, overrun, and framing errors
	Hardware flow control	CTSn_RTSn pin can be used in controlling transmission/reception
	Transmission/reception	Selectable as either 1-stage register or 16-stage FIFO
	Address match	Interrupt request/event output can be issued when detecting a match between receive data and the value of the compare match register
	Address non-match (SCI0 only) receive data	The snooze end request can be issued when detecting a non-match between the received data and the value of the compare match register
	Start-bit detection	Selectable as either low level or falling edge
	Break detection	When a framing error occurs, a break can be detected by reading the SPTR register
	Clock source	An internal or external clock can be selected
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Multi-processor communications function	Serial communication between multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters

Table 29.1 SCI specifications (2 of 2)

Parameter		Description
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Clock source	An internal clock (master mode) or external clock (slave mode) can be selected
	Hardware flow control	CTSn_RTSn pin can be used to control transmission/reception
	Transmission/reception	Selectable as either 1-stage register or 16-stage FIFO
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are 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
	Detection of errors	Overrun error
	Clock source	An internal clock (master mode) or external clock (slave mode) can be selected
	SS input pin function	Applying the high level to the SSn pin can cause the output pins to enter the high-impedance state
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors
Event link function		Error (receive error or error signal detection) event output (SCIn_ERI*1)
		Receive data full event output (SCIn_RXI*1, *2)
		Transmit data empty event output (SCIn_TXI*1, *2)
		Transmit end event output (SCIn_TEI*1, *2)
		Address match event output (SCIn_AM*1)

Note 1. Channel number (n = 0 to 4, 9)
Note 2. Using this event link function is prohibited when the FIFO operation is selected in asynchronous mode.

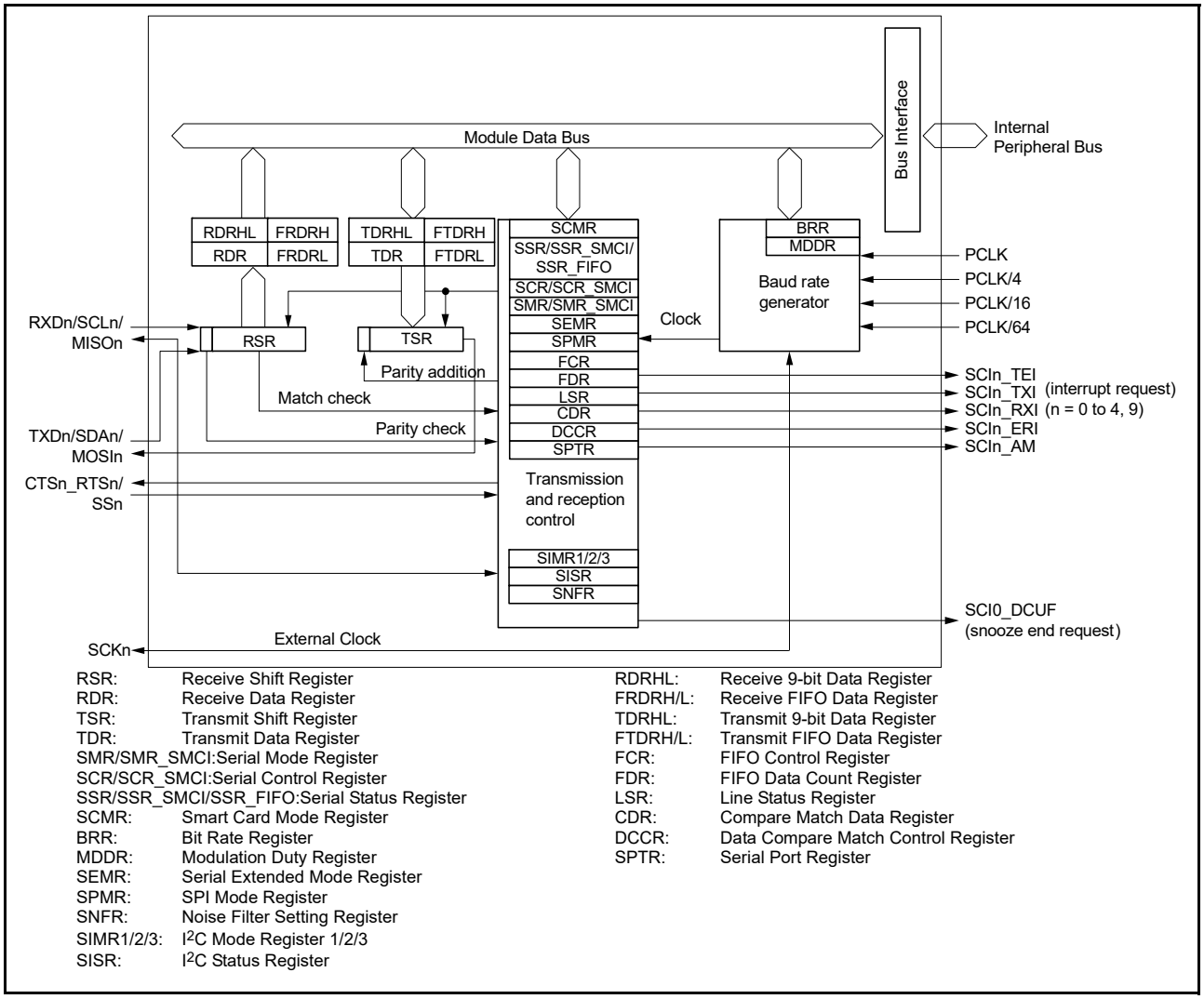


Figure 29.1 SCI block diagram

Table 29.2 SCI input/output pins (1 of 2)

Channel	Pin name	Input/output	Function
SCI0	SCK0	Input/output	SCI0 clock input/output
	RXD0/SCL0/ MISO0	Input/output	SCI0 receive data input SCI0 I2C clock input/output SCI0 slave transmit data input/output
	TXD0/SDA0/ MOSI0	Input/output	SCI0 transmit data output SCI0 I2C data input/output SCI0 master transmit data input/output
	SS0/CTS0_RTS0	Input/output	SCI0 chip select input, active-low SCI0 transfer start control input/output, active-low
SCI1	SCK1	Input/output	SCI1 clock input/output
	RXD1/SCL1/ MISO1	Input/output	SCI1 receive data input SCI1 I2C clock input/output SCI1 slave transmit data input/output
	TXD1/SDA1/ MOSI1	Input/output	SCI1 transmit data output SCI1 I2C data input/output SCI1 master transmit data input/output
	SS1/CTS1_RTS1	Input/output	SCI1 chip select input, active-low SCI1 transfer start control input/output, active-low

Table 29.2 SCI input/output pins (2 of 2)

Channel	Pin name	Input/output	Function
SCI2	SCK2	Input/output	SCI2 clock input/output
	RXD2/SCL2/ MISO2	Input/output	SCI2 receive data input SCI2 I^2C clock input/output SCI2 slave transmit data input/output
	TXD2/SDA2/ MOSI2	Input/output	SCI2 transmit data output SCI2 I^2C data input/output SCI2 master transmit data input/output
	SS2/CTS2_RTS2	Input/output	SCI2 chip select input, active-low SCI2 transfer start control input/output, active-low
SCI3	SCK3	Input/output	SCI3 clock input/output
	RXD3/SCL3/ MISO3	Input/output	SCI3 receive data input SCI3 I^2C clock input/output SCI3 slave transmit data input/output
	TXD3/SDA3/ MOSI3	Input/output	SCI3 transmit data output SCI3 I^2C data input/output SCI3 master transmit data input/output
	SS3/CTS3_RTS3	Input/output	SCI3 chip select input, active-low SCI3 transfer start control input/output, active-low
SCI4	SCK4	Input/output	SCI4 clock input/output
	RXD4/SCL4/ MISO4	Input/output	SCI4 receive data input SCI4 I^2C clock input/output SCI4 slave transmit data input/output
	TXD4/SDA4/ MOSI4	Input/output	SCI4 transmit data output SCI4 I^2C data input/output SCI4 master transmit data input/output
	SS4/CTS4_RTS4	Input/output	SCI4 chip select input, active-low SCI4 transfer start control input/output, active-low
SCI9	SCK9	Input/output	SCI9 clock input/output
	RXD9/SCL9/ MISO9	Input/output	SCI9 receive data input SCI9 I^2C clock input/output SCI9 slave transmit data input/output
	TXD9/SDA9/ MOSI9	Input/output	SCI9 transmit data output SCI9 I^2C data input/output SCI9 master transmit data input/output
	SS9/CTS9_RTS9	Input/output	SCI9 chip select input, active-low SCI9 transfer start control input/output, active-low

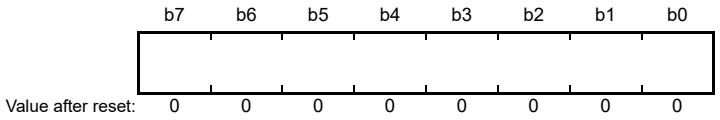
29.2 Register Descriptions

29.2.1 Receive Shift Register (RSR)

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 RDR register, RDRHL register, or the receive FIFO. The RSR register cannot be directly accessed by the CPU.

29.2.2 Receive Data Register (RDR)

Address(es): SCI0.RDR 4007 0005h, SCI1.RDR 4007 0025h, SCI2.RDR 4007 0045h, SCI3.RDR 4007 0065h, SCI4.RDR 4007 0085h, SCI9.RDR 4007 0125h



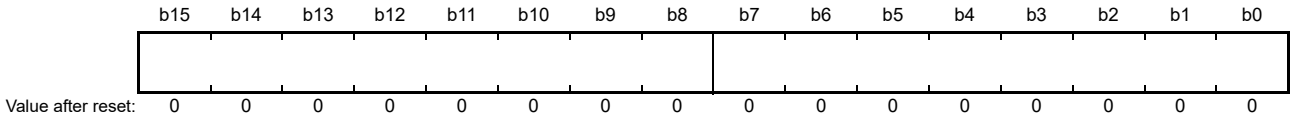
RDR is an 8-bit register that stores receive data. When one frame of serial data is received, the received serial data is transferred from RSR to RDR, and the RSR register can receive more data. Because RSR and RDR function as a double buffer, continuous receive operations can be performed.

Read the RDR register only once after a receive data full interrupt (SCIn_RXI) occurs.

Note: If the next frame of data is received before reading the receive data from RDR, an overrun error occurs. RDR cannot be written to by the CPU.

29.2.3 Receive 9-bit Data Register (RDRHL)

Address(es): SCI0.RDRHL 4007 0010h, SCI1.RDRHL 4007 0030h, SCI2.RDRHL 4007 0050h, SCI3.RDRHL 4007 0070h, SCI4.RDRHL 4007 0090h, SCI9.RDRHL 4007 0130h



RDRHL is a 16-bit register that stores receive 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. For example, access to RDRHL affects RDR. Access to RDRHL 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 RSR to the RDR/RDRHL registers, allowing RSR register to receive more data.

RSR and RDRHL 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.

Bits [9] to [15] of the RDRHL register are fixed to 0. These bits are read as 0. The write value should be 0.

29.2.4 Receive FIFO Data Register H, L, HL (FRDRH, FRDRL, FRDRHL)

Receive FIFO Data Register H (FRDRH)

Address(es): SCI0.FRDRH 4007 0010h, SCI1.FRDRH 4007 0030h, SCI2.FRDRH 4007 0050h, SCI3.FRDRH 4007 0070h, SCI4.FRDRH 4007 0090h, SCI9.FRDRH 4007 0130h

Receive FIFO Data Register L (FRDRL)

Address(es): SCI0.FRDRL 4007 0011h, SCI1.FRDRL 4007 0031h, SCI2.FRDRL 4007 0051h, SCI3.FRDRL 4007 0071h, SCI4.FRDRL 4007 0091h, SCI9.FRDRL 4007 0131h

Receive FIFO Data Register HL (FRDRHL)

Address(es): SCI0.FRDRHL 4007 0010h, SCI1.FRDRHL 4007 0030h, SCI2.FRDRHL 4007 0050h, SCI3.FRDRHL 4007 0070h, SCI4.FRDRHL 4007 0090h, SCI9.FRDRHL 4007 0130h

SCIn.FRDRHL

SCIn.FRDRH | SCIn.FRDRL

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	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	Bit name	Description	R/W
b8 to b0	RDAT[8:0]	Serial Receive Data	Received serial data, valid only in asynchronous mode (including multi-processor) or clock synchronous mode, with FIFO selected	R
b9	MPB	Multi-Processor Bit Flag	Multi-processor bit corresponding to serial receive data (RDAT[8:0]): 0: Data transmission cycle 1: ID transmission cycle. MPB is valid only in asynchronous mode with SMR.MP = 1 and FIFO selected.	R
b10	DR	Receive Data Ready Flag	0: Receiving is in progress, or no received data remains in FRDRH and FRDRL after normal receive completes 1: Next receive data is not received for a period after normal receive completes.	R*1
b11	PER	Parity Error Flag	0: No parity error occurred in the first data of FRDRH and FRDRL 1: A parity error occurred in the first data of FRDRH and FRDRL.	R
b12	FER	Framing Error Flag	0: No framing error occurred in the first data of FRDRH and FRDRL 1: A framing error occurred in the first data of FRDRH and FRDRL.	R
b13	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error occurred.	R*1
b14	RDF	Receive FIFO Data Full Flag	0: The amount of receive data written in FRDRH and FRDRL is below 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
b15	—	Reserved	This bit is read as 0.	R

Note 1. If this flag is read, it is same as a 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 FRDRL and FRDRH. FRDRH and FRDRL constitute a 16-stage FIFO register that stores serial receive data and related status information. This register is valid only 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 of received data, subsequent serial receive data is lost. The CPU can read from FRDRH and FRDRL but cannot write to them.

Reading 1 from the RDF, ORER, or DR flags of the FRDRH register is the same as reading those bits from 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 the order from FRDRH to FRDRL. FRDRHL can be accessed in 16-bit units.

29.2.5 Transmit Data Register (TDR)

Address(es): SCI0.TDR 4007 0003h, SCI1.TDR 4007 0023h, SCI2.TDR 4007 0043h, SCI3.TDR 4007 0063h, SCI4.TDR 4007 0083h, SCI9.TDR 4007 0123h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

TDR is an 8-bit register that stores transmit data.

When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission.

The double-buffered structure of TDR and TSR enable 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 TSR 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).

29.2.6 Transmit 9-Bit Data Register (TDRHL)

Address(es): SCI0.TDRHL 4007 000Eh, SCI1.TDRHL 4007 002Eh, SCI2.TDRHL 4007 004Eh, SCI3.TDRHL 4007 006Eh, SCI4.TDRHL 4007 008Eh, SCI9.TDRHL 4007 012Eh

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

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. For example, access to TDRHL affects TDR. Access to the TDRHL register is prohibited if 7-bit or 8-bit data length is selected.

When empty space is detected in TSR, the transmit data stored in TDRHL is transferred to TSR and transmission is started.

TSR and TDRHL form 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 TSR.

The CPU can read and write to TDRHL. Bits [9] to [15] in TDRHL are fixed to 1. These bits are read as 1. The write value should be 1.

Write transmit data to TDRHL only once when a transmit data empty interrupt (SCIn_TXI) request is issued.

29.2.7 Transmit FIFO Data Register H, L, HL (FTDRH, FTDRL, FTDRHL)

Transmit FIFO Data Register H (FTDRH)

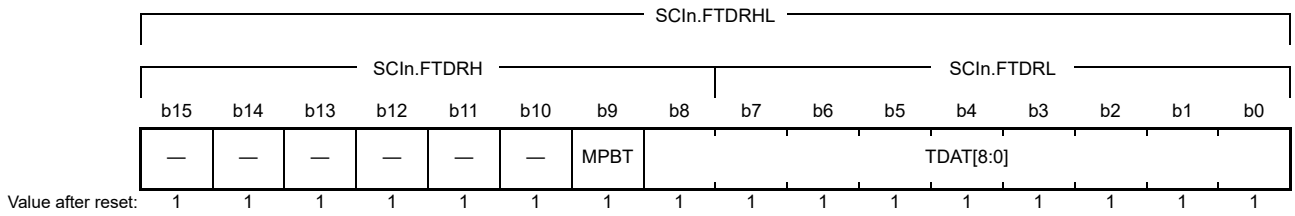
Address(es): SCI0.FTDRH 4007 000Eh, SCI1.FTDRH 4007 002Eh, SCI2.FTDRH 4007 004Eh, SCI3.FTDRH 4007 006Eh, SCI4.FTDRH 4007 008Eh, SCI9.FTDRH 4007 012Eh

Transmit FIFO Data Register L (FTDRL)

Address(es): SCI0.FTDRL 4007 000Fh, SCI1.FTDRL 4007 002Fh, SCI2.FTDRL 4007 004Fh, SCI3.FTDRL 4007 006Fh, SCI4.FTDRL 4007 008Fh, SCI9.FTDRL 4007 012Fh

Transmit FIFO Data Register HL (FTDRHL)

Address(es): SCI0.FTDRHL 4007 000Eh, SCI1.FTDRHL 4007 002Eh, SCI2.FTDRHL 4007 004Eh, SCI3.FTDRHL 4007 006Eh, SCI4.FTDRHL 4007 008Eh, SCI9.FTDRHL 4007 012Eh



Bit	Symbol	Bit name	Description	R/W
b8 to b0	TDAT[8:0]	Serial Transmit Data	Serial write data, valid only in asynchronous mode, including multi-processor or clock synchronous mode, with FIFO selected	W
b9	MPBT	Multi-Processor Transfer Bit Flag	Value of the multi-processor bit in the transmission frame: 0: Data transmission cycle 1: ID transmission cycle. MPBT is valid only in asynchronous mode with SMR.MP = 1 and FIFO selected.	W
b15 to b10	—	Reserved	The write value should be 1	W

FTDRHL is a 16-bit register that consists of FTDRH and FTDRL.

FTDRH and FTDRL constitute a 16-stage FIFO register that stores data for serial transmission and the multi-processor transfer bit. These registers are valid only in asynchronous mode, including multi-processor mode or clock synchronous mode.

When the SCI detects that TSR is empty, it transmits data written in FTDRH and FTDRL into TSR 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 FTDRH and FTDRL but cannot read them.

When writing to both the FTDRH and FTDRL registers, write in the order from FTDRH to FTDRL.

MPBT bit (Multi-Processor Transfer Bit Flag)

Selects the multi-processor bit of the transmit frame.

When FCR.FM = 1, SSR.MPBT is not valid.

29.2.8 Transmit Shift Register (TSR)

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 TSR.

29.2.9 Serial Mode Register (SMR) for Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI0.SMR 4007 0000h, SCI1.SMR 4007 0020h, SCI2.SMR 4007 0040h, SCI3.SMR 4007 0060h, SCI4.SMR 4007 0080h, SCI9.SMR 4007 0120h

	b7	b6	b5	b4	b3	b2	b1	b0
	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 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
b2	MP	Multi-Processor Mode	Valid only in asynchronous mode: 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled.	R/W*4
b3	STOP	Stop Bit Length	Valid only in asynchronous mode: 0: 1 stop bit 1: 2 stop bits.	R/W*4
b4	PM	Parity Mode	Valid only when the PE bit is 1: 0: Selects even parity 1: Selects odd parity.	R/W*4
b5	PE	Parity Enable	Valid only in asynchronous mode: • When transmitting: 0: Parity bit addition is not performed 1: The parity bit is added. • When receiving: 0: Parity bit checking is not performed 1: The parity bit is checked.	R/W*4
b6	CHR	Character Length	Selects character length in combination with the CHR1 bit in SCMR: CHR1 CHR 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length.*3 CHR is valid only in asynchronous mode.*2	R/W*4
b7	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 BRR, see section 29.2.17, Bit Rate Register (BRR).
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 TDR is not transmitted.
Note 4. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

SMR sets the communication format and clock source for the on-chip baud rate generator.

CKS[1:0] bits (Clock Select)

These 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 29.2.17, Bit Rate Register (BRR).

MP bit (Multi-Processor Mode)

Disables or enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

STOP bit (Stop Bit Length)

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)

Selects the parity mode (even or odd) for transmission and reception.

The setting of the PM bit is invalid in multi-processor mode.

PE bit (Parity Enable)

When this bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception.

Regardless of the setting of the PE bit, the parity bit is not added or checked in multi-processor format.

CHR bit (Character Length)

Selects the data length for transmission and reception in combination with the CHR1 bit in SCMR.

In modes other than asynchronous mode, a fixed data length of 8 bits is used.

CM bit (Communication Mode)

Selects the communication mode:

- Asynchronous mode or simple IIC mode
- Clock synchronous mode or simple SPI mode

29.2.10 Serial Mode Register for Smart Card Interface Mode (SMR_SMCI) (SCMR.SMIF = 1)

Address(es): SCI0.SMR_SMCI 4007 0000h, SCI1.SMR_SMCI 4007 0020h, SCI2.SMR_SMCI 4007 0040h, SCI3.SMR_SMCI 4007 0060h, SCI4.SMR_SMCI 4007 0080h, SCI9.SMR_SMCI 4007 0120h

	b7	b6	b5	b4	b3	b2	b1	b0
	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 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
b3, b2	BCP[1:0]	Base Clock Pulse	Selects the number of base clock cycles in combination with the BCP2 bit in SCMR. Table 29.3 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*2
b4	PM	Parity Mode	Valid only when the PE bit is 1: 0: Selects even parity 1: Selects odd parity.	R/W*2
b5	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
b6	BLK	Block Transfer Mode	0: Non-block transfer mode operation 1: Block transfer mode operation.	R/W*2
b7	GM	GSM Mode	0: Non-GSM mode operation 1: GSM mode operation.	R/W*2

Note 1. n is the decimal notation of the value of n in BRR, see section 29.2.17, Bit Rate Register (BRR).
Note 2. Writable only when TE in SCR_SMCI = 0 and RE in SCR_SMCI = 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)

These 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 29.2.17, Bit Rate Register (BRR).

BCP[1:0] bits (Base Clock Pulse)

These 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 BCP2 bit in SCMR.

For details, see section 29.6.4, Receive Data Sampling Timing and Reception Margin.

Table 29.3 Combinations of SCMR.BCP2 bit 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
0	0	0	93 clock cycles (S = 93)*1
0	0	1	128 clock cycles (S = 128)*1
0	1	0	186 clock cycles (S = 186)*1
0	1	1	512 clock cycles (S = 512)*1
1	0	0	32 clock cycles (S = 32)*1 (Initial Value)
1	0	1	64 clock cycles (S = 64)*1
1	1	0	372 clock cycles (S = 372)*1
1	1	1	256 clock cycles (S = 256)*1

Note 1. S is the value of S in BRR (see section 29.2.17, Bit Rate Register (BRR)).

PM bit (Parity Mode)

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 29.6.2, 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 this bit to 1 enables block transfer mode operation.

For details, see section 29.6.3, Block Transfer Mode.

GM bit (GSM Mode)

Setting this bit to 1 enables GSM mode operation.

In GSM mode, the SSR_SMCI.TEND flag set timing is moved forward to 11.0 ETU (elementary time unit = 1-bit transfer time) from the start and the clock output control function is enabled. For details, see section 29.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 29.6.8, Clock Output Control.

29.2.11 Serial Control Register (SCR) for Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI0.SCR 4007 0002h, SCI1.SCR 4007 0022h, SCI2.SCR 4007 0042h, SCI3.SCR 4007 0062h, SCI4.SCR 4007 0082h, SCI9.SCR 4007 0122h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	Asynchronous mode: b1 b0 0 0: On-chip baud rate generator The SCKn pin is available for use as an I/O port according to the I/O port settings. 0 1: On-chip baud rate generator A clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock A clock with a frequency 16 times the bit rate should be input from the SCKn pin when SEMR.ABCS bit is 0. Input a clock signal with a frequency eight times the bit rate when the SEMR.ABCS bit is 1. Clock synchronous mode: b1 b0 0 x: Internal clock The SCKn pin functions as the clock output pin. 1 x: External clock The SCKn pin functions as the clock input pin.	R/W*1
b2	TEIE	Transmit End Interrupt Enable	0: SCIn_TEI interrupt request is disabled 1: SCIn_TEI interrupt request is enabled.	R/W
b3	MPIE	Multi-Processor Interrupt Enable	Valid in asynchronous mode when SMR.MP = 1: 0: Non-multi-processor 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 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 non-multi-processor reception is resumed.	R/W*3
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled.	R/W*2
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled.	R/W*2
b6	RIE	Receive Interrupt Enable	0: SCIn_RXI and SCIn_ERI interrupt requests are disabled 1: SCIn_RXI and SCIn_ERI interrupt requests are enabled.	R/W
b7	TIE	Transmit Interrupt Enable	0: SCIn_TXI interrupt request is disabled 1: SCIn_TXI interrupt request is enabled.	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0, and 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 in this register during 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 read-modify-write when using a bit manipulation instruction.

SCR sets control and the clock source selection for transmission and reception.

CKE[1:0] bits (Clock Enable)

These bits select the clock source and SCKn pin function.

TEIE bit (Transmit End Interrupt Enable)

The TEIE bit enables or disables an SCIn_TEI interrupt request.

Setting the TEIE bit to 0 disables 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 (STI). 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, RDF, ORER, and FER in SSR/SSR_FIFO to 1 is disabled. When data with the multi-processor bit set to 1 is received, the MPIE is automatically set to 0, and non-multi-processor reception resumes. For details, see section 29.4, Multi-Processor Communication Function.

When the receive data includes the MPB is set to 0, the receive data is not transferred from the RSR to the RDR, a receive error is not detected, and setting the flags ORER and FER to 1 is disabled.

When the receive data includes the MPB bit is set to 1, the MPIE bit is automatically set to 0, the SCIn_RXI and SCIn_ERI interrupt requests are enabled (if the RIE bit in SCR is set to 1), and the setting of the ORER and FER flags to 1 is enabled.

MPIE should be set to 0 if the multi-processor communications function is not used.

RE bit (Receive Enable)

Enables or disables serial reception.

When this 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. SMR must be set prior to setting the RE bit to 1 to designate the reception format.

When non-FIFO operation is selected and reception is halted by setting the RE bit to 0, the RDRF, ORER, FER, and PER flags in SSR are not affected and the previous value is saved.

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 value is saved.

TE bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission starts by writing transmit data to TDR. SMR must be set prior to setting the TE bit to 1 to designate the transmission format.

RIE bit (Receive Interrupt Enable)

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.

To cancel an SCIn_ERI interrupt request, read 1 from the ORER, FER, or PER flag in SSR/SSR_FIFO, then set the flag to 0, or set the RIE bit to 0.

TIE bit (Transmit Interrupt Enable)

Enables or disables SCIn_TXI interrupt request.

Setting the TIE bit to 0 disables an SCIn_TXI interrupt request. TIE must be set to 1 when the TE bit is 1. The SCIn_TXI interrupt occurs after the TE and TIE bits are set to 1 simultaneously before the transfer starts.

29.2.12 Serial Control Register for Smart Card Interface Mode (SCR_SMCI) (SCMR.SMIF = 1)

Address(es): SCI0.SCR_SMCI 4007 0002h, SCI1.SCR_SMCI 4007 0022h, SCI2.SCR_SMCI 4007 0042h, SCI3.SCR_SMCI 4007 0062h, SCI4.SCR_SMCI 4007 0082h, SCI9.SCR_SMCI 4007 0122h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	• When GM in SMR_SMCI = 0: b1 b0 0 0: Output disabled The SCKn pin is available for use as an I/O port according to the I/O port settings. 0 1: Clock output 1 x: (Setting prohibited). • When GM in SMR_SMCI = 1: b1 b0 0 0: Output fixed low x 1: Clock output 1 0: Output fixed high.	R/W*1
b2	TEIE	Transmit End Interrupt Enable	This bit should be 0 in smart card interface mode	R/W
b3	MPIE	Multi-Processor Interrupt Enable	This bit should be 0 in smart card interface mode	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled.	R/W*2
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled.	R/W*2
b6	RIE	Receive Interrupt Enable	0: SCIn_RXI and SCIn_ERI interrupt requests are disabled 1: SCIn_RXI and SCIn_ERI interrupt requests are enabled.	R/W
b7	TIE	Transmit Interrupt Enable	0: SCIn_TXI interrupt request is disabled 1: SCIn_TXI interrupt request is enabled.	R/W

x: Don't care

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.

SCR_SMCI sets transmission control, interrupt control, and reception and clock source selection for transmission and reception.

For details on interrupt requests, see section 29.10, Interrupt Sources.

CKE[1:0] bits (Clock Enable)

These bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, see section 29.6.8, Clock Output Control.

TEIE bit (Transmit End Interrupt Enable)

This bit should be 0 in smart card interface mode.

RE bit (Receive Enable)

Enables or disables serial reception.

When this bit is set to 1, serial reception starts by detecting the start bit.

Note: SMR_SMCI should be set prior to setting the RE bit to 1 to designate the reception format.

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 value is saved.

TE bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission starts by writing transmit data to TDR.

Note: SMR_SMCI should be set prior to setting the TE bit to 1 to designate the transmission format.

RIE bit (Receive Interrupt Enable)

Enables or disables SCIn_RXI and SCIn_ERI interrupt requests.

An 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_SMCI and then setting the flag to 0, or setting the RIE bit to 0.

TIE bit (Transmit Interrupt Enable)

Enables or disables SCIn_TXI interrupt request.

Setting the TIE bit to 0 disables an SCIn_TXI interrupt request. Set TIE to 1 when the TE bit is 1. The SCIn_TXI interrupt occurs after the TE and TIE bits are set to 1 simultaneously, before transfer starts.

29.2.13 Serial Status Register (SSR) for Non-Smart Card Interface and Non-FIFO Mode (SCMR.SMIF = 0 and FCR.FM = 0)

Address(es): SCI0.SSR 4007 0004h, SCI1.SSR 4007 0024h, SCI2.SSR 4007 0044h, SCI3.SSR 4007 0064h, SCI4.SSR 4007 0084h, SCI9.SSR 4007 0124h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	Sets the multi-processor bit for the transmission frame: 0: Data transmission cycle 1: ID transmission cycle.	R/W
b1	MPB	Multi-Processor	Value of the multi-processor bit in the reception frame: 0: Data transmission cycle 1: ID transmission cycle.	R
b2	TEND	Transmit End Flag	0: A character is transmitted 1: Character transfer is complete.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error occurred.	R/(W)*1
b4	FER	Framing Error Flag	0: No framing error occurred 1: A framing error occurred.	R/(W)*1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error occurred.	R/(W)*1
b6	RDRF	Receive Data Full Flag	0: No received data is in RDR register 1: Received data is in RDR register.	R/(W)*1
b7	TDRE	Transmit Data Empty Flag	0: Transmit data is in TDR register 1: No transmit data is in TDR register.	R/(W)*1

Note 1. Only 0 can be written to clear the flag after reading 1.

The SSR register provides the SCI status flag and transmission/reception multi-processor bits.

MPBT bit (Multi-Processor Bit Transfer)

Selects the multi-processor bit of the transmit frame.

MPB bit (Multi-Processor)

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)

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 at the time of transmission of the tail-end bit of a character being transmitted.

[Clearing conditions]

- When transmit data is written to the TDR register when the SCR.TE bit is 1.
- When 0 is written to TDRE after reading TDRE = 1 when the SCR.TE bit is 1.

PER flag (Parity Error Flag)

Indicates that a parity error occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception in asynchronous mode and the address match function is disabled (DCCR.DCME = 0).

  Although receive data when the parity error occurs is transferred to RDR, no SCIn_RXI interrupt request occurs. When 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 PER, read the PER bit to check that it was actually set to 0).

When the RE bit in SCR is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

FER flag (Framing Error Flag)

Indicates that a framing error occurs during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When 0 is sampled as the stop bit during reception in asynchronous mode and the address match function is disabled (DCCR.DCME = 0).

  In 2-stop-bit mode, only the first stop bit is checked if it is 1 but the second stop bit is not checked. Although receive data when the framing error occurs is transferred to RDR, no SCIn_RXI interrupt request occurs. In addition, when the FER flag is set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to FER after reading FER = 1 (after writing 0 to FER, read the FER bit to check that it was actually set to 0).

When the RE bit in SCR is set to 0, the FER flag is not affected and retains its previous value.

ORER flag (Overrun Error Flag)

Indicates that an overrun error occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data that does not have a parity error, and a frame error is read from RDR.

  In RDR, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, reception data is not forwarded to RDR. 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 ORER, read the ORER bit to check that it was actually set to 0).

When the RE bit in SCR is set to 0, the ORER flag is not affected and retains its previous value.

RDRF flag (Receive Data Full Flag)

Indicates the presence of receive data in RDR.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing conditions]

- When it is set to 0 after 1 is read
- When data is read from the RDR register.

Note: Do not clear the RDRF flag by accessing RDRF bit in SSR register unless communication is aborted.

TDRE flag (Transmit Data Empty 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 it is set to 0 after 1 is read
- When the SCR.TE bit is 1, data is forwarded to the TDR register.

Note: Do not clear the TDRE flag by accessing TDRE bit in SSR register unless communication is aborted.

29.2.14 Serial Status Register for Non-Smart Card Interface and FIFO Mode (SSR_FIFO) (SCMR.SMIF = 0 and FCR.FM = 1)

Address(es): SCI0.SSR_FIFO 4007 0004h, SCI1.SSR_FIFO 4007 0024h, SCI2.SSR_FIFO 4007 0044h, SCI3.SSR_FIFO 4007 0064h, SCI4.SSR_FIFO 4007 0084h, SCI9.SSR_FIFO 4007 0124h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDFE	RDF	ORER	FER	PER	TEND	—	DR
Value after reset:	1	0	0	0	0	0	X	0

Bit	Symbol	Bit name	Description	R/W
b0	DR	Receive Data Ready Flag	0: Receiving is in progress, or no received data remains in FRDRHL after successfully completed reception (receive FIFO is empty) 1: Next receive data is not received for a period after normal receiving is complete, and when the amount of data stored in the FIFO is equal to or less than the receive triggering number.	R/(W)*1
b1	—	Reserved	The read value is undefined. The write value should be 1.	R/W
b2	TEND	Transmit End Flag	0: A character is transmitted 1: Character transfer is complete.	R/(W)*1
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error occurred.	R/(W)*1
b4	FER	Framing Error Flag	0: No framing error occurred 1: A framing error occurred.	R/(W)*1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error occurred.	R/(W)*1
b6	RDF	Receive FIFO Data Full Flag	0: The amount of receive data written in FRDRHL is below 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
b7	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 SCI with FIFO mode status flags.

DR flag (Receive Data Ready 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 from the last stop bit in asynchronous mode. This flag is valid only in asynchronous mode, including multi-processor mode, and when the FIFO is selected.

In clock synchronous mode, this flag is not set to 1.

[Setting condition]

- DR is set to 1 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]

- DR is set to 0 when 1 is read from DR and 0 is written after all received data are read.
- When the FCR.FM bit is switched from 0 to 1.

Note 1. This is equivalent to one and a half (1.5) frames in the 8-bit format with one stop bit (ETU).

TEND flag (Transmit End Flag)

Indicates that FTDRHL does not contain valid data when transmitting the last bit of a serial character, therefore the transmission is halted.

[Setting condition]

- TEND is set to 1 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 when 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 switched from 0 to 1.

PER flag (Parity Error 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]

- PER is set to 1 when a data is received and a parity error is detected and 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 when receive data is stored to the FRDRHL register even when a parity error occurs during reception.

When the SCR.RE bit is cleared, the PER flag is not affected and the previous state is kept.

FER flag (Framing Error 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]

- FER is set to 1 when 0 is sampled as the stop bit during reception and 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 when receive data is stored to the FRDRHL register even when a framing error occurs during reception.

When the SCR.RE bit is cleared, the FER flag is not affected and the previous state is kept.

ORER flag (Overrun Error Flag)

Indicates that receive operation abnormally stops due to occurrence of an overrun error.

[Setting condition]

- ORER is set to 1 when the next serial reception completes while the receive FIFO is full of 16-byte receive data.

[Clearing condition]

- ORER is set to 0 when 0 is written after 1 is read from ORER.

Clearing the SCR.RE bit to 0 does not affect the ORER flag, which retains its previous state.

RDF flag (Receive FIFO Data Full Flag)

Indicates that receive data is transferred to FRDRHL, and the amount of data in FRDRHL is equal to or exceeds the specified receive triggering number. However, 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]

- RDF is set to 1 when the amount of receive data is equal to or greater than the specified receive triggering number that is stored in FRDRHL*1 and the FIFO is not empty.

[Clearing conditions]

- RDF is set to 0 after 1 is read from RDF.
- RDF is set to 0 when FRDRHL is read by the DMAC or the DTC but only when the block transfer is the last transmission.
- When the setting condition and clearing condition occur at the same time, the RDF flag is 0. After that, when the amount of data stored in the FRDRHL register is the same or greater than the RTRG value, RDF is set to 1 after 1 PCLK.

Note 1. Because the 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.

Note: Do not clear the RDF flags by accessing the RDF bit in the SSR register before reading receive data unless communication is aborted.

TDFE bit (Transmit FIFO Data Empty Flag)

Indicates that data is transferred from FTDRHL into TSR, the amount of data in FTDRHL is below the specified transmit triggering number, and writing of transmit data to FTDRHL is enabled.

[Setting conditions] TDFE is set to 1 when the TE bit in SCR is 0.

- TDFE is set to 1 when the amount of transmit data written in FTDRHL is equal to or less than the specified transmit triggering number*1.

[Clearing conditions]

- TDFE is set to 0 when writing to FTDRHL is executed on the last transmission while the DTC or DMAC is activated.
- TDFE is set to 0 when 0 is written to the TDFE bit after reading TDFE = 1.
  When the setting condition and the clearing condition occur at the same time, the TDFE flag is cleared. After that, when the amount of data stored in 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, the maximum amount of data that can be written when the TDFE flag is indicated in "16 - FDR.T[4:0]". If more data is written, data is discarded.

Note: Do not clear the TDFE flags by accessing the TDFE bit in the SSR register before writing transmit data unless communication is aborted.

29.2.15 Serial Status Register for Smart Card Interface Mode (SSR_SMCI) (SCMR.SMIF = 1)

Address(es): SCI0.SSR_SMCI 4007 0004h, SCI1.SSR_SMCI 4007 0024h, SCI2.SSR_SMCI 4007 0044h, SCI3.SSR_SMCI 4007 0064h, SCI4.SSR_SMCI 4007 0084h, SCI9.SSR_SMCI 4007 0124h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	This bit should be 0 in smart card interface mode	R/W
b1	MPB	Multi-Processor	This bit should be 0 in smart card interface mode	R
b2	TEND	Transmit End Flag	0: A character is being transmitted 1: Character transfer is complete.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error occurred.	R/(W)*1
b4	ERS	Error Signal Status Flag	0: Low error signal is not sampled 1: Low error signal is sampled.	R/(W)*1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error occurred.	R/(W)*1
b6	RDRF	Receive Data Full Flag	0: No received data in RDR 1: Received data in RDR.	R/(W)*1
b7	TDRE	Transmit Data Empty Flag	0: Transmit data in TDR 1: No transmit data in TDR.	R/(W)*1

Note 1. Only 0 can be written to clear the flag after reading 1.

The SSR_SMCI register provides SCI with smart card interface mode status flags.

TEND flag (Transmit End Flag)

With no error signal from the receiving side, this bit 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 changes 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 etu after the start of transmission
  - When SMR_SMCI.GM = 0 and SMR_SMCI.BLK = 1, 11.5 etu after the start of transmission
  - When SMR_SMCI.GM = 1 and SMR_SMCI.BLK = 0, 11.0 etu after the start of transmission
  - When SMR_SMCI.GM = 1 and SMR_SMCI.BLK = 1, 11.0 etu 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 and the SCR_SMCI.TE bit is 1.

PER flag (Parity Error 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 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 RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1. After writing 0 to the PER bit, read the bit 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 keeps 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)

Indicates that an overrun error occurred during reception and the reception ended abnormally.

[Setting condition]

- When the next data is received before receive data is read from RDR that does not have a parity error. In RDR, the data received before an overrun error occurred is saved, but data received after the error is lost. When the ORER flag is set to 1, received 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 bit, read the bit 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 keeps its previous value.

RDRF flag (Receive Data Full 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 1 is read
- When data is read from the RDR register.

Note: Do not clear the RDRF flags by accessing the RDRF bit in the SSR register unless communication is aborted.

TDRE flag (Transmit Data Empty 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 1 is read
- When the SCR_SMCI.TE bit is 1 and it forwards data to the TDR register.

Note: Do not clear the TDRE flags by accessing the TDRE bit in the SSR register unless communication is aborted.

29.2.16 Smart Card Mode Register (SCMR)

Address(es): SCI0.SCMR 4007 0006h, SCI1.SCMR 4007 0026h, SCI2.SCMR 4007 0046h, SCI3.SCMR 4007 0066h, SCI4.SCMR 4007 0086h, SCI9.SCMR 4007 0126h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCP2	—	—	CHR1	SDIR	SINV	—	SMIF
Value after reset:	1	1	1	1	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b0	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
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b2	SINV	Transmitted/Received Data Invert	0: TDR contents are transmitted as they are. Receive data is stored as received in the RDR. 1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in the RDR. This bit can be used in the following modes: • Smart card interface mode • Asynchronous mode (multi-processor mode) • Clock synchronous mode • Simple SPI mode. Set this bit to 0 for operation in simple IIC mode.	R/W*1
b3	SDIR	Transmitted/Received Data Transfer Direction	0: Transfer with LSB first. 1: Transfer with MSB first. This bit can be used in the following modes: • Smart card interface mode • Asynchronous mode (multi-processor mode) • Clock synchronous mode • Simple SPI mode. Set this bit to 1 for operation in simple IIC mode.	R/W*1
b4	CHR1	Character Length 1	Only valid in asynchronous mode)*2 Selects the character length in combination with the CHR bit in SMR: CHR1 CHR 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length.*3	R/W*1
b6, b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	BCP2	Base Clock Pulse 2	Selects the number of base clock cycles in combination with the SMR_SMCI.BCP[1:0] bits. Table 29.4 lists the combinations of the SCMR.BCP2 bit and SMR_SMCI.BCP[1:0] bits.	R/W*1

Note 1. Writable only when TE in SCR/SCR_SMCI = 0 and RE in SCR/SCR_SMCI = 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 should be selected and the value of 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)

Inverts the transmit/receive data logic level. This bit 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)

Selects the data length of transmit/receive data in combination with the CHR bit in SMR.

A fixed data length of 8 bits is used in modes other than asynchronous mode.

BCP2 bit (Base Clock Pulse 2)

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 29.4 Combinations of the SCMR.BCP2 bit 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
0	0	0	93 clock cycles (S = 93)*1
0	0	1	128 clock cycles (S = 128)*1
0	1	0	186 clock cycles (S = 186)*1
0	1	1	512 clock cycles (S = 512)*1
1	0	0	32 clock cycles (S = 32)*1 (Initial Value)
1	0	1	64 clock cycles (S = 64)*1
1	1	0	372 clock cycles (S = 372)*1
1	1	1	256 clock cycles (S = 256)*1

Note 1. S is the value of S in the Bit Rate Register (BRR), see section 29.2.17, Bit Rate Register (BRR).

29.2.17 Bit Rate Register (BRR)

Address(es): SCI0.BRR 4007 0001h, SCI1.BRR 4007 0021h, SCI2.BRR 4007 0041h, SCI3.BRR 4007 0061h, SCI4.BRR 4007 0081h, SCI9.BRR 4007 0121h

	b7	b6	b5	b4	b3	b2	b1	b0
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. Table 29.5 shows the relationship between the setting (N) in the BRR and the bit rate (B) for asynchronous mode, multi-processor transfer, clock synchronous mode, smart card interface mode, simple SPI mode, and simple IIC mode.

The initial value of BRR is FFh. BRR 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 29.5 Relationship between N setting in BRR and bit rate B

Mode	SEMR settings			BRR setting	Error
	BGDM bit	ABCS bit	ABCSE bit		
Asynchro-nous, multi-processor transfer	0	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \} \times 100$
	1	0	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1 \} \times 100$
	Don't care	Don't care	1	$N = \frac{PCLK \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$Error\ (\%) = \{ \frac{PCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (N+1)} - 1 \} \times 100$
Clock synchronous, simple SPI				$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface				$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error\ (\%) = \{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \} \times 100$
Simple IIC*1				$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	

B: Bit rate (bps)
N: BRR setting for on-chip baud rate generator ($0 \leq N \leq 255$)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR/SMR_SMCI and SCMR registers as listed in Table 29.7 and Table 29.8.
Note 1. Adjust the bit rate so that the high and low level SCLn output widths in simple IIC mode satisfy the IIC standard.

Table 29.6 Calculating widths at high and low level for SCL

Mode	SCL	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 29.7 Clock source settings

SMR or SMR_SMCI.CKS[1:0] bit setting		
CKS[1:0] bits	Clock source	n
0 0	PCLK clock	0
0 1	PCLK/4 clock	1
1 0	PCLK/16 clock	2
1 1	PCLK/64 clock	3

Table 29.8 Base clock settings in smart card interface mode

SCMR.BCP2 bit setting	SMR_SMCI.BCP[1:0] bit setting		
BCP2 bit	BCP[1:0] bits	Base clock cycles for 1-bit period	S
0	0 0	93 clock cycles	93
0	0 1	128 clock cycles	128
0	1 0	186 clock cycles	186
0	1 1	512 clock cycles	512
1	0 0	32 clock cycles	32
1	0 1	64 clock cycles	64
1	1 0	372 clock cycles	372
1	1 1	256 clock cycles	256

Table 29.9 and Table 29.10 list examples of BRR (N) settings in asynchronous mode. Table 29.11 lists the maximum bit rate selectable for each operating frequency. Table 29.14 lists examples of BRR (N) settings in smart card interface mode. Table 29.17 lists examples of BRR (N) settings in simple IIC 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 29.6.4, Receive Data Sampling Timing and Reception Margin. Table 29.12 and Table 29.14 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 that as listed in Table 29.16. When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 29.9 Examples of BRR settings for different bit rates in asynchronous mode (1)

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
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

Bit Rate (bps)	Operating frequency PCLK (MHz)														
	14			16			17.2032			18			19.6608		
	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
150	2	181	0.16	2	207	0.16	2	223	0.00	2	233	0.16	2	255	0.00
300	2	90	0.16	2	103	0.16	2	111	0.00	2	116	0.16	2	127	0.00
600	1	181	0.16	1	207	0.16	1	223	0.00	1	233	0.16	1	255	0.00
1200	1	90	0.16	1	103	0.16	1	111	0.00	1	116	0.16	1	127	0.00
2400	0	181	0.16	0	207	0.16	0	223	0.00	0	233	0.16	0	255	0.00
4800	0	90	0.16	0	103	0.16	0	111	0.00	0	116	0.16	0	127	0.00

Bit Rate (bps)	Operating frequency PCLK (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	45	-0.93	0	51	0.16	0	55	0.00	0	58	-0.69	0	63	0.00
19200	0	22	-0.93	0	25	0.16	0	27	0.00	0	28	1.02	0	31	0.00
31250	0	13	0.00	0	15	0.00	0	16	1.20	0	17	0.00	0	19	-1.70
38400	-	-	-	0	12	0.16	0	13	0.00	0	14	-2.34	0	15	0.00

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 = 1 and BGDM = 1, the bit rate increases four times.

Table 29.10 Examples of BRR settings for different bit rates in asynchronous mode (2)

Bit Rate (bps)	Operating frequency PCLK (MHz)														
	20			25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	-0.25	3	110	-0.02	3	132	0.13	3	145	0.33	3	177	-0.25
150	3	64	0.16	3	80	0.47	3	97	-0.35	3	106	0.39	3	129	0.16
300	2	129	0.16	2	162	-0.15	2	194	0.16	2	214	-0.07	3	64	0.16
600	2	64	0.16	2	80	0.47	2	97	-0.35	2	106	0.39	2	129	0.16
1200	1	129	0.16	1	162	-0.15	1	194	0.16	1	214	-0.07	2	64	0.16
2400	1	64	0.16	1	80	0.47	1	97	-0.35	1	106	0.39	1	129	0.16
4800	0	129	0.16	0	162	-0.15	0	194	0.16	0	214	-0.07	1	64	0.16
9600	0	64	0.16	0	80	0.47	0	97	-0.35	0	106	0.39	0	129	0.16
19200	0	32	-1.36	0	40	-0.76	0	48	-0.35	0	53	-0.54	0	64	0.16
31250	0	19	0.00	0	24	0.00	0	29	0.00	0	32	0.00	0	39	0.00
38400	0	15	1.73	0	19	1.73	0	23	1.73	0	26	-0.54	0	32	-1.36

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 = 1 and BGDM = 1, the bit rate increases four times.

Table 29.11 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	250,000	17.2032	0	0	0	0	0	537,600
		1	0	0	0	500,000			1	0	0	0	1,075,200
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1,000,000			1	0	0	0	2,150,400
	Don't care	Don't care	1	0	0	1,333,333		Don't care	Don't care	1	0	0	2,867,200
9.8304	0	0	0	0	0	307,200	18	0	0	0	0	0	562,500
		1	0	0	0	614,400			1	0	0	0	1,125,000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1,228,800			1	0	0	0	2,250,000
	Don't care	Don't care	1	0	0	1,638,400		Don't care	Don't care	1	0	0	3,000,000

Table 29.11 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	312,500	19.6608	0	0	0	0	0	614,400
		1	0	0	0	625,000			1	0	0	0	1,228,800
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1,250,000			1	0	0	0	2,457,600
	Don't care	Don't care	1	0	0	1,666,666		Don't care	Don't care	1	0	0	3,276,800
12	0	0	0	0	0	375,000	20	0	0	0	0	0	625,000
		1	0	0	0	750,000			1	0	0	0	1,250,000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1,500,000			1	0	0	0	2,500,000
	Don't care	Don't care	1	0	0	2,000,000		Don't care	Don't care	1	0	0	3,333,333
12.288	0	0	0	0	0	384,000	25	0	0	0	0	0	781,250
		1	0	0	0	768,000			1	0	0	0	1,562,500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1,536,000			1	0	0	0	3,125,000
	Don't care	Don't care	1	0	0	2,048,000		Don't care	Don't care	1	0	0	4,166,666
14	0	0	0	0	0	437,500	30	0	0	0	0	0	937,500
		1	0	0	0	875,000			1	0	0	0	1,875,000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1,750,000			1	0	0	0	3,750,000
	Don't care	Don't care	1	0	0	2,333,333		Don't care	Don't care	1	0	0	5,000,000
16	0	0	0	0	0	500,000	33	0	0	0	0	0	1,031,250
		1	0	0	0	1,000,000			1	0	0	0	2,062,500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2,000,000			1	0	0	0	4,125,000
	Don't care	Don't care	1	0	0	2,666,666		Don't care	Don't care	1	0	0	5,500,000
40	0	0	0	0	0	1,250,000							
		1	0	0	0	2,500,000							
	1	0	0	0	0								
		1	0	0	0	5,000,000							
	Don't care	Don't care	1	0	0	6,666,666							

Table 29.12 Maximum bit rate with external clock input in asynchronous mode (1 of 2)

		Maximum bit rate (bps)	
PCLK (MHz)	External input clock (MHz)	SEMR.ABCS bit = 0	SEMR.ABCS bit = 1
8	2.0000	125,000	250,000
9.8304	2.4576	153,600	307,200
10	2.5000	156,250	312,500
12	3.0000	187,500	375,000
12.288	3.0720	192,000	384,000

Table 29.12 Maximum bit rate with external clock input in asynchronous mode (2 of 2)

		Maximum bit rate (bps)	
PCLK (MHz)	External input clock (MHz)	SEMR.ABCS bit = 0	SEMR.ABCS bit = 1
14	3.5000	218,750	437,500
16	4.0000	250,000	500,000
17.2032	4.3008	268,800	537,600
18	4.5000	281,250	562,500
19.6608	4.9152	307,200	614,400
20	5.0000	312,500	625,000
25	6.2500	390,625	781,250
30	7.5000	468,750	937,500
33	8.2500	515,625	1,031,250
40	10.0000	625,000	1,250,000

Table 29.13 BRR settings for different bit rates in clock synchronous and simple SPI modes

	Operating frequency PCLK (MHz)															
	8		10		16		20		25		30		33		40	
Bit rate (bps)	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110																
250	3	124	—	—	3	249										
500	2	249	—	—	3	124	—	—			3	233				
1 k	2	124	—	—	2	249	—	—	3	97	3	116	3	128	3	155
2.5 k	1	199	1	249	2	99	2	124	2	155	2	187	2	205	2	249
5 k	1	99	1	124	1	199	1	249	2	77	2	93	2	102	2	124
10 k	0	199	0	249	1	99	1	124	1	155	1	187	1	205	1	249
25 k	0	79	0	99	0	159	0	199	0	249	1	74	1	82	1	99
50 k	0	39	0	49	0	79	0	99	0	124	0	149	0	164	1	49
100 k	0	19	0	24	0	39	0	49	0	62	0	74	0	82	0	99
250 k	0	7	0	9	0	15	0	19	0	24	0	29	0	32	0	39
500 k	0	3	0	4	0	7	0	9	—	—	0	14	—	—	0	19
1 M	0	1			0	3	0	4	—	—	—	—	—	—	0	9
2.5 M			0	0*1			0	1	—	—	0	2	—	—	0	3
5 M							0	0*1	—	—	—	—	—	—	0	1
7.5 M											0	0*1				

Space: Setting prohibited.

—: Can be set, but an error will occur.

Note 1. Continuous transmission or reception is impossible. After transmitting or receiving one frame of data, a 1-bit period elapses before starting transmitting or receiving the next frame of data. The output of the synchronization clock is stopped for a 1-bit period. For this reason, 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 29.14 Maximum bit rate with external clock input in clock synchronous and simple SPI modes (1 of 2)

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

Table 29.14 Maximum bit rate with external clock input in clock synchronous and simple SPI modes (2 of 2)

PCLK (MHz)	External input clock (MHz)	Maximum bit rate (Mbps)
16	2.6667	2.6666667
18	3.0000	3.0000000
20	3.3333	3.3333333
25	4.1667	4.1666667
30	5.0000	5.0000000
33	5.5000	5.5000000
40	6.6667	6.6666667

Table 29.15 BRR settings for different bit rates in smart card interface mode, n = 0, S = 372

	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

	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

	Operating frequency PCLK (MHz)											
	25.00			30.00			33.00			40.00		
Bit rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	3	-12.49	0	3	5.01	0	4	-7.59	0	5	-6.66

Table 29.16 Maximum bit rate for each operating frequency in smart card interface mode, S = 32

PCLK (MHz)	Maximum bit rate (bps)	n	N
10.00	156,250	0	0
10.7136	167,400	0	0
13.00	203,125	0	0
16.00	250,000	0	0
18.00	281,250	0	0
20.00	312,500	0	0
25.00	390,625	0	0
30.00	468,750	0	0
33.00	515,625	0	0
40.00	625,000	0	0

Table 29.17 BRR settings for different bit rates in simple IIC mode (1 of 2)

Bit rate (bps)	Operating frequency PCLK (MHz)														
	8			10			16			20			25		
	n	N	Error (%)	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	1	19	-2.3
25 k	0	9	0.0	0	12	-3.8	1	4	0.0	1	5	4.2	1	7	-2.3
50 k	0	4	0.0	0	5	4.2	1	2	-16.7	1	2	4.2	1	3	-2.3
100 k*1	0	2	-16.7	0	3	-21.9	0	4	0.0	0	6	-10.7	1	1	-2.3
250 k	0	0	0.0	0	0	25	0	1	0.0	0	2	-16.7	0	2	4.2

Table 29.17 BRR settings for different bit rates in simple IIC mode (2 of 2)

Bit rate (bps)	Operating frequency PCLK (MHz)														
	8			10			16			20			25		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
350 k										0	1	-10.7	0	1	11.6*2
400 k*1										0	1	-21.9	0	1	-2.3*2

Bit rate (bps)	Operating frequency PCLK (MHz)								
	30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	1	22	1.9	1	25	-0.8	0	124	0.0
25 k	1	8	4.2	1	9	3.1	0	49	0.0
50 k	1	4	-6.3	1	4	3.1	0	24	0.0
100 k*1	1	2	-21.9	1	2	-14.1	0	12	-3.9
250 k	0	3	-6.3	0	3	3.1	0	4	0.0
350 k	0	2	-10.7	0	2	-1.8	0	3	-10.7
400 k*1	0	2	-21.9	0	2	-14.1	0	3	-21.9

Note 1. The bit rate of 100 kbps and 400 kbps indicates the set value at which the error is on the negative side.
Note 2. The minimum value of low width is smaller than 1.3 µs which is the standard value of Fast mode.

Table 29.18 Minimum widths at high and low level for SCL at different bit rates in simple IIC mode

Bit rate (bps)	Operating frequency PCLK (MHz)											
	8			10			16			20		
	n	N	Min. widths at high/low level for SCL (µs)	n	N	Min. widths at high/low level for SCL (µs)	n	N	Min. widths at high/low level for SCL (µs)	n	N	Min. widths at high/low level for SCL (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	4.38/5.00	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

Bit rate (bps)	Operating frequency PCLK (MHz)											
	25			30			33			40		
	n	N	Min. widths at high/low level for SCL (µs)	n	N	Min. widths at high/low level for SCL (µs)	n	N	Min. widths at high/low level for SCL (µs)	n	N	Min. widths at high/low level for SCL (µs)
10 k	1	19	44.80/51.20	1	22	42.93/49.60	1	25	44.12/50.42	0	124	43.75/50.00
25 k	1	7	17.92/20.48	1	8	16.80/19.20	1	9	16.97/19.39	0	49	17.50/20.00
50 k	1	3	8.96/10.24	1	4	9.33/10.66	1	4	8.48/9.70	0	24	8.75/10.00
100 k	1	1	4.48/5.12	1	2	5.60/6.40	1	2	5.09/5.82	0	12	4.55/5.20
250 k	0	2	1.68/1.92	0	3	1.86/2.13	0	3	1.70/1.94	0	4	1.75/2.00
350 k	0	1	1.12/1.28*1	0	2	1.40/1.60	0	2	1.27/1.45	0	3	1.40/1.60
400 k	0	1	1.12/1.28*1	0	2	1.40/1.60	0	2	1.27 /1.45	0	3	1.40/1.60

Note 1. The minimum value of low width is smaller than 1.3 µs, which is the standard value of the Fast-mode. The setting values are the same as in Table 29.17, BRR settings for different bit rates in simple IIC mode.

29.2.18 Modulation Duty Register (MDDR)

Address(es): SCI0.MDDR 4007 0012h, SCI1.MDDR 4007 0032h, SCI2.MDDR 4007 0052h, SCI3.MDDR 4007 0072h, SCI4.MDDR 4007 0092h, SCI9.MDDR 4007 0132h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

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 according to the settings of MDDR (M/256). Table 29.19 lists the relationship between the MDDR setting (M) and the bit rate (B).

The initial value of MDDR is FFh. 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 29.19 Relationship between MDDR setting (M) and bit rate (B) when bit rate modulation function is used

Mode	SEMR settings			BRR setting	Error
	BGDM bit	ABCS bit	ABCSE bit		
Asynchronous, multi-processor transfer	0	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	Error (%) $= \{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1 \} \times 100$
	1	0	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times (256/M) \times B} - 1$	Error (%) $= \{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1 \} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times (256/M) \times B} - 1$	Error (%) $= \{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1 \} \times 100$
	Don't care	Don't care	1	$N = \frac{PCLK \times 10^6}{12 \times 2^{2n-1} \times (256/M) \times B} - 1$	Error (%) $= \{ \frac{PCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1 \} \times 100$
Clock synchronous, simple SPI*1				$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times (256/M) \times B} - 1$	
Smart card interface				$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times (256/M) \times B} - 1$	Error (%) $= \{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (256/M) \times (N+1)} - 1 \} \times 100$
Simple IIC*2				$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	

B: Bit rate (bps)
M: MDDR setting (128 ≤ MDDR ≤ 255)
N: BRR setting for baud rate generator (0 ≤ N ≤ 255)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR/SMR_SMCI and SCMR registers as listed in Table 29.7 and Table 29.8. See section 29.2.17, Bit Rate Register (BRR).

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 29.20 lists examples of N settings in BRR and M settings in MDDR in asynchronous mode.

Table 29.20 Examples of BRR and MDDR settings for different bit rates in asynchronous mode (1)

Bit rate (bps)	Operating frequency PCLK (MHz)														
	8					9.8304					16				
	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
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

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

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

Bit rate (bps)	Operating frequency PCLK (MHz)														
	19.6608					20					25				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	15	(256)*1	0	0.00	0	10	173	0	-0.01	0	11	151	0	0.00
57600	0	9	240	0	0.00	0	9	236	0	0.03	0	7	151	0	0.00
115200	0	4	240	0	0.00	0	4	236	0	0.03	0	3	151	0	0.00
230400	0	1	192	0	0.00	0	4	236	1	0.03	0	1	151	0	0.00
460800	0	0	192	0	0.00	0	0	189	0	0.14	0	0	151	0	0.00

Bit rate (bps)	Operating frequency PCLK (MHz)														
	30					33					40				
	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	36	194	1	0.01	0	14	143	0	0.01	0	21	173	0	-0.01
57600	0	10	173	0	-0.01	0	9	143	0	0.01	0	38	230	1	-0.01
115200	0	10	173	1	-0.01	0	4	143	0	0.01	0	9	236	0	0.03
230400	0	6	220	1	-0.09	0	4	143	1	0.01	0	4	236	0	0.03
460800	0	3	252	1	0.14	0	1	229	0	0.10	0	4	236	1	0.03

Note 1. In this example, the ABCS and ABCSE in SEMR are 0. SEMR.BRME = 0 (M = 256) disables the bit rate modulation function.

29.2.19 Serial Extended Mode Register (SEMR)

Address(es): SCI0.SEMR 4007 0007h, SCI1.SEMR 4007 0027h, SCI2.SEMR 4007 0047h, SCI3.SEMR 4007 0067h, SCI4.SEMR 4007 0087h, SCI9.SEMR 4007 0127h

	b7	b6	b5	b4	b3	b2	b1	b0
	RXDESEL	BGDM	NFEN	ABCS	ABCSE	BRME	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	BRME	Bit Rate Modulation Enable	0: Bit rate modulation function is disabled 1: Bit rate modulation function is enabled.	R/W*1
b3	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 is decided with combination of BGDM and ABCS in SEMR 1: Baud rate is 6 base clock cycles for 1-bit period.	R/W*1
b4	ABCS	Asynchronous Mode Base Clock Select	Valid only in asynchronous mode: 0: Selects 16 base clock cycles for 1-bit period 1: Selects 8 base clock cycles for 1-bit period.	R/W*1
b5	NFEN	Digital Noise Filter Function Enable	In asynchronous mode: 0: Noise cancellation function for the RXDn input signal is disabled 1: Noise cancellation function for the RXDn input signal is enabled. In Simple IIC mode: 0: Noise cancellation function for the SCLn and SDAn input signals is disabled 1: Noise cancellation function for the SCLn and SDAn input signals is enabled.	R/W*1
b6	BGDM	Baud Rate Generator Double-Speed Mode Select	Valid only in asynchronous mode with SCR.CKE[1] = 0: 0: Baud rate generator outputs the clock with single frequency 1: Baud rate generator outputs the clock with double frequency.	R/W*1
b7	RXDESEL	Asynchronous Start Bit Edge Detection Select	Valid only in asynchronous mode: 0: A low level on the RXDn pin is detected as the start bit 1: A falling edge on the RXDn pin is detected as the start bit.	R/W*1

Note 1. Writable only when TE in SCR/SCR_SMCI = 0 and RE in SCR/SCR_SMCI = 0 (both serial transmission and reception are disabled).

SEMR selects the clock source for 1-bit period in asynchronous mode.

BRME bit (Bit Rate Modulation Enable)

Enables and disables the bit rate modulation function. The bit rate generated by on-chip baud rate generator is evenly corrected when this function is enabled.

ABCSE bit (Asynchronous Mode Extended Base Clock Select 1)

The pulse number for a base clock at 1-bit period is 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 this bit to 0 except in asynchronous mode.

ABCS bit (Asynchronous Mode Base Clock Select)

Selects the clock cycles for 1-bit period. Set this bit to 0 except in asynchronous mode.

NFEN bit (Digital Noise Filter Function Enable)

This 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 IIC mode.

In all other modes, set the NFEN bit to 0 to disable the digital noise filter function. When the digital noise filter function is disabled, input signals are transferred as received.

BGDM bit (Baud Rate Generator Double-Speed Mode Select)

Selects the cycle of output clock for the baud rate generator to be either single or double frequency.

This 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). For the clock output from the baud rate generator, either normal or doubled frequency can be selected. 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)

Selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data receiving operation depends must be started without keeping the RXDn pin input at 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.

29.2.20 Noise Filter Setting Register (SNFR)

Address(es): SCI0.SNFR 4007 0008h, SCI1.SNFR 4007 0028h, SCI2.SNFR 4007 0048h, SCI3.SNFR 4007 0068h, SCI4.SNFR 4007 0088h, SCI9.SNFR 4007 0128h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	NFCS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	NFCS[2:0]	Noise Filter Clock Select	In asynchronous mode, the standard setting for the base clock is as follows: b2 b0 0 0 0: The clock signal divided by 1 is used with the noise filter. In simple IIC mode, the standard settings for the clock source of the on-chip baud rate generator selected by the SMR.CKS[1:0] bits are as follows: b2 b0 0 0 1: The clock signal divided by 1 is used with the noise filter 0 1 0: The clock signal divided by 2 is used with the noise filter 0 1 1: The clock signal divided by 4 is used with the noise filter 1 0 0: The clock signal divided by 8 is used with the noise filter. Other settings are prohibited.	R/W*1
b7 to b3	—	Reserved	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 IIC mode, set the bits to a value in the range from 001b to 100b.

29.2.21 I²C Mode Register 1 (SIMR1)

Address(es): SCI0.SIMR1 4007 0009h, SCI1.SIMR1 4007 0029h, SCI2.SIMR1 4007 0049h, SCI3.SIMR1 4007 0069h, SCI4.SIMR1 4007 0089h, SCI9.SIMR1 4007 0129h

	b7	b6	b5	b4	b3	b2	b1	b0
	IICDL[4:0]					—	—	IICM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	IICM	Simple IIC Mode Select	SMIF IICM 0 0: Asynchronous mode, multi-processor mode, clock synchronous mode, or simple SPI mode 0 1: Simple IIC mode 1 0: Smart card interface mode 1 1: Setting prohibited.	R/W*1
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b3	IICDL[4:0]	SDA Delay Output Select	The following cycles are of the clock signal from the on-chip baud rate generator: b7 b3 0 0 0 0 0: No output delay 0 0 0 0 1: 0 to 1 cycle 0 0 0 1 0: 1 to 2 cycles 0 0 0 1 1: 2 to 3 cycles 0 0 1 0 0: 3 to 4 cycles 0 0 1 0 1: 4 to 5 cycles : : 1 1 1 1 0: 29 to 30 cycles 1 1 1 1 1: 30 to 31 cycles.	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in SCR 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 SMIF bit in SCMR, this bit selects the operating mode.

IICDL[4:0] bits (SDA Delay Output Select)

The IICDL[4:0] bits set a delay for output 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 these bits to 00000b unless operation is in simple IIC mode. In simple IIC mode, set the bits to a value in the range from 00001b to 11111b.

29.2.22 I2C Mode Register 2 (SIMR2)

Address(es): SCI0.SIMR2 4007 000Ah, SCI1.SIMR2 4007 002Ah, SCI2.SIMR2 4007 004Ah, SCI3.SIMR2 4007 006Ah, SCI4.SIMR2 4007 008Ah, SCI9.SIMR2 4007 012Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	IICACKT	—	—	—	IICCSC	IICINTM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	IICINTM	I2C Interrupt Mode Select	0: Use ACK/NACK interrupts. 1: Use reception and transmission interrupts.	R/W*1
b1	IICCSC	Clock Synchronization	0: Do not synchronize with clock signal 1: Synchronize with clock signal	R/W*1
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	IICACKT	ACK Transmission Data	0: ACK transmission 1: NACK transmission and reception of ACK/NACK	R/W
b7, b6	—	Reserved	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 are 0 (serial reception and transmission disabled).

SIMR2 selects how reception and transmission are controlled in simple IIC mode.

IICINTM bit (I2C 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 SCL clock signal is not synchronized if the IICCSC bit is 0. The SCL clock signal is generated according to the rate selected in the BRR 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 this bit to 1 when ACK and NACK bits are received.

29.2.23 I²C Mode Register 3 (SIMR3)

Address(es): SCI0.SIMR3 4007 000Bh, SCI1.SIMR3 4007 002Bh, SCI2.SIMR3 4007 004Bh, SCI3.SIMR3 4007 006Bh, SCI4.SIMR3 4007 008Bh, SCI9.SIMR3 4007 012Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	IICSCLS[1:0]		IICSDAS[1:0]		IICSTIF	IICSTP REQ	IICRST AREQ	IICSTA REQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	IICSTAREQ	Start Condition Generation	0: Do not generate start condition 1: Generate start condition.*1, *3, *5, *6	R/W
b1	IICRSTAREQ	Restart Condition Generation	0: Do not generate restart condition 1: Generate restart condition.*2, *3, *5, *6	R/W
b2	IICSTPREQ	Stop Condition Generation	0: Do not generate stop condition 1: Generate stop condition.*2, *3, *5, *6	R/W
b3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag	0: No requests made for generating conditions, or a condition is being generated 1: Generation of start, restart, or stop condition complete. When 0 is written to IICSTIF, it is set to 0.*4	R/W*4
b5, b4	IICSDAS[1:0]	SDA Output Select	b5 b4 0 0: Serial data output 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
b7, b6	IICSCLS[1:0]	SCL Output Select	b7 b6 0 0: Serial clock output 0 1: Generate a 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 it is free.
Note 2. Generate a restart or stop condition after checking the bus state and confirming that it 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.

IICSTAREQ bit (Start Condition Generation)

When a start condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits 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 the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b and set 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 the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b and set 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 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 this 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, confirm that the IICSTIF flag is 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 (SDA Output Select)

The IICSDAS[1:0] bits control output from the SDAn pin. Set the IICSDAS[1:0] and IICSCLS[1:0] bits to the same value.

IICSCLS[1:0] bits (SCL Output Select)

The IICSCLS[1:0] bits control output from the SCLn pin. Set the IICSCLS[1:0] and IICSDAS[1:0] bits to the same value.

29.2.24 I2C Status Register (SISR)

Address(es): SCI0.SISR 4007 000Ch, SCI1.SISR 4007 002Ch, SCI2.SISR 4007 004Ch, SCI3.SISR 4007 006Ch, SCI4.SISR 4007 008Ch, SCI9.SISR 4007 012Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IICACKR
Value after reset:	0	0	x	x	0	x	0	0

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	IICACKR	ACK Reception Data Flag	0: ACK received 1: NACK received.	R
b1	—	Reserved	This bit is read as 0	R
b2	—	Reserved	The read value is undefined	R
b3	—	Reserved	This bit is read as 0	R
b5, b4	—	Reserved	The read values are undefined	R
b7, b6	—	Reserved	These bits are read as 0	R

SISR monitors state in simple IIC mode.

IICACKR flag (ACK Reception Data Flag)

Received ACK and NACK bits can be read from the IICACKR flag. This flag is updated on the rising edge of the SCLn clock for the ACK/NACK receiving bit.

29.2.25 SPI Mode Register (SPMR)

Address(es): SCI0.SPMR 4007 000Dh, SCI1.SPMR 4007 002Dh, SCI2.SPMR 4007 004Dh, SCI3.SPMR 4007 006Dh, SCI4.SPMR 4007 008Dh, SCI9.SPMR 4007 012Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	CKPH	CKPOL	—	MFF	—	MSS	CTSE	SSE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SSE	SSn Pin Function Enable	0: Disable SSn pin function 1: Enable SSn pin function.	R/W*1
b1	CTSE	CTS Enable	0: Disable CTS function (RTS output function is enabled) 1: Enable CTS function.	R/W*1
b2	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
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	MFF	Mode Fault Flag	0: No mode fault error 1: Mode fault error.	R/W*2
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	CKPOL	Clock Polarity Select	0: Do not invert clock polarity 1: Invert clock polarity.	R/W*1
b7	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 SCR are 0 (both serial transmission and reception are disabled).

Note 2. Only 0 can be written to these bits 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 if the SSn pin is to be used to control transmission and reception in simple SPI mode. Set this bit to 0 in all other modes. In addition, for usage in simple SPI mode, the SSn pin on the master side is not required to control reception and transmission when master mode (SCR.CKE[1:0] = 00b and MSS = 0) is selected and there is a single master. Therefore, the setting for the SSE bit is 0. Do not set both the SSE and CTSE bits to enabled as 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 of 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 enabled as 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 the MSS 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 29.70 for details.

Set the bit to 0 in 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 29.70 for details.

Set the bit to 0 in modes other than simple SPI mode and clock synchronous mode.

29.2.26 FIFO Control Register (FCR)

Address(es): SCI0.FCR 4007 0014h, SCI1.FCR 4007 0034h, SCI2.FCR 4007 0054h, SCI3.FCR 4007 0074h, SCI4.FCR 4007 0094h, SCI9.FCR 4007 0134h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit name	Description	R/W
b0	FM	FIFO Mode Select	Valid only in asynchronous mode, including multi-processor, 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
b1	RFRST	Receive FIFO Data Register Reset	Valid only when FCR.FM = 1: 0: Do not reset FRDRHL 1: Reset FRDRHL.	R/W
b2	TFRST	Transmit FIFO Data Register Reset	Valid only when FCR.FM = 1: 0: Do not reset FTDRHL 1: Reset FTDRHL.	R/W
b3	DRES	Receive Data Ready Error Select	Select the interrupt request when detecting receive data ready: 0: Receive data full interrupt (SCIn_RXI) 1: Receive error interrupt (SCIn_ERI).	R/W
b7 to b4	TTRG[3:0]	Transmit FIFO Data Trigger Number	Valid only in asynchronous mode, including multi-processor, or clock synchronous mode: 0000: Trigger number 0 : 1111: Trigger number 15.	R/W
b11 to b8	RTRG[3:0]	Receive FIFO Data Trigger Number	Valid only in asynchronous mode, including multi-processor, or clock synchronous mode: 0000: Trigger number 0 : 1111: Trigger number 15.	R/W
b15 to b12	RSTRG[3:0]	RTS Output Active Trigger Number Select	Valid only in asynchronous mode, including multi-processor, or clock synchronous mode, while FCR.FM = 1, SPMR.CTSE = 0, and SPMR.SSE = 0: 0000: Trigger number 0 : 1111: Trigger number 15.	R/W

Note 1. Writable only when TE = 0 and RE = 0.

FCR selects FIFO mode, reset FTDRHL/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)

The FRDRHL register is reset when the RFRST bit is set to 1, and the number of receive data is reset to 0.

After writing 1, this bit is set to 0 after 1 PCLK.

TFRST bit (Transmit FIFO Data Register Reset)

The FTDRHL register is reset when the TFRST bit is set to 1, and the transmit data count is reset to 0. After writing 1, this bit is set to 0 after 1 PCLK.

DRES bit (Receive Data Ready Error Select)

On detecting a receive data ready error, the DRES bit selects the interrupt request from an SCIn_RXI interrupt request or an SCIn_ERI interrupt request. Set the DRES bit to 1 when starting the DMAC or DTC and reading the FRDRH and FRDRL registers.

TTRG[3:0] bits (Transmit FIFO Data Trigger Number)

The TDFE flag is set to 1 when the amount of transmit data in the Transmit FIFO Data Register (FTDRHL) is equal to or less than the specified transmit triggering number, and software can write data to FTDRHL. If SCR.TIE = 1, SCIn_TXI interrupt request occurred.

RTRG[3:0] bits (Receive FIFO Data Trigger Number)

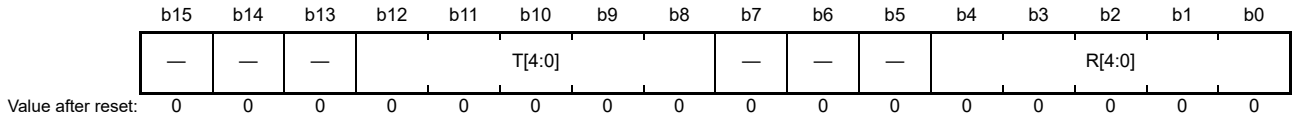
The RDF flag is set to 1 when the amount of receive data in the Receive FIFO Data Register (FRDRHL) is equal to or greater than the specified receive triggering number, and software can read data from FRDRHL. If SCR.RIE = 1, SCIn_RXI interrupt request occurred. When RTRG[3:0] is set to 0, the RDF flag is not set even when the amount of the data in the receive FIFO is equal to 0. Additionally, 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 the Receive FIFO Data Register (FRDRHL) is equal to or greater than the specified receive triggering number, the RTS signal goes high. When RSTRG[3:0] is set to 0, the RTS signal does not go high even when the amount of data in the receive FIFO is equal to 0.

29.2.27 FIFO Data Count Register (FDR)

Address(es): SCI0.FDR 4007 0016h, SCI1.FDR 4007 0036h, SCI2.FDR 4007 0056h, SCI3.FDR 4007 0076h, SCI4.FDR 4007 0096h, SCI9.FDR 4007 0136h



Bit	Symbol	Bit name	Description	R/W
b4 to b0	R[4:0]	Receive FIFO Data Count	Indicates the amount of receive data stored in FRDRHL (valid only in asynchronous mode (including multi-processor) or clock synchronous mode, when FCR.FM = 1)	R
b7 to b5	—	Reserved	These bits are read as 0.	R
b12 to b8	T[4:0]	Transmit FIFO Data Count	Indicates the amount of non-transmit data stored in FTDRHL (valid only in asynchronous mode (including multi-processor) or clock synchronous mode, when FCR.FM = 1).	R
b15 to b13	—	Reserved	These bits are read as 0.	R

The FDR register indicates the amount of data stored in FRDRHL/FTDRHL.

R[4:0] bits (Receive FIFO Data Count)

The R[4:0] bits indicate the amount of receive data stored in FRDRHL. 00h means no receive data, and 10h 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. 00h means no transmit data, and 10h means that all (maximum count) of the data to be transmitted is stored in FTDRHL.

29.2.28 Line Status Register (LSR)

Address(es): SCI0.LSR 4007 0018h, SCI1.LSR 4007 0038h, SCI2.LSR 4007 0058h, SCI3.LSR 4007 0078h, SCI4.LSR 4007 0098h, SCI9.LSR 4007 0138h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	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	Bit name	Description	R/W
b0	ORER	Overrun Error Flag	Valid only in asynchronous mode, including multi-processor, or clock synchronous mode, and when FIFO is selected: 0: No overrun error occurred. 1: An overrun error occurred.	R*1
b1	—	Reserved	This bit is read as 0.	R
b6 to b2	FNUM[4:0]	Framing Error Count	Indicates the amount of data with a framing error in the receive data stored in the Receive FIFO Data Register (FRDRHL).	R
b7	—	Reserved	This bit is read as 0.	R
b12 to b8	PNUM[4:0]	Parity Error Count	Indicates the amount of data with a parity error in the receive data stored in the Receive FIFO Data Register (FRDRHL).	R
b15 to b13	—	Reserved	These bits are read as 0.	R

Note 1. If this flag is 1, the read of SSR_FIFO register is not complete. Write 0 to SSR_FIFO.ORER to clear the flag.

The LSR register indicates the status of receive error.

ORER bit (Overrun Error Flag)

The ORER bit reflects the value in SSR_FIFO.ORER.

FNUM[4:0] bits (Framing Error Count)

The FNUM[4:0] value indicates the amount of data stored in the FRDRHL register with a framing error.

PNUM[4:0] bits (Parity Error Count)

The PNUM[4:0] value indicates the amount of data stored in the FRDRHL register with a parity error.

29.2.29 Compare Match Data Register (CDR)

Address(es): SCI0.CDR 4007 001Ah, SCI1.CDR 4007 003Ah, SCI2.CDR 4007 005Ah, SCI3.CDR 4007 007Ah, SCI4.CDR 4007 009Ah, SCI9.CDR 4007 013Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPD[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b8 to b0	CMPD[8:0]	Compare Match Data	Compare data pattern for address match wakeup function	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The CDR register sets 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 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.

29.2.30 Data Compare Match Control Register (DCCR)

Address(es): SCI0.DCCR 4007 0013h, SCI1.DCCR 4007 0033h, SCI2.DCCR 4007 0053h, SCI3.DCCR 4007 0073h, SCI4.DCCR 4007 0093h, SCI9.DCCR 4007 0133h

	b7	b6	b5	b4	b3	b2	b1	b0
	DCME	IDSEL	—	DFER	DPER	—	—	DCMF
Value after reset:	0	1	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	DCMF	Data Compare Match Flag	0: Not matched 1: Matched	R/(W)*1
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	DPER	Data Compare Match Parity Error Flag	0: No parity error occurred 1: A parity error occurred.	R/(W)*1
b4	DFER	Data Compare Match Framing Error Flag	0: No framing error occurred 1: A framing error occurred.	R/(W)*1
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	IDSEL	ID Frame Select	Valid only in asynchronous mode, including multi-processor: 0: Always compare data regardless of the MPB bit value 1: Compare data when the MPB bit = 1 (ID frame).	R/W
b7	DCME	Data Compare Match Enable	Valid only in asynchronous mode, including multi-processor: 0: Address match function is disabled 1: Address match function is enabled.	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 RE bit to 0 in the Serial Control Register (SCR) does not affect the DCMF flag, which retains its previous state.

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 the frame where an address match is detected.

[Clearing conditions]

- When 0 is written after 1 is read from DPER.
- When the RE bit in SCR 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 (reception data match detection).

[Setting conditions]

- When a stop bit of the frame in which an address match is detected is 0.
- When in 2-stop mode, only the first bit of the stop bits is checked for a value of 1 (the second bit is not checked).

[Clearing conditions]

- When 0 is written after 1 is read from DFER.
- When the RE bit in SCR 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 selects whether the address match function (data compare match function) is enabled or not.

If the SCI detects a match to the comparison data (CDR.CMPD) with the receive data, DCME clears automatically, after which SCI operation mode is receive mode without the data compare match function. See section 29.3.6, Address Match (Receive Data Match Detection) Function.

The write value should be 0 for any mode other than asynchronous mode.

29.2.31 Serial Port Register (SPTR)

Address(es): SCI0.SPTR 4007 001Ch, SCI1.SPTR 4007 003Ch, SCI2.SPTR 4007 005Ch, SCI3.SPTR 4007 007Ch, SCI4.SPTR 4007 009Ch, SCI9.SPTR 4007 013Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPB2IO	SPB2DT	RXDMON
Value after reset:	0	0	0	0	0	0	1	1

Bit	Symbol	Bit name	Description	R/W
b0	RXDMON	Serial Input Data Monitor	The state of the RXDn pin: 0: RXDn pin is low 1: RXDn pin is high.	R
b1	SPB2DT	Serial Port Break Data Select	The output level of TXDn pin when SCR.TE = 0: 0: Output low on TXDn pin 1: Output high on TXDn pin.	R/W
b2	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 in TXDn pin 1: Output value of SPB2DT bit in TXDn pin.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPTR register provides confirmation of serial reception pin (RXDn pin) status and sets transmission pin (TXDn 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 29.21.

Table 29.21 TXDn pin status

Value of SCR.TE	Value of SPTR.SPB2IO	Value of SPTR.SPB2DT	TXDn pin status
0	0	x	Hi-Z (initial value)
0	1	0	Low level output
0	1	1	High level output
1	x	x	Serial transmit data is output

x: Don't care.

Note: Use the SPTR in asynchronous mode only. Using this register in any other mode is not guaranteed.

29.3 Operation in Asynchronous Mode

Figure 29.2 shows the general format for asynchronous serial communications.

One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level).

In asynchronous serial communications, the communications line is held in the mark state (high level) when not communicating.

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 the 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.

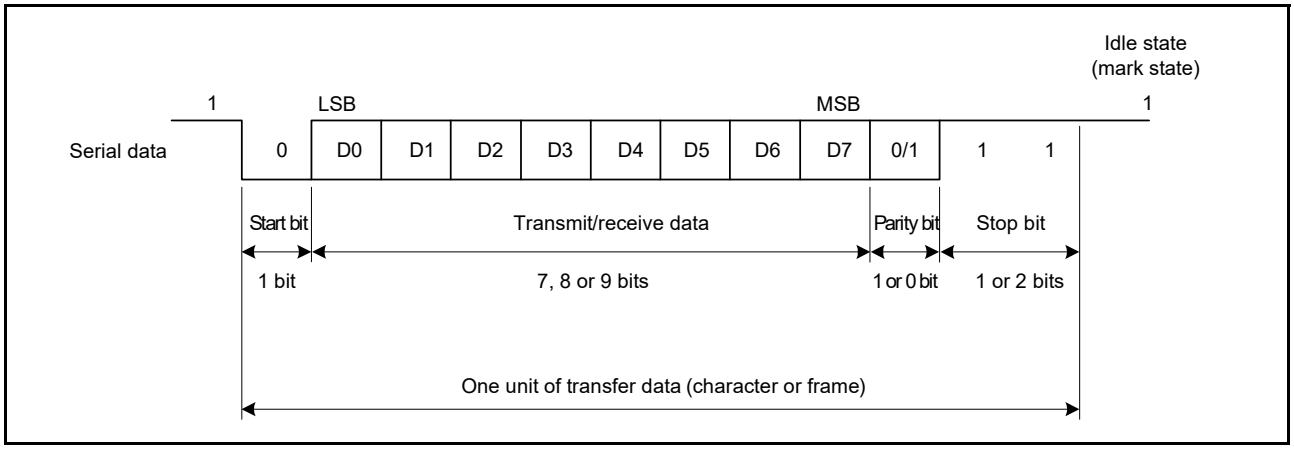


Figure 29.2 Data format in asynchronous serial communications with 8-bit data, parity bit, and 2 stop bits

29.3.1 Serial Data Transfer Format

Table 29.22 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 29.4, Multi-Processor Communication Function.

Table 29.22 Serial transfer formats (asynchronous mode) (1 of 2)

SCMR setting	SMR setting				Serial transfer format and frame length
					1 2 3 4 5 6 7 8 9 10 11 12 13
CHR1	CHR	PE	MP	STOP	
0	0	0	0	0	S 9-bit data STOP
0	0	0	0	1	S 9-bit data STOP STOP
0	0	1	0	0	S 9-bit data P STOP
0	0	1	0	1	S 9-bit data P STOP STOP
1	0	0	0	0	S 8-bit data STOP
1	0	0	0	1	S 8-bit data STOP STOP
1	0	1	0	0	S 8-bit data P STOP

Table 29.22 Serial transfer formats (asynchronous mode) (2 of 2)

SCMR setting	SMR setting				Serial transfer format and frame length
					1 2 3 4 5 6 7 8 9 10 11 12 13
CHR1	CHR	PE	MP	STOP	
1	0	1	0	1	S | 8-bit data | P | STOP | STOP
1	1	0	0	0	S | 7-bit data | STOP
1	1	0	0	1	S | 7-bit data | STOP | STOP
1	1	1	0	0	S | 7-bit data | P | STOP
1	1	1	0	1	S | 7-bit data | P | STOP | STOP
0	0	—	1	0	S | 9-bit data | MPB | STOP
0	0	—	1	1	S | 9-bit data | MPB | STOP | STOP
1	0	—	1	0	S | 8-bit data | MPB | STOP
1	0	—	1	1	S | 8-bit data | MPB | STOP | STOP
1	1	—	1	0	S | 7-bit data | MPB | STOP
1	1	—	1	1	S | 7-bit data | MPB | STOP | STOP

S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multi-processor bit

29.3.2 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 29.3. The reception margin in asynchronous mode is determined by the following formula (1):

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\ F - \frac{D - 0.5}{N}\ (1 + F) \right| \times 100\ [\%] \cdots \text{Formula (1)}$$

M: Reception margin
N: Ratio of bit rate to clock
(N = 16 when ABCSE in SEMR = 0 and ABCS in SEMR = 0,
N = 8 when ABCS in SEMR = 1, N = 6 when ABCSE in SEMR = 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 by the following formula:

$$M = \{0.5 - 1/(2 \times 16)\} \times 100\ (\%) = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

Note 1. In this example, the ABCS bit in SEMR is 0 and ABCSE bit in SEMR 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.

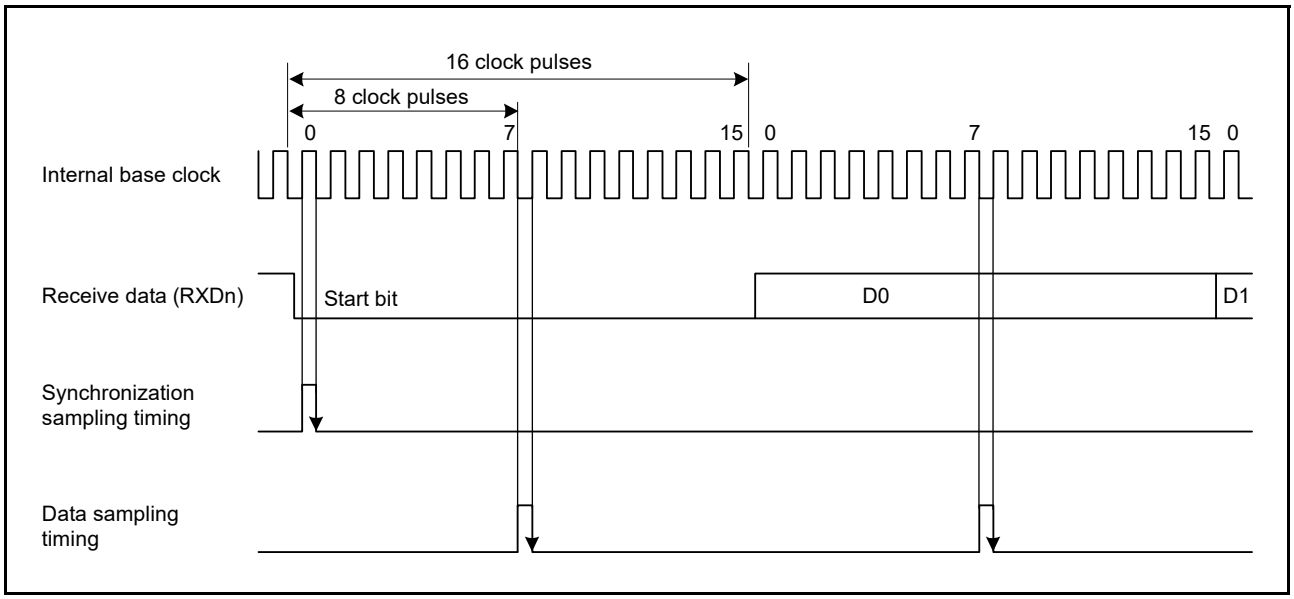


Figure 29.3 Receive data sampling timing in asynchronous mode

29.3.3 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 SCI transfer clock based on the CM setting in SMR and the CKE[1:0] setting in SCR.

When an external clock is input to the SCKn pin, the clock frequency must be 16 times the bit rate when ABCS in SEMR = 0 or 8 times the bit rate when ABCS in SEMR = 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 such that the rising edge of the clock is in the middle of the transmit data, as Figure 29.4 shows.

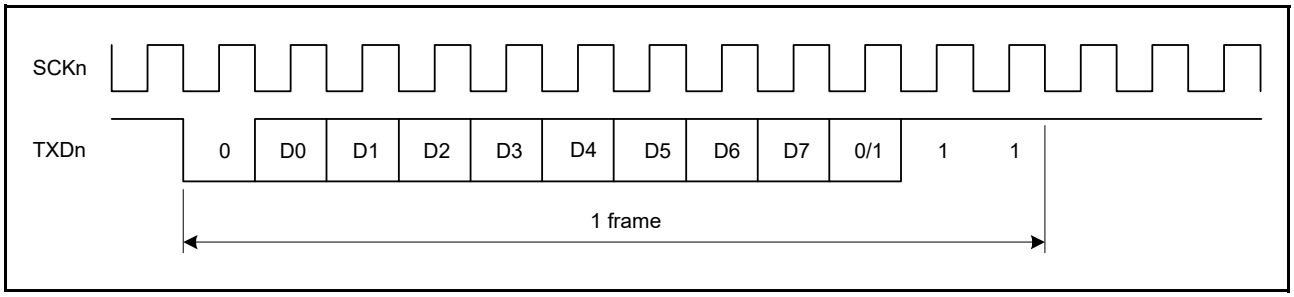


Figure 29.4 Phase relationship between output clock and transmit data in asynchronous mode when SMR.CHR = 0, PE = 1, MP = 0, and STOP = 1

29.3.4 Double-Speed Operation and Frequency of 6 Times the Bit Rate

When the Asynchronous Mode Base Clock Select (ABCS) bit in SEMR 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 when ABCS is set to 0. When the Baud Rate Generator Double-speed Mode (BGDM) bit in SEMR is set to 1, the cycle of the base clock is half and the bit rate is double that when BGDM is set to 0. When the CKE[1] bit in SCR 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 ABCSE bit in SEMR is set to 1, the number of basic clock pulses are 6 during a period of 1 bit, and SCI operates at a bit rate 16/3 times that when SEMR.ABCS = 0, SEMR.BGDM = 0 and SMER.ABCSE = 0.

As shown by Formula (1) in section 29.3.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, the reception margin decreases when the ABCS bit or ABCSE bit in SEMR 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 ABCSE and ABCS set to 0.

29.3.5 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, driving the CTSn_RTSn pin low 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 that uses output on the CTSn_RTSn pin, a low level is output when reception becomes possible. Conditions for output low level and high level are shown in this section.

[Conditions for low-level output]

(a) Non-FIFO selected, when all of the following conditions are satisfied

- The value of the RE bit in SCR is 1
- Reception is not in progress
- There is no received data yet to be read
- The ORER, FER, and PER flags in SSR are all 0.

(b) FIFO selected, when all of the following conditions are satisfied

- The value of the RE bit in SCR is 1
- When the amount of receive data written in FRDRHL is equal to or less than the specified receive triggering number
- The ORER flag in SSR_FIFO (ORER in the FRDRH) is 0.

[Condition for high-level output]

(a) Non-FIFO selected

- The conditions for low-level output are not satisfied.
- When reception is terminated with SCR.RE = 0 without reading the RDR register after reception is completed, RTS remains High. At this time, read the SCR register for dummy after writing SCR.RE = 0.

(b) FIFO selected

- The conditions for low-level output are not satisfied.

29.3.6 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*4, when one frame of data is received, the SCI compares that received data with the data set in CDR.CMPD. If SCI detects a match to the comparison data (CDR.CMPD*3) 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 = 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. Receive data where the MPB bit is 0 is always treated as a non-match.

If DCCR.IDSEL is set to 0, the SCI performs address match or non-match regardless of the MPB bit value of the received data.

Until the SCI detects a match to the comparison data (CDR.CMPD*3) with receive data, received data is skipped (discarded), and the SCI cannot detect parity error or framing error. When the SCI detects a match, DCCR.DCME is automatically cleared, and DCCR.DCMF is set to 1.

If DCCR.IDSEL is set to 1, the SCR.MPIE bit is automatically cleared. If DCCR.IDSEL is set to 0, the value of SCR.MPIE bit is retained. If SCR.RIE 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, DCCR.DFER is set to 1, and if the SCI detects a parity error in that frame, DCCR.DPER is set to 1. The compared receive data is not stored in RDR*1, and SSR.RDRF remains 0.*2

After 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 29.5 and Figure 29.6.

Note 1. When FCR.FM = 1, this refers to the FRDRHL register.
Note 2. When FCR.FM = 1, this refers to the SSR_FIFO.RDF flag.
Note 3. 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 4. Set the DCCR.DCME bit to 1 before receiving the start bit of the received frame that performs address matching.

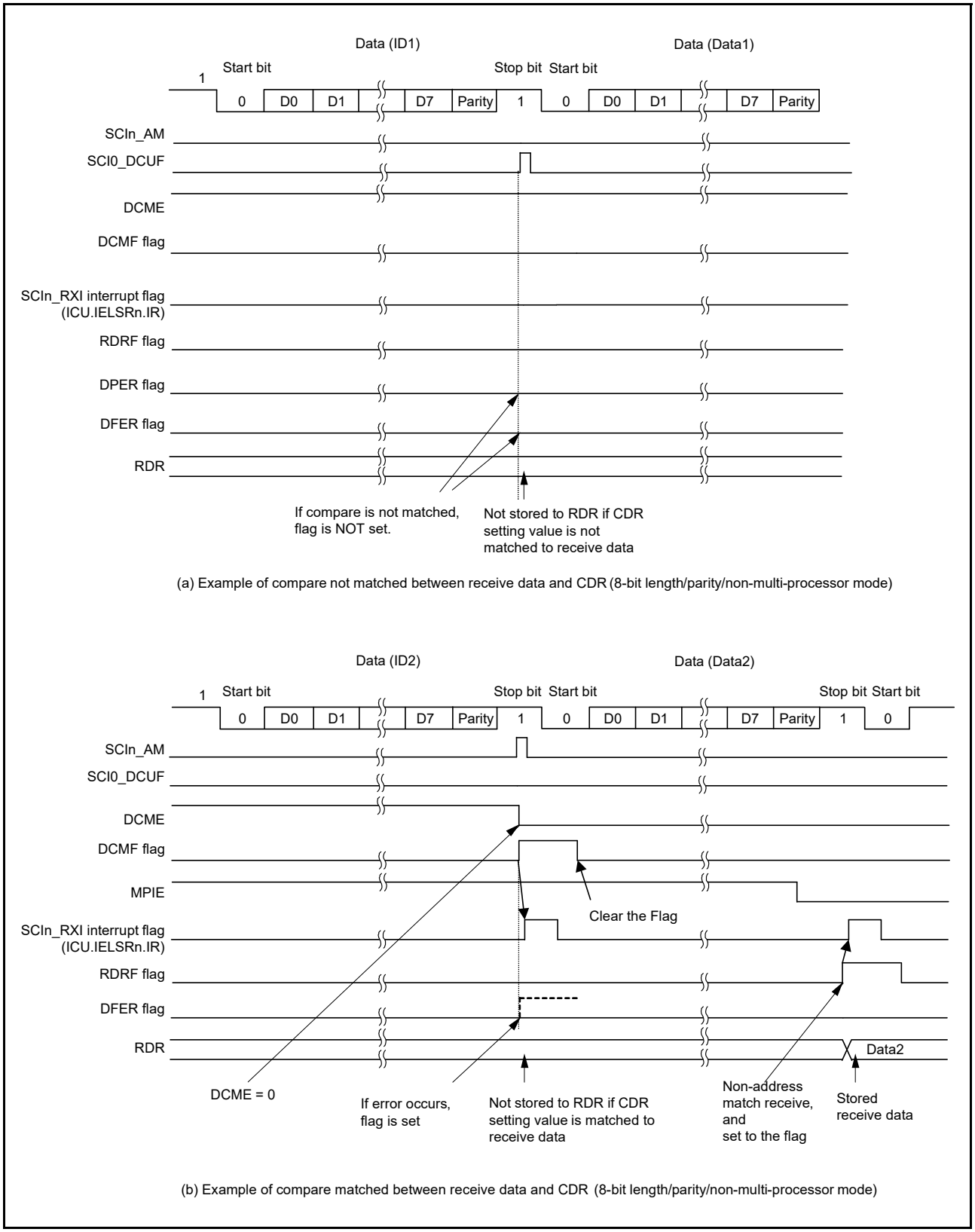


Figure 29.5 Example of address match (1) non-multi-processor mode

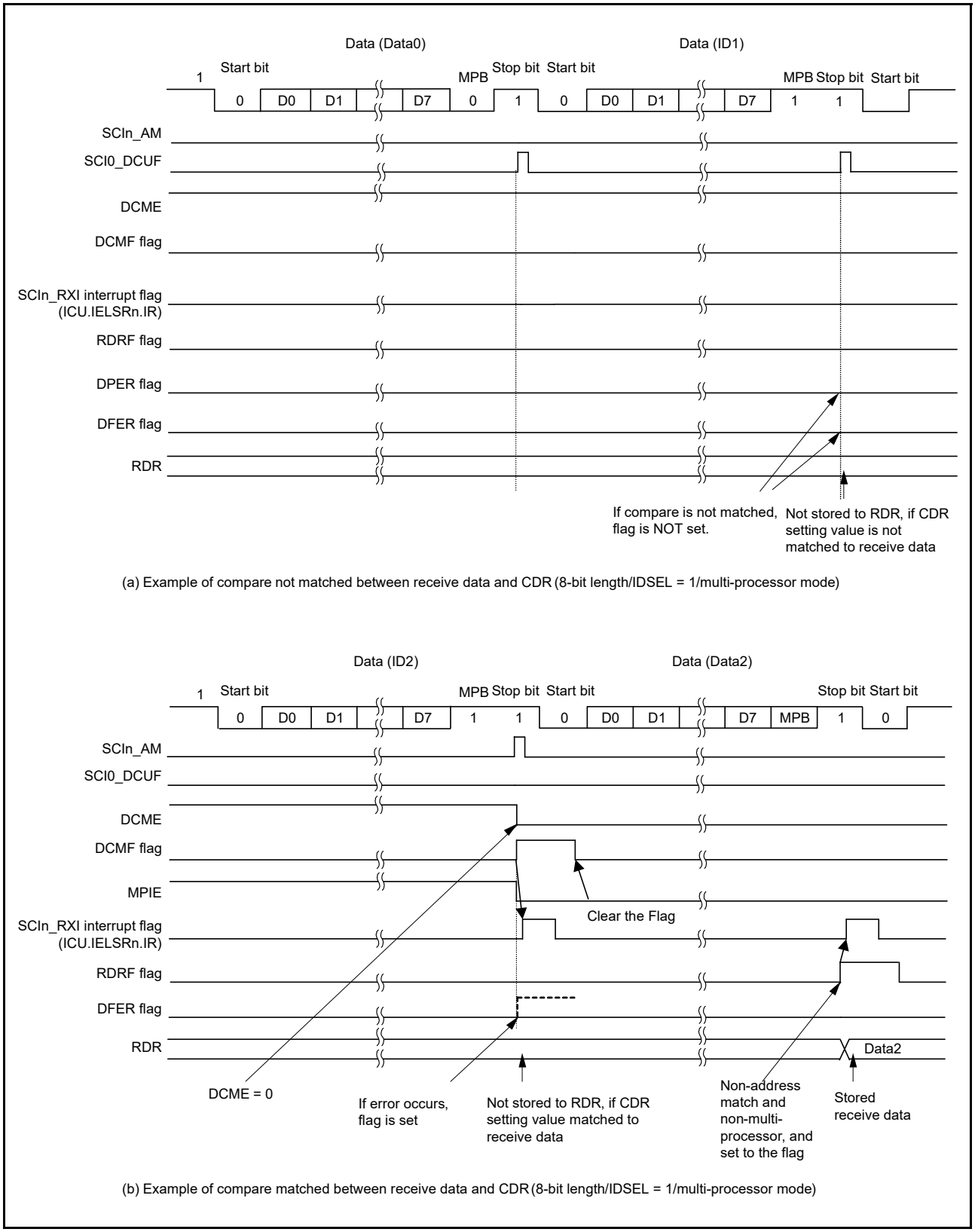


Figure 29.6 Example of address match (2) multi-processor mode

29.3.7 SCI Initialization in Asynchronous Mode

Before transmitting and receiving data, start by writing the initial value 00h to SCR, then continue through the SCI procedure (select non-FIFO or FIFO) shown in Figure 29.7 and Figure 29.8. Whenever the operating mode or transfer format is to be changed, SCR 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 SCR.TE bit is set to 0, the TEND flag for the selected FIFO buffer is not initialized.

Note: 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.

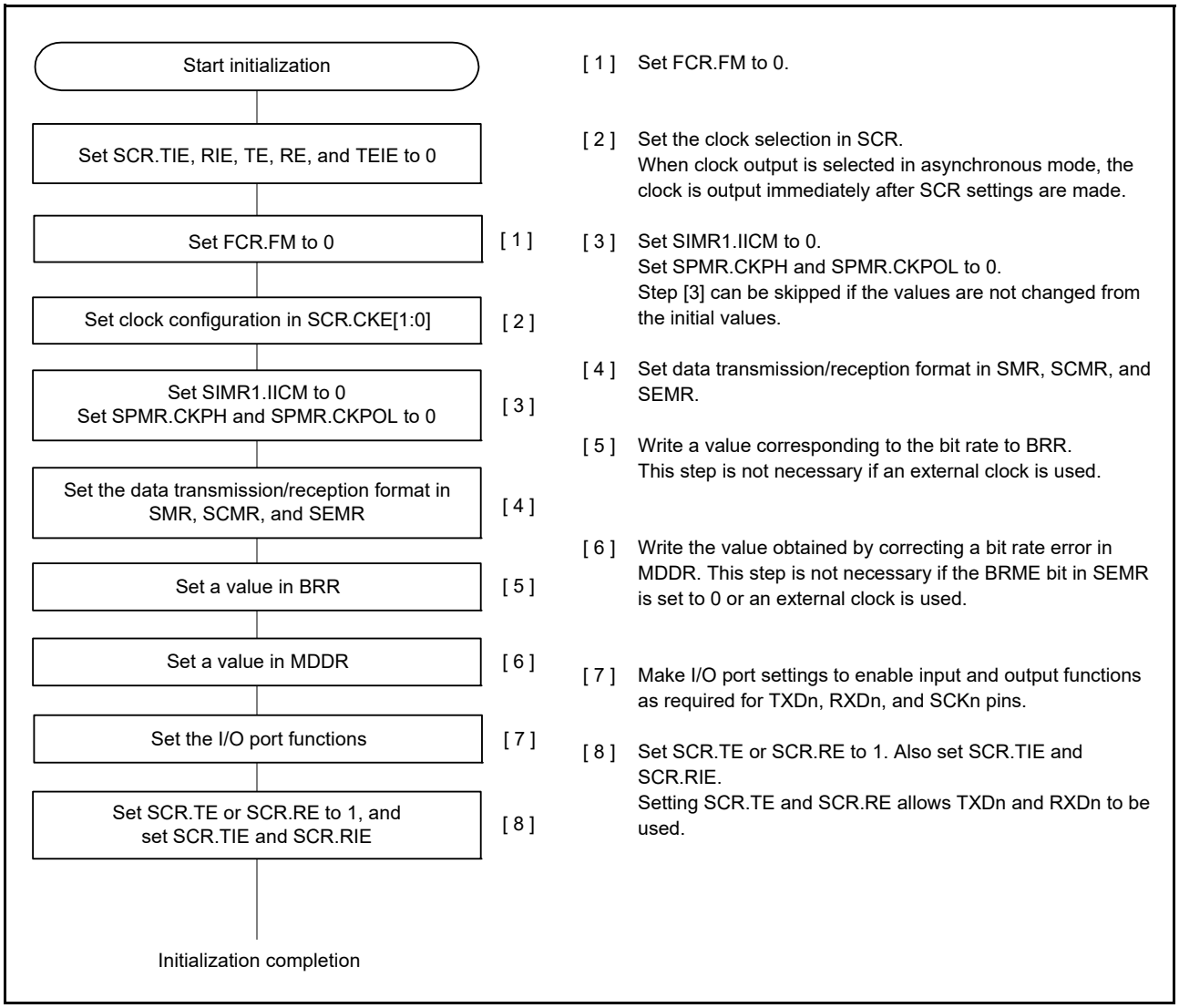


Figure 29.7 Example flow of SCI initialization in asynchronous mode with non-FIFO selected

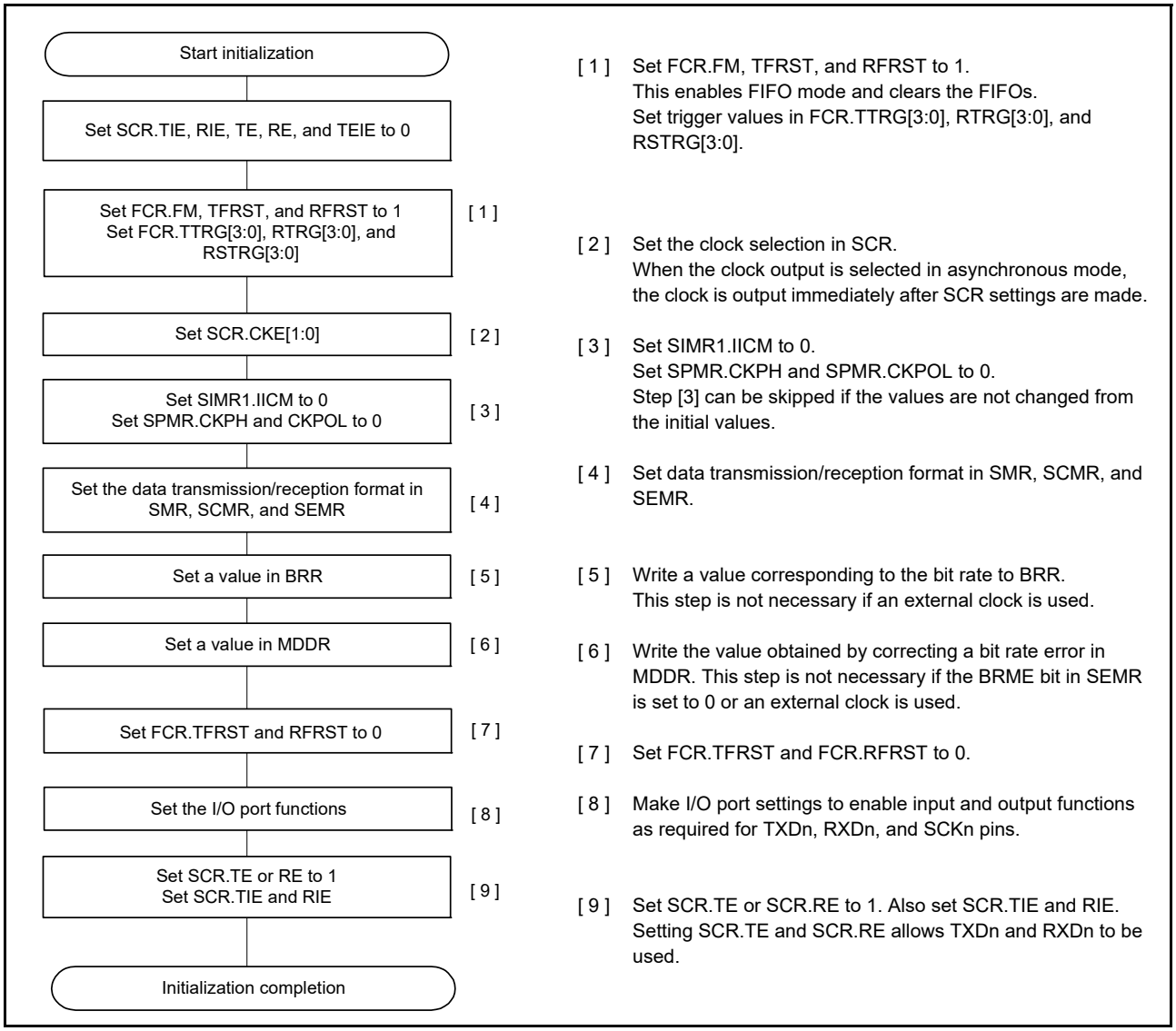


Figure 29.8 Example flow of SCI initialization in asynchronous mode with FIFO selected

29.3.8 Serial Data Transmission in Asynchronous Mode

(1) Non-FIFO selected

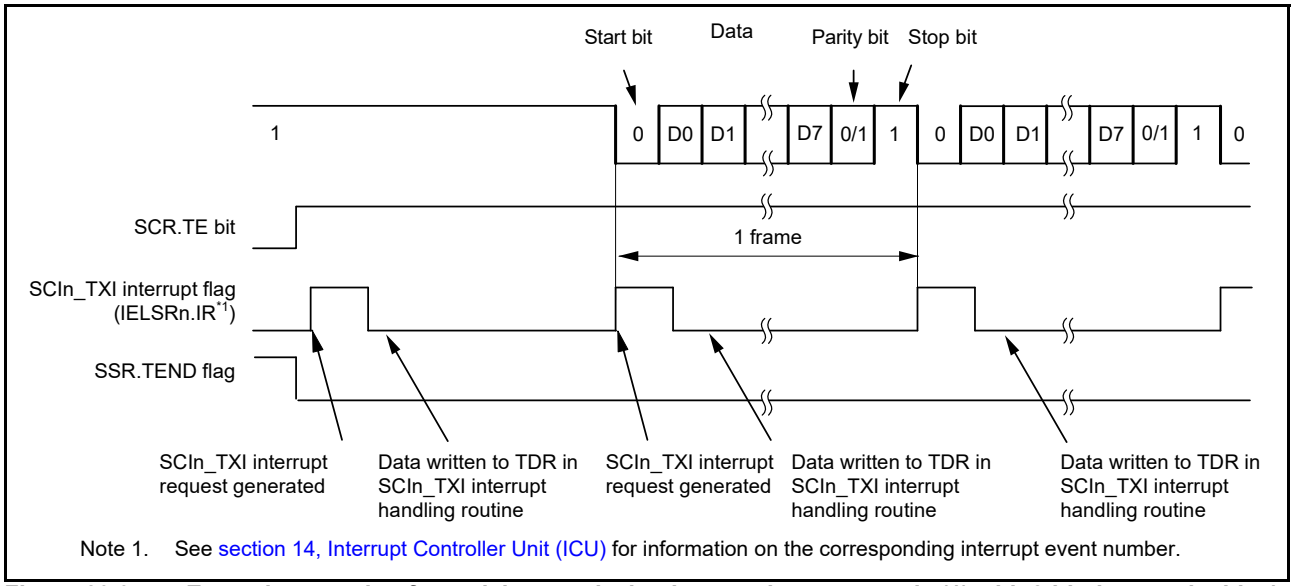
Figure 29.9, Figure 29.10, and Figure 29.11 show examples of serial transmission in asynchronous mode.

In serial transmission, the SCI operates as described in this section. When the SCR.TE bit is set to 1, the high level for one frame (preamble) is output to TXD.

1. The SCI transfers data from TDR*1 to TSR 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 TDR*1 to TSR. If the TIE bit in SCR is 1, an SCIn_TXI interrupt request is generated. Continuous transmission is possible by writing the next transmit data to TDR*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 SCR.TIE to 0 (an SCIn_TXI interrupt request is disabled) and SCR.TEIE to 1 (an SCIn_TEI interrupt request is enabled) after the last of the data to be transmitted is written to the TDR*1 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 on output of the stop bit.
5. When TDR 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 TDR*1 to TSR and transmission of the stop bit, after which serial transmission of the next frame starts.
6. If TDR is not updated, the TEND flag in SSR is set to 1, the stop bit is sent, and the mark state is entered where 1 is output. If the TEIE bit in SCR is 1, the TEND flag in SSR is set to 1 and an SCIn_TEI interrupt request is generated.

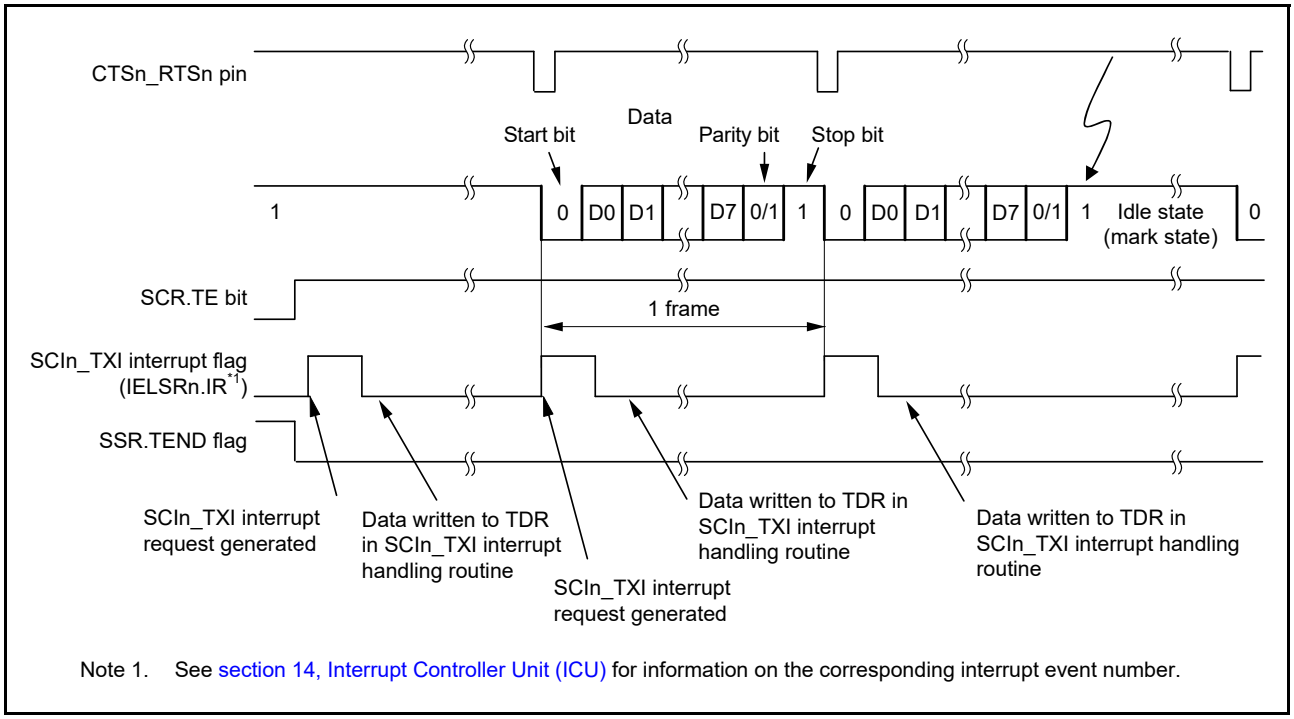
Note 1. Only write data to TDRHL when 9-bit data length is selected.

Figure 29.9, Figure 29.10, and Figure 29.11 show a sample flowchart for serial transmission in asynchronous mode.



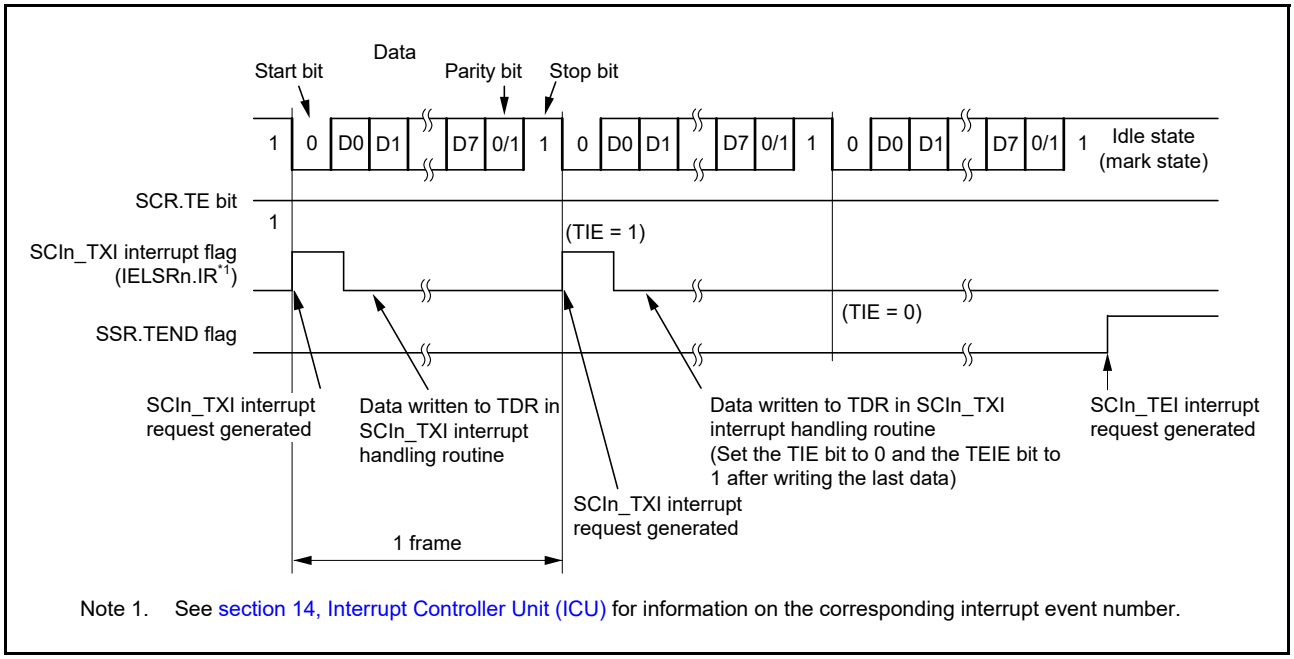
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 29.9 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



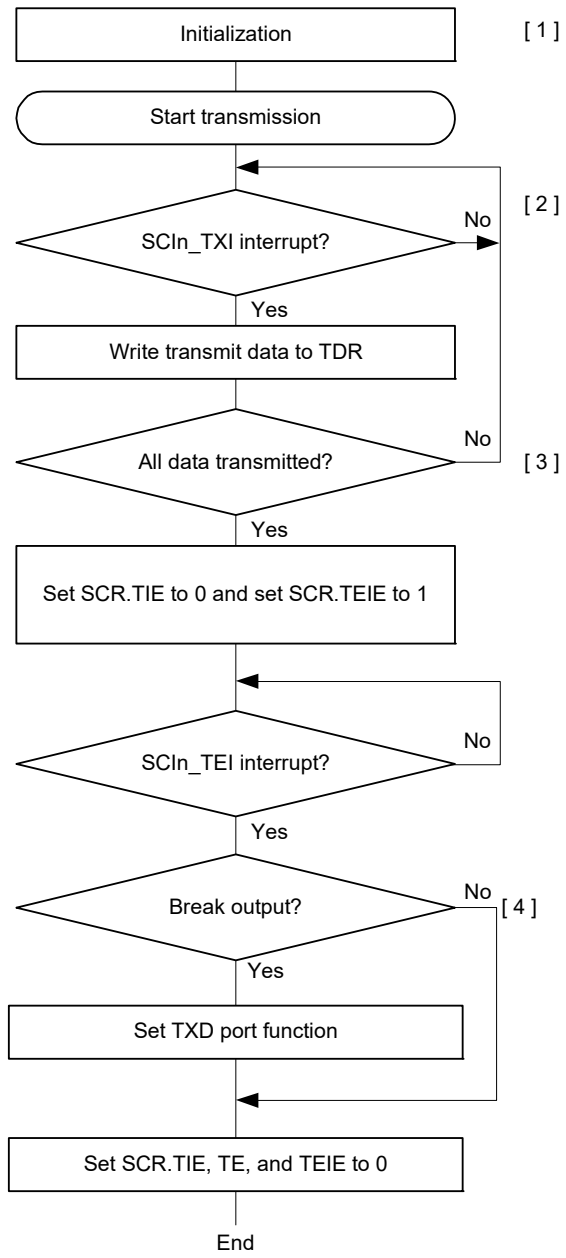
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 29.10 Example operation for serial transmission in asynchronous mode (2) with 8-bit data, parity bit, 1 stop bit, CTS function used, and at the beginning of transmission



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 29.11 Example operation for serial transmission in asynchronous mode (3) with 8-bit data, parity bit, 1 stop bit, CTS function not used, and from the middle of transmission until transmission completion



[[1] SCI Initialization:
Set data transmission.
After SCR.TE is set to 1, 1 is output for a frame (preamble), and transmission is enabled.

[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] 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 DMAC or DTC.
When SCIn_TEI interrupt requests are in use, set SCR.TIE to 0 and SCR.TEIE to 1 after the last of the data to be transmitted are written to 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 TXDn pin by SPTR.SPB2IO and SPTR.SPB2DT bits, set SCR.TE to 0.

Note: TDR becomes TDRHL when 9-bit data length is selected.

Figure 29.12 Example flow of serial transmission in asynchronous mode with non-FIFO selected

(2) FIFO selected

Figure 29.13 shows an example of a data format that is written to FTDRH and FTDRL in asynchronous mode.

Data is set to FTDRH and FTDRL that corresponds to the 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	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								

—: Invalid. The write value should be 0.

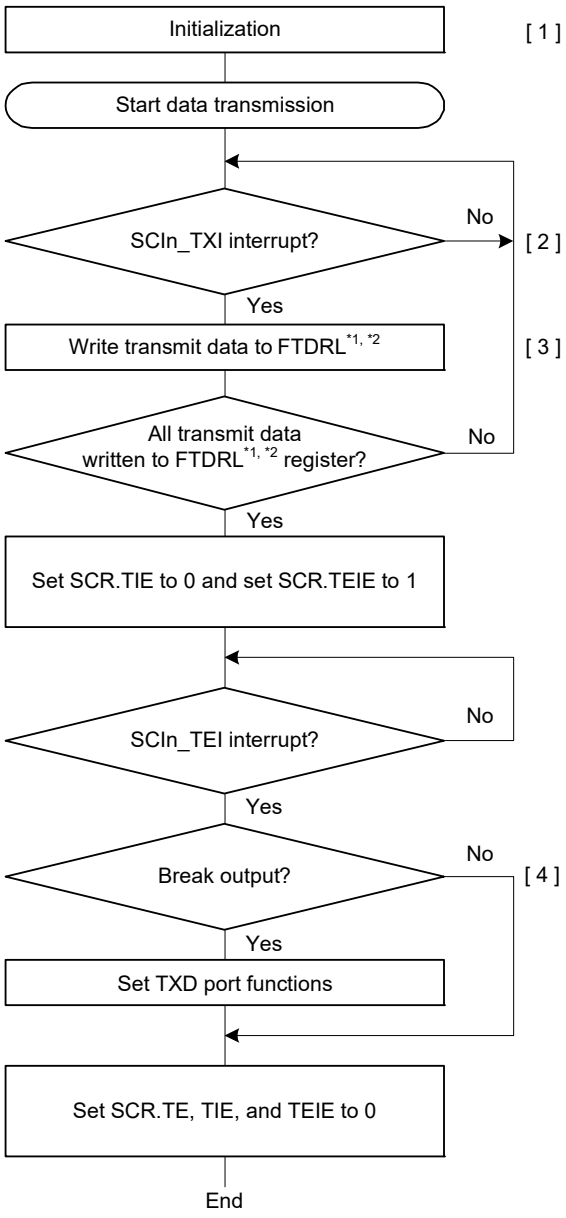
Figure 29.13 Data format written to FTDRH and FTDRL (FIFO selected)

In serial transmission, the SCI operates as described in this section. When the SCR.TE bit is set to 1, the high level for one frame (preamble) is output to TXD.

1. The SCI transfers data from FTDRL*1 to TSR 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 TE and TIE bits in SCR are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CTSE bit in SPMR is set to 0 (CTS function is disabled) and a low level on the CTSn_RTSn pin causes data transfer from FTDRL*1 to TSR. 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 TIE bit in SCR 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 SCR.TIE to 0 (an SCIn_TXI interrupt request is disabled) and SCR.TEIE 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 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 whether non-transmitted data remains in FTDRL*3 or not on output of the stop bit.
5. When data is set to FTDRL*3, setting of SPMR.CTSE 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 TEIE bit in SCR is 1, the TEND flag in SSR_FIFO is set to 1 and an SCIn_TEI interrupt request is generated.

Note 1. Write data to the FTDRH and FTDRL registers when 9-bit data length is selected.
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 29.14 shows an example flow of serial transmission in asynchronous mode with FIFO selected.



[1] SCI initialization:
Set data transmission.
After SCR.TE is set to 1, 1 is output for a frame (preamble), 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 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*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 is 16 - (the number of stored data in the transmit FIFO).
Transmit data can also be written to FTDRL by activating the DMAC or DTC. When writing the data to FTDRL by DMAC or DTC, the TDFE flag is cleared automatically, so do not write to the TDFE flag.
When SCIn_TEI interrupt requests are in use, set SCR.TIE to 0 and SCR.TEIE to 1 after the last of the data to be transmitted is written to the 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 TXDn pin by SPTR.SPB2IO and SPTR.SPB2DT, set SCR.TE 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 the order from FTDRH to FTDRL.

Figure 29.14 Example flow of serial transmission in asynchronous mode with FIFO selected

29.3.9 Serial Data Reception in Asynchronous Mode

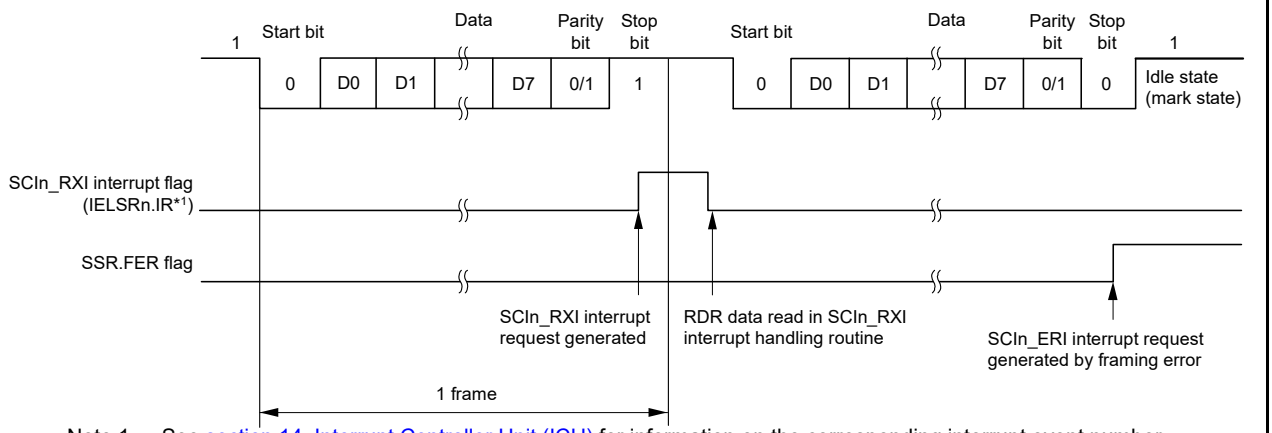
(1) Non-FIFO selected

Figure 29.15 and Figure 29.16 show an example of the operation for serial data reception in asynchronous mode.

In serial data reception, the SCI operates as follows:

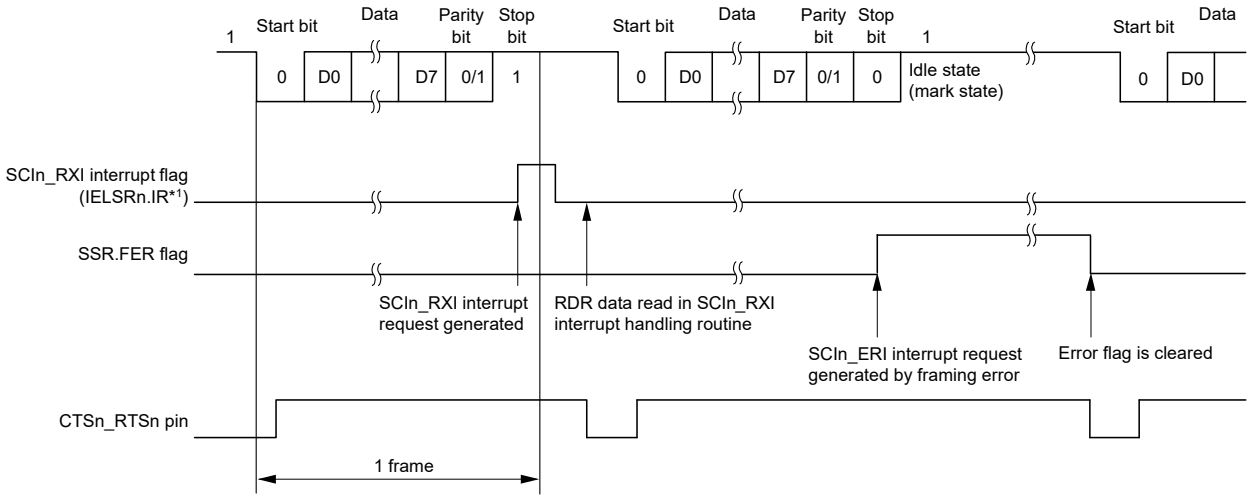
1. When the value of the RE bit in SCR becomes 1, the output signal on the CTSn_RTSn pin goes low.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If an overrun error occurs, the ORER flag in SSR is set to 1. If the RIE bit in SCR is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to RDR*1.
4. If a parity error is detected, the PER flag in SSR is set to 1 and receive data is transferred to RDR*1. If the RIE bit in SCR is 1, an SCIn_ERI interrupt request is generated.
5. If a frame error is detected, the FER bit in the SSR is set to 1 and receive data is transferred to RDR*1. If the RIE bit in the SCR is 1, an SCIn_ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to RDR*1. If the RIE bit in the SCR is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR in the SCIn_RXI interrupt handling routine before reception of the next receive data is complete. Reading the received data that is transferred to RDR causes the CTSn_RTSn pin to output low.

Note 1. Only read data in the RDRHL register when 9-bit data length is selected.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 29.15 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 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 29.16 Example of SCI operation for serial reception in asynchronous mode (2) when the RTS function is used, and with 8-bit data, parity bit, and 1 stop bit

Table 29.23 lists the states of the flags in the SSR status register and receive data handling when a receive error is detected.

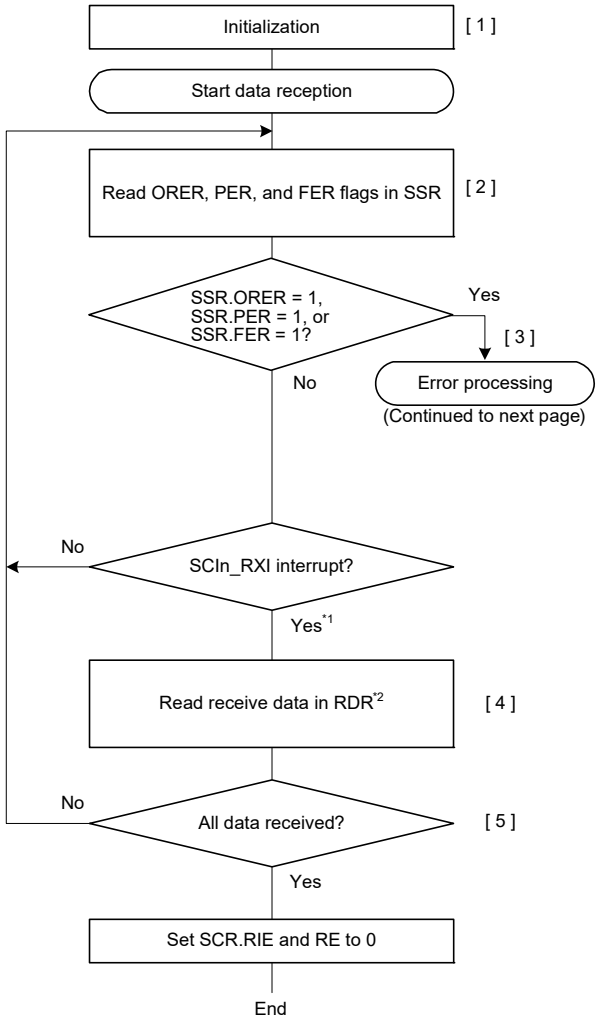
If a receive error is detected, an SCIn_ERI interrupt request is generated but an SCIn_RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER bits to 0 before resuming reception. In addition, be sure to read RDR or RDRHL 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 read might be left in the RDR or RDRHL.

Figure 29.17 and Figure 29.18 show example flows of serial data reception.

Table 29.23 Flags in SSR Status Register and receive data handling

Flags in the SSR Status Register				
ORER	**FER**	**PER**	**Receive data**	**Receive error type**
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR	Framing error
0	0	1	Transferred to RDR	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR	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. An error 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. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RXDn pin.

[4] Read the receive data in RDR once in the SCIn_RXI interrupt handling routine.

[5] Serial reception continuation procedure:
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 DMAC or DTC.

Note 1. Do not set 0 to RE before read RDR here.
Note 2. RDR becomes RDRHL when 9-bit data length is selected.

Figure 29.17 Example flow of serial reception in asynchronous mode with non-FIFO selected (1)

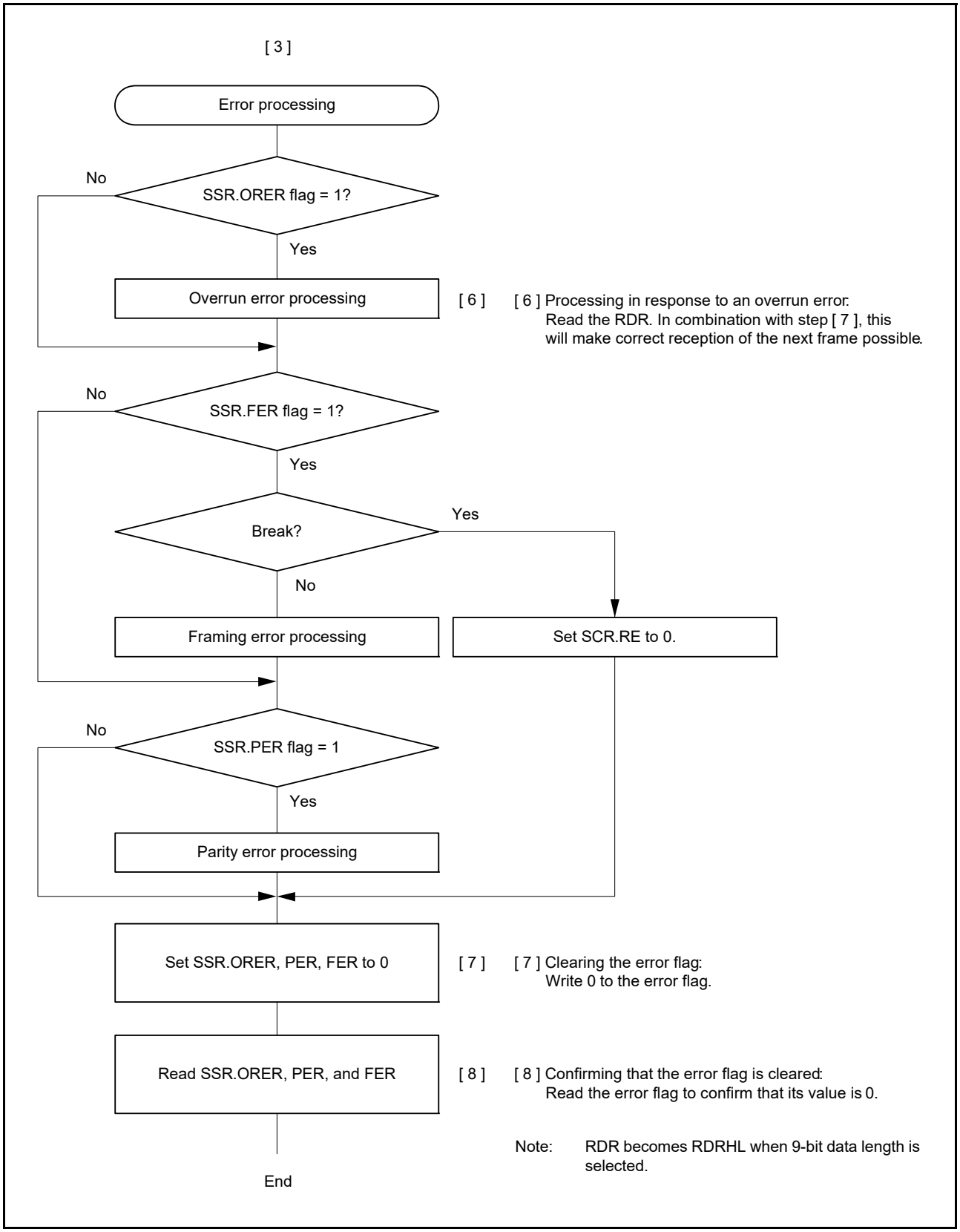


Figure 29.18 Example flow of serial reception in asynchronous mode with non-FIFO selected (2)

(2) FIFO selected

Figure 29.19 shows an example of a data format that is written to FRDRH and FRDRL in asynchronous mode.

In asynchronous mode, 0 is written to the MPB flag 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, 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 FRDRH 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	b7	b6	b5	b4	b3	b2	b1	b0	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								

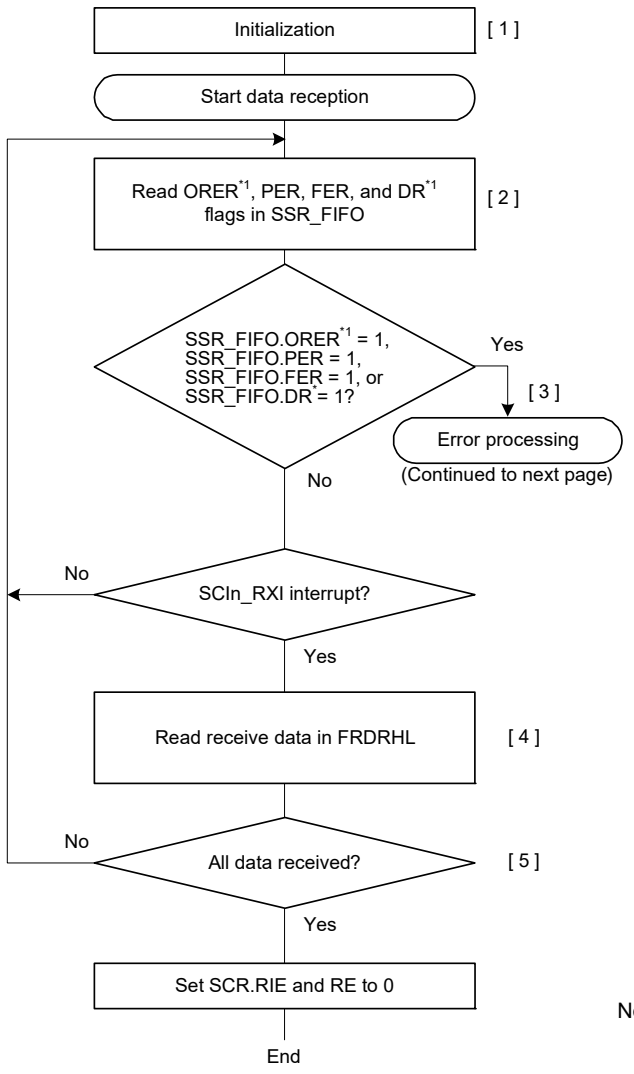
Note: 0 is always read for MPB flag (FRDRH[1]).
When data length is 7 bits, 0 is always read for FRDRH[0] and FRDRL[7].
When data length is 8 bits, 0 is always read for FRDRH[0].
FRDRH[7] bit is read as an indefinite value.

Figure 29.19 Data format stored to FRDRH and FRDRL with FIFO selected

In serial data reception, the SCI operates as follows:

1. When the value of the RE bit in SCR becomes 1, the output signal on the CTSn_RTSn pin goes low.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If an overrun error occurs, the ORER flag in SSR_FIFO is set to 1. When the RIE bit in SCR is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to FRDRL*1.
4. If a parity error is detected, the PER flag and receive data are transferred to FRDRL*1. When SCR.RIE is set to 1, an SCIn_ERI interrupt request is generated.
5. If a frame error is detected, the FER flag and receive data are transferred to FRDRL*1. When SCR.RIE is set to 1, an SCIn_ERI interrupt request is generated.
6. After a frame error is detected and when SCI detects that the continuous receive data is for one frame, reception stops.
7. When the amount of data stored in the receive FIFO data register (FRDRL) 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, SSR_FIFO.DR is set to 1. When SCR.RIE is 1 and the FCR.DRES bit is 0, SCI generates an SCIn_RXI interrupt request. When FCR.DRES is 1, SCI generates an SCIn_ERI interrupt request.
8. When reception finishes successfully, receive data is transferred to FRDRL*1. RDF is set to 1 when the amount of receive data written to FRDRHL is equal to or greater than the specified receive triggering number. When SCR.RIE is 1, an SCIn_RXI interrupt request is generated. Continuous 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 received data that is transferred to FRDRL*3 is less than the RTS trigger number, the CTSn_RTSn pin outputs low.

Note 1. Only read data in the FRDRH and FRDRL registers when 9-bit data length is selected.
Note 2. Read data in order from FRDRH to FRDRL when 9-bit data length is selected.
Note 3. 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 bit. 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 the 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 receive data in the FIFO by FDR.R bits.

[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 DMAC or DTC. The RDF flag is cleared automatically in this case, so do not write to the RDF flag.

Note 1. Can be read by the FRDRHL.ORER and DR flags.

Figure 29.20 Example flow of serial reception in asynchronous mode with FIFO selected (1)

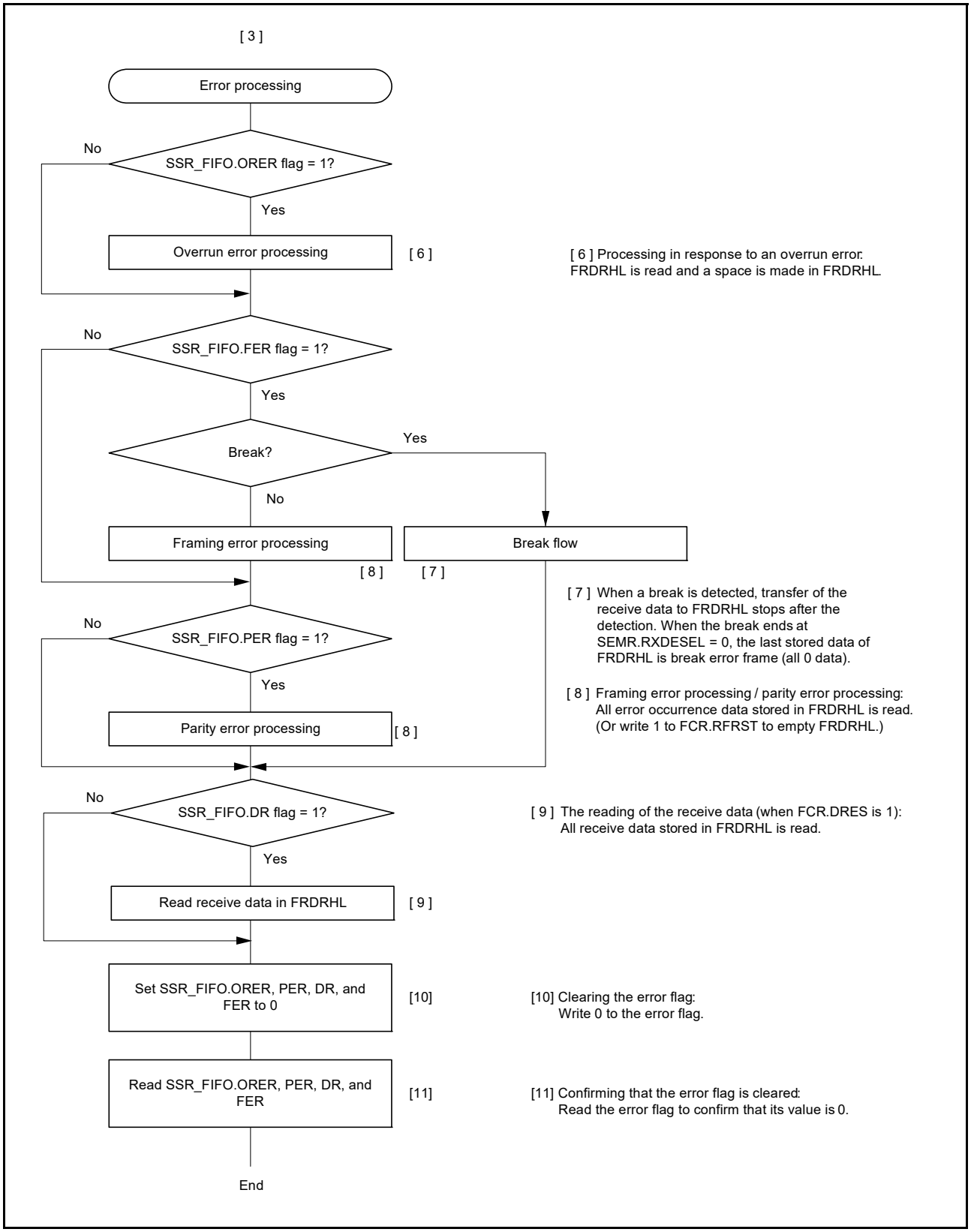


Figure 29.21 Example flow of serial reception in asynchronous mode with FIFO selected (2)

29.4 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 29.22 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 the data again in which the multi-processor bit is set to 1.

(1) Non-FIFO selected

To support this function, the SCI provides the MPIE bit in SCR. When MPIE 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 RSR to RDR (RDRHL when 9-bit data length is selected)
- Detection of a receive error
- Setting of the respective status flags RDRF, ORER, and FER in SSR.

On receiving a reception character in which the multi-processor bit is set to 1, the MPBT bit in SSR is set to 1 and the MPIE bit in SCR is automatically cleared, returning the SCI to a non-multi-processor reception operation. During this time, SCIn_RXI interrupt is generated if the RIE bit in SCR is set.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in non-multi-processor asynchronous mode. The clock used for the multi-processor communication is the same as the clock used in non-multi-processor asynchronous mode.

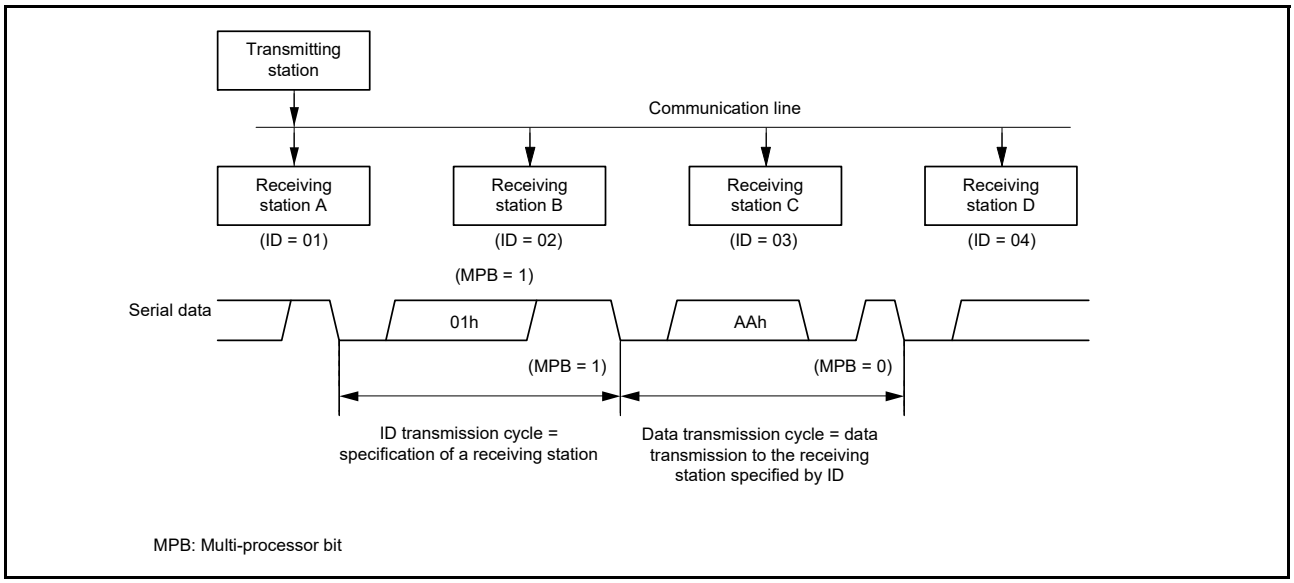


Figure 29.22 Example of communication using multi-processor format with transmission of data AAh to receiving station A

(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 FRDRHL.MPB and receive data is written to FRDRL.

When the MPIE bit is set to 1, the following functions are disabled until reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from RSR to FRDRHL
- Detection of a receive error
- Break
- Setting of the respective status flags RDF, ORER, and FER in SSR_FIFO.

On receiving an 8-bit character in which the multi-processor bit is set to 1, the MPB bit in FRDRHL is set to 1 and receive data is written to FRDRHL.RDAT. The MPIE bit in SCR is automatically cleared, therefore returning the SCI to non-multi-processor reception operation. An SCIn_RXI interrupt is generated if the RIE bit in SCR is set.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in non-multi-processor asynchronous mode with FIFO selected.

29.4.1 Multi-Processor Serial Data Transmission

(1) Non-FIFO selected

Figure 29.23 shows an example flow of multi-processor data transmission. In the ID transmission cycle, the ID must be transmitted with the MPBT bit in SSR 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.

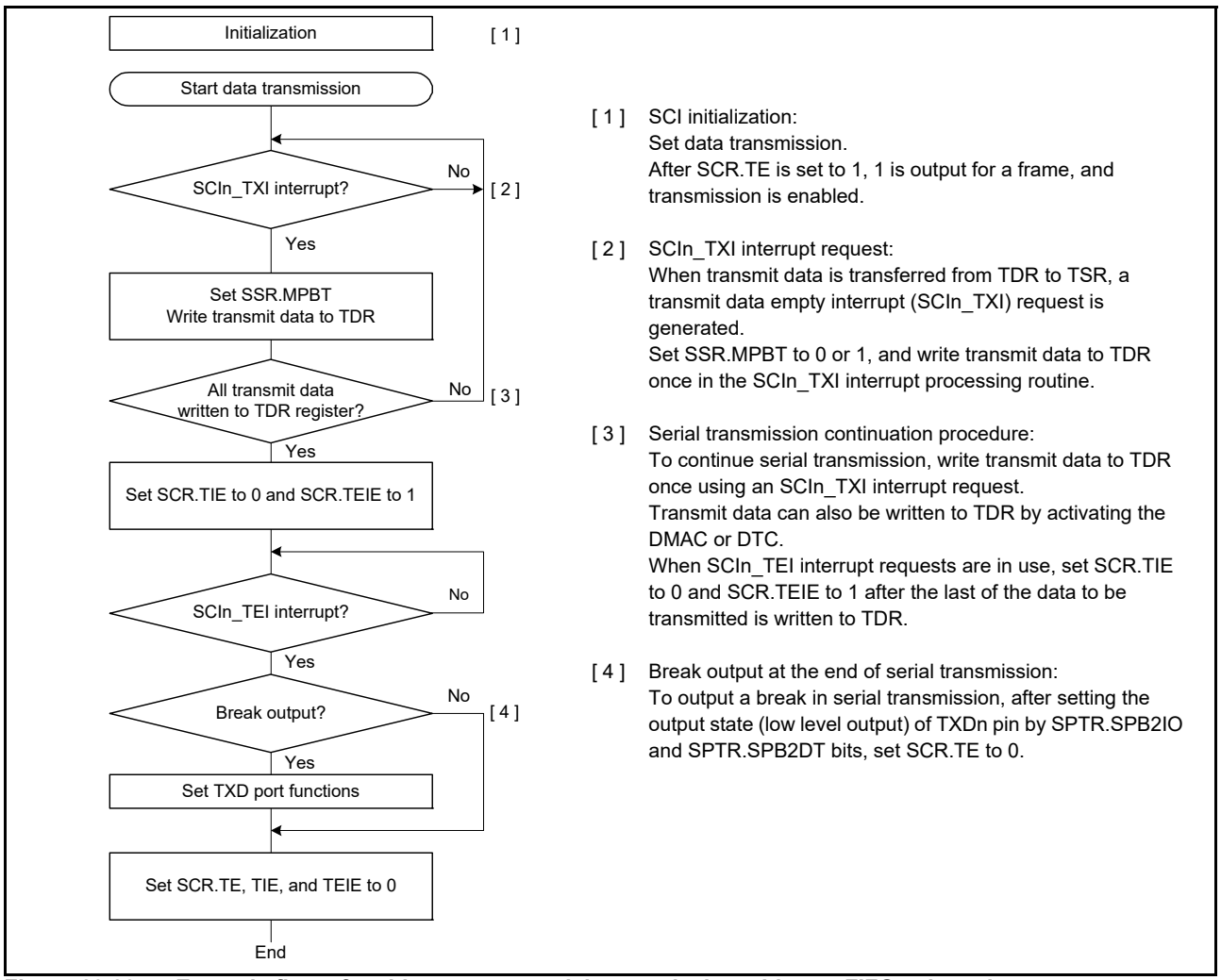


Figure 29.23 Example flow of multi-processor serial transmission with non-FIFO selected

(2) FIFO selected

Figure 29.24 shows an example of data format that is written to FTDRH and FTDRL in multi-processor mode.

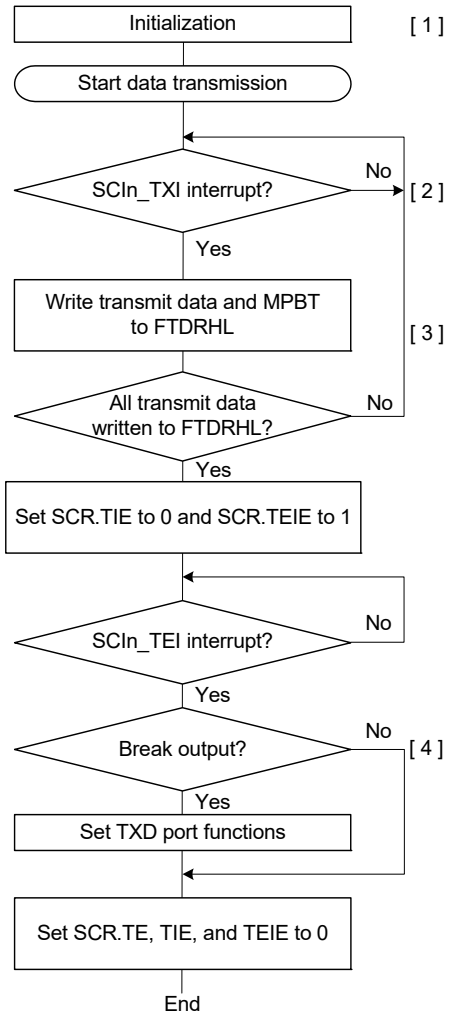
MPBT is set to 1 in FTDRH. Data is set to FTDRH and FTDRL with the correct data length. Write 0 for unused bits. Write in order from FTDRH to FTDRL.

Data Length	Register Setting		Transmit data in FTDRH, FTDRL															
			FTDRHL															
			FTDRH								FTDRL							
	SCMR.CHR1	SMR.CHR	b7	b6	b5	b4	b3	b2	b1	b0	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								

—: Invalid. The write value should be 0.

Figure 29.24 Data format written to FTDRH and FTDRL in multi-processor mode with FIFO selected

Figure 29.25 shows an example flow of multi-processor data transmission with FIFO selected. In the ID transmission cycle, the ID must be transmitted with the MPBT bit in FTDRH 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 non-FIFO selected.



[1] SCI initialization:
Set data transmission.
After SCR.TE 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 amount 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 DMAC or DTC.
When SCIn_TEI interrupt requests are in use, set SCR.TIE to 0 and SCR.TEIE 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 TXDn pin by SPTR.SPB2IO and SPTR.SPB2DT bits, set SCR.TE to 0.

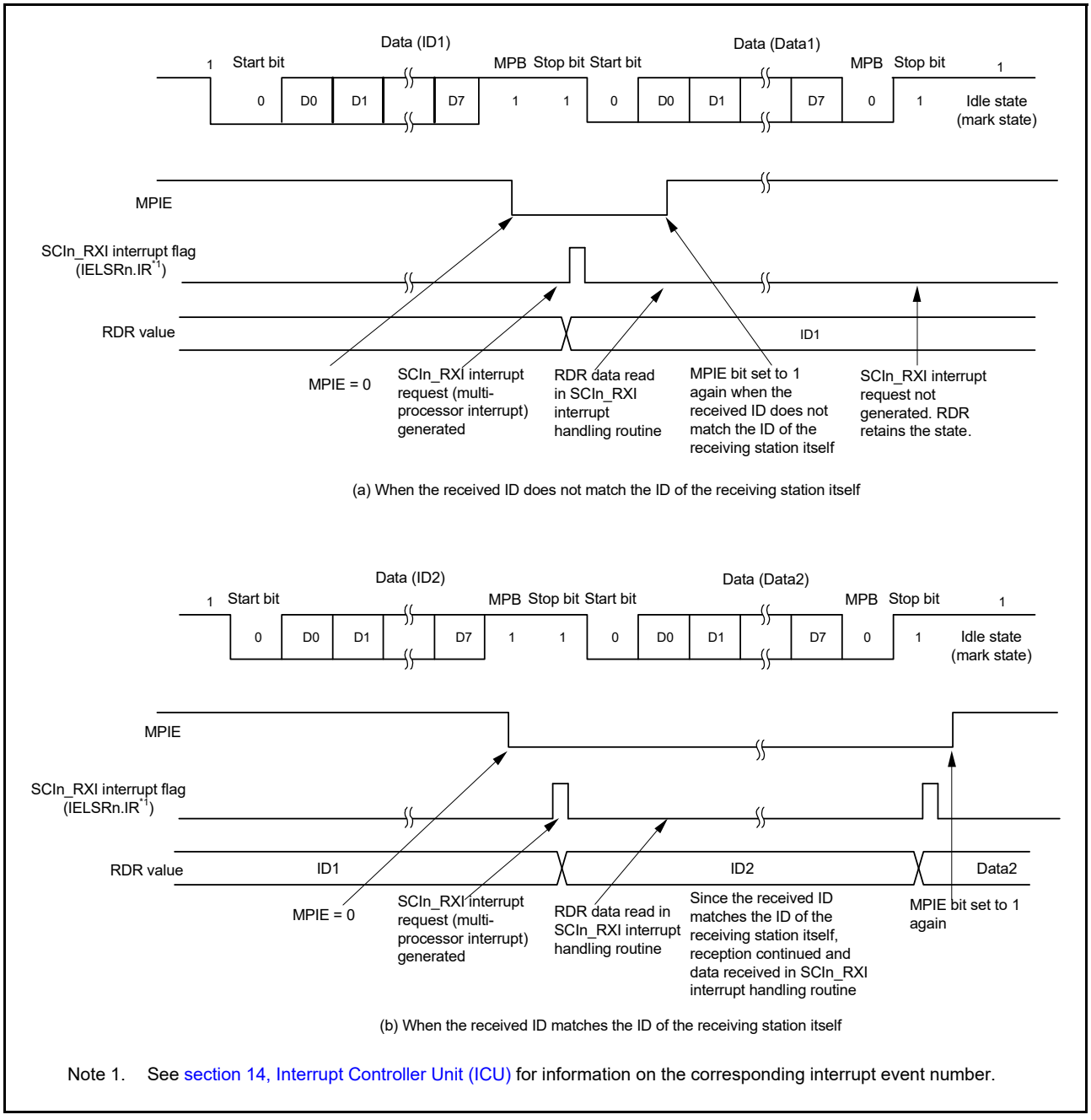
Figure 29.25 Example flow of serial transmission in multi-processor mode with FIFO selected

29.4.2 Multi-Processor Serial Data Reception

(1) Non-FIFO selected

Figure 29.27 and Figure 29.28 are example flows of multi-processor data reception. When the MPIE bit in SCR 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 RDR (RDRHL 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.

Figure 29.26 shows an example operation for data reception.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 29.26 Example of SCI reception with 8-bit data, multi-processor bit, and 1 stop bit

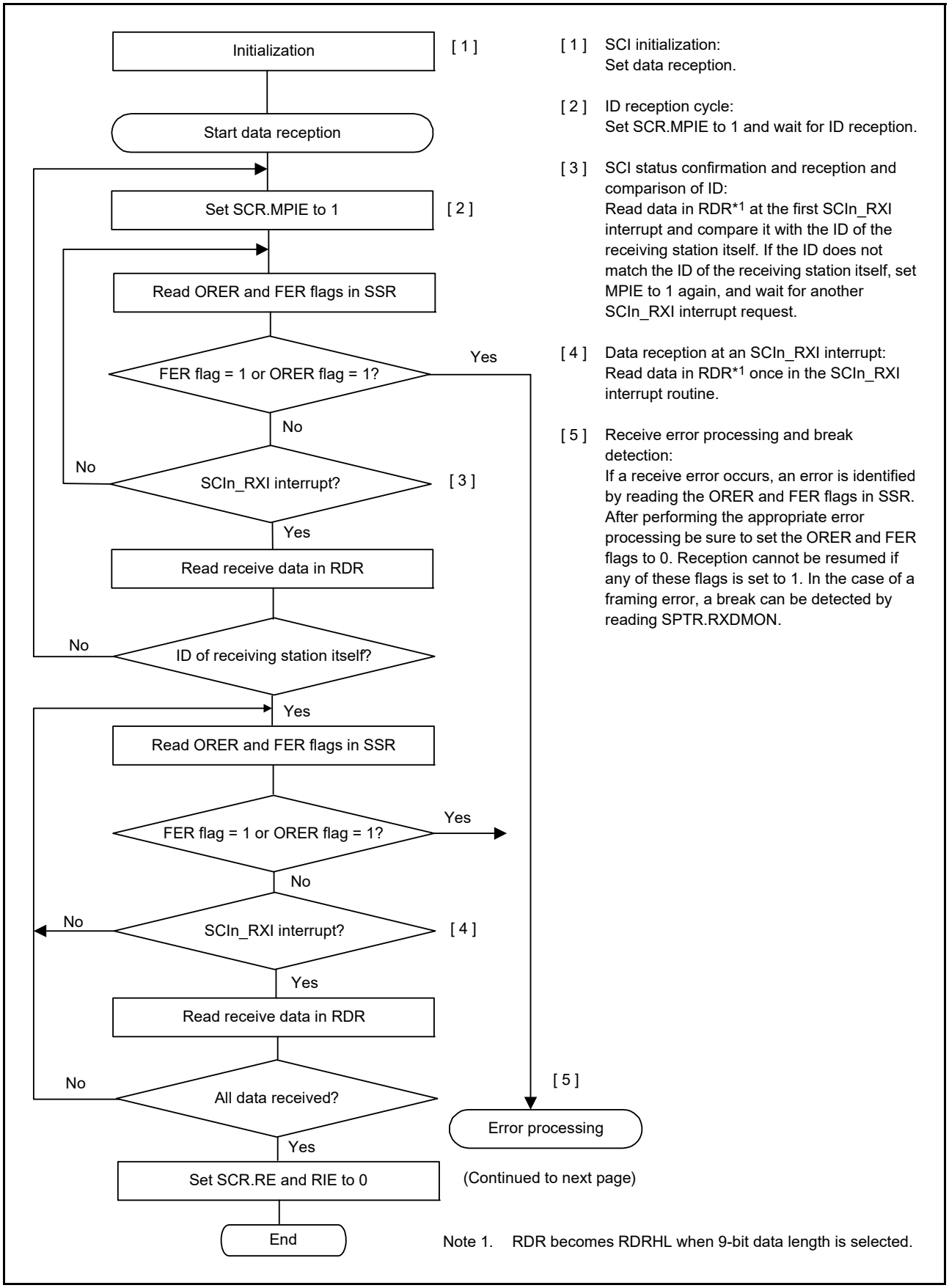


Figure 29.27 Example flow of multi-processor serial reception with non-FIFO selected (1)

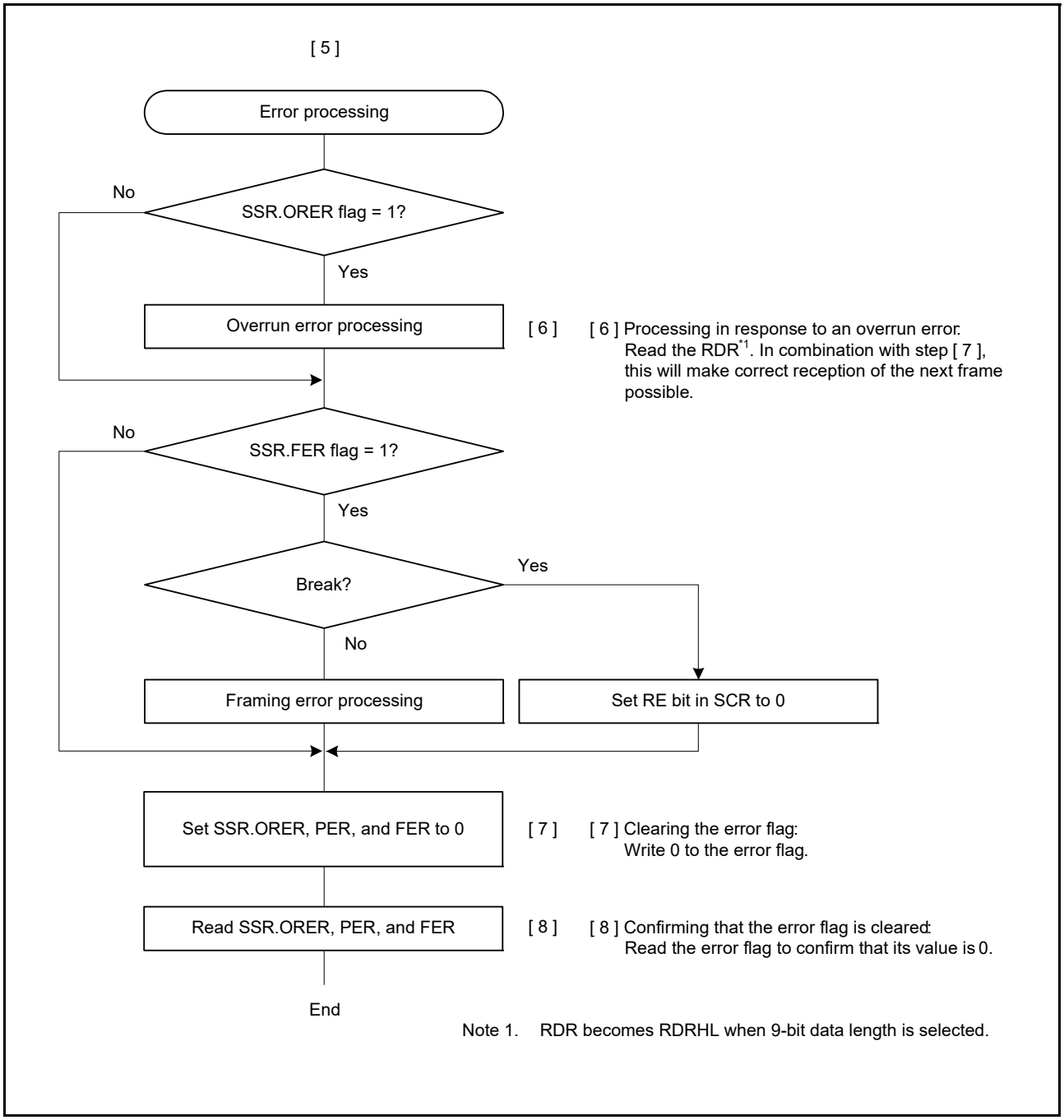


Figure 29.28 Example flow of multi-processor serial reception with non-FIFO selected (2)

(2) FIFO selected

Figure 29.29 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 MPB flag in FRDRH. A value of 0 is written to the PER flag in FRDRH. 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. If software reads FRDRL, the SCI updates FER, MPB and receive data (RDAT[8:0]) in FRDRL with the next data. The flags RDF, ORER and DR in FRDRH always reflect the associated flags in the SSR_FIFO register.

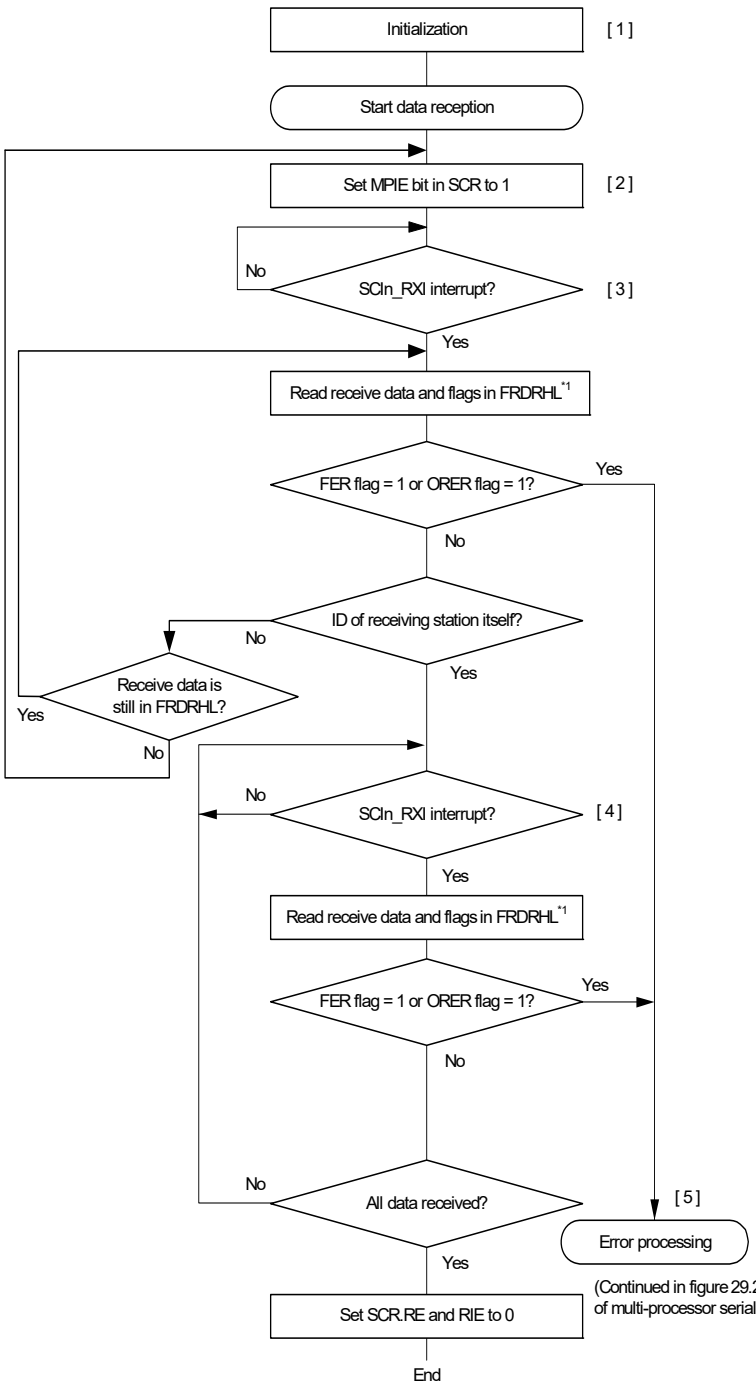
Data Length	Register Setting		Receive data in FRDRH, FRDRL															
			FRDRHL															
			FRDRH								FRDRL							
	SCMR. CHR1	SMR. CHR	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	0	—	RDF	ORER	FER	0	DR	MPB	0	0	7-bit receive data						
8 bits	1	1	—	RDF	ORER	FER	0	DR	MPB	0	8-bit 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]
FRDRH[7] bit is read as an indefinite value.

Figure 29.29 Data format stored to FRDRH and FRDRL in multi-processor mode with FIFO selected

Figure 29.30 shows a sample flowchart for multi-processor data reception with FIFO selected. When the MPIE bit in SCR 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 FRDRHL. The MPIE bit in SCR is automatically cleared and non-multi-processor reception continues.

If a frame error occurs and SSR_FIFO.FER 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 SCR.MPIE to 1 and wait for ID reception.

[3] SCI status confirmation and reception and comparison of ID:
SCI stores first data (MPB = 1), after which all received data are stored in FRDRHL.
RDF is set to 1 and an SCIn_RXI interrupt request is generated when the amount of receive data equal to or greater than the specified receive triggering number is stored in FRDRHL.
When the amount of data stored in the receive FIFO data register (FRDRHL) falls below the specified receive triggering number and received data is equal to or greater than 1, and no new 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, read until the data with MPB = 1, and compares next ID. If it is not a data with MPB = 1 in FRDRHL, set MPIE 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 SSR_FIFO.ORER and SSR_FIFO.FER to 0. Reception cannot be resumed if SSR_FIFO.ORER is set to 1. In the case of a framing error, a break can be detected by reading SPTR.RXDMON.

Note 1. If FRDRH and FRDRL are used instead of FRDRHL, read in order from FRDRH to FRDRL.

Figure 29.30 Example flow of serial reception in multi-processor mode with FIFO selected

29.5 Operation in Clock Synchronous Mode

Figure 29.31 shows the data format for clock synchronous serial data communications.

In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. 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. 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 SPMR.CKPH is 1 in slave mode, the SCI holds the first bit output state.

Within the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by using a common clock. Both the transmitter and the receiver 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 do continuous transfer in the fastest bit rate setting (BRR = 00h and SMR.CKS[1:0] = 00b), therefore when the FIFO is selected, this setting (BRR = 00h and SMR.CKS[1:0] = 00b) is not available.

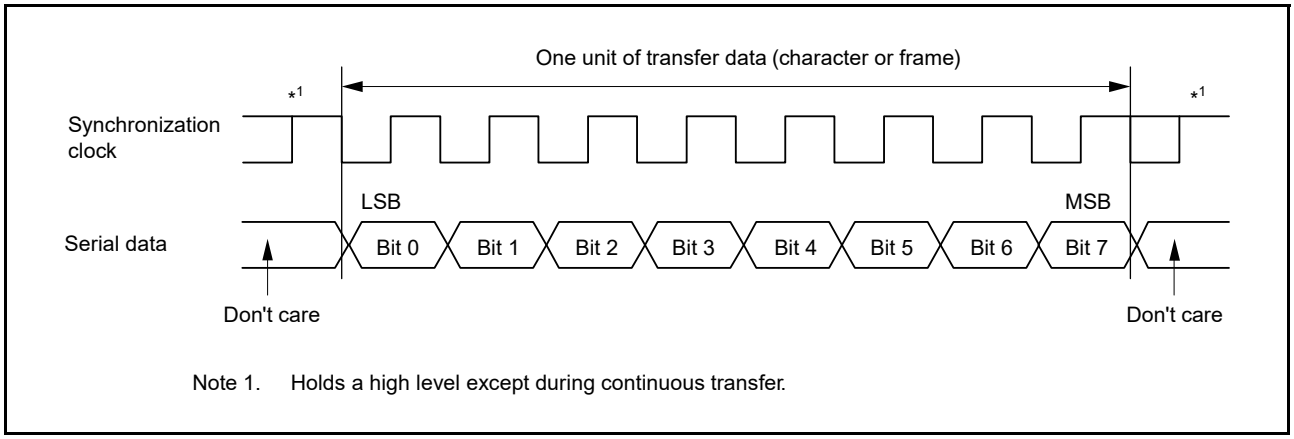


Figure 29.31 Data format in clock synchronous serial communications (LSB first)

29.5.1 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 and the CTS function is disabled, the synchronization clock output starts when SCR.RE is set to 1. The synchronization clock stops when it is held high*1 and when an overrun error occurs, or when 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 SCR.RE is set to 1 and the CTSn_RTSn input is high. The synchronization clock output starts when SCR.RE is set to 1 and the CTSn_RTSn input is low. When the CTSn_RTSn input is high on completion of the frame reception, the synchronization clock output stops when it goes high. If the CTSn_RTSn input continues to be low, the synchronization clock stops when it is held high*1 and when an overrun error occurs, or when the SCR.RE bit is set to 0.

Note 1. The signal is held high while (SPMR.CKPH bit = 0 && SPMR.CKPOL bit = 0) or (SPMR.CKPH bit = 1 && SPMR.CKPOL bit = 1). It is held low while (SPMR.CKPH bit = 0 && SPMR.CKPOL bit = 1) or (SPMR.CKPH bit = 1 && SPMR.CKPOL bit = 0).

29.5.2 CTS and RTS Functions

In the CTS function, the CTSn_RTSn input controls the start of data reception or transmission when the clock source is the internal clock. Setting SPMR.CTSE 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 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 becomes possible. Conditions for output of CTSn_RTSn low and high are as follows:

[Conditions for low output]

(a) Non-FIFO selected, when all of the following conditions are satisfied

- The value of the RE or TE bit in SCR is 1
- When serial communication is enabled
- There is no received data available to be read when SCR.RE is 1
- Data is available for transmission in the TSR register when SCR.TE is 1
- The ORER flag in SSR is 0.

(b) FIFO selected, when all of the following conditions are satisfied

- The value of the RE or TE bit in the SCR is 1
- When serial communication is enabled
- When the amount of receive data written in FRDRHL is less than the specified CTSn_RTSn output triggering number (when SCR.RE = 1)
- Data that has not been transmitted is available in FTDRHL (when SCR.TE is 1 and SCR.CKE[1] is 0)
- Data is available for transmission in TSR (when SCR.TE is 1 and SCR.CKE[1] is 1
- The ORER in the SSR_FIFO is 0.

[Condition for high output]

(a) Non-FIFO selected

- The conditions for low output are not satisfied.
- When reception is terminated with SCR.RE = 0 without reading the RDR register after reception is completed, RTS remains high. At this time, read the SCR register for dummy after writing SCR.RE = 0.

(b) FIFO selected

- The conditions for low output are not satisfied.

29.5.3 SCI Initialization in Clock Synchronous Mode

Before transmitting and receiving data, start by writing the initial value 00h to SCR, then continue through the SCI procedure in 29.5.2 CTS and RTS Functions. Any time the operating mode or transfer format is to be changed, SCR must be initialized before the change can be made.

Note: When the SCR.RE bit is set to 0, the ORER, FER, RDRF, RDF, PER, and DR flags in SSR/SSR_FIFO, and the RDR and RDRHL registers are not initialized. When the SCR.TE bit is set to 0, the TEND flag for the selected FIFO buffer is not initialized.

Note: 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.

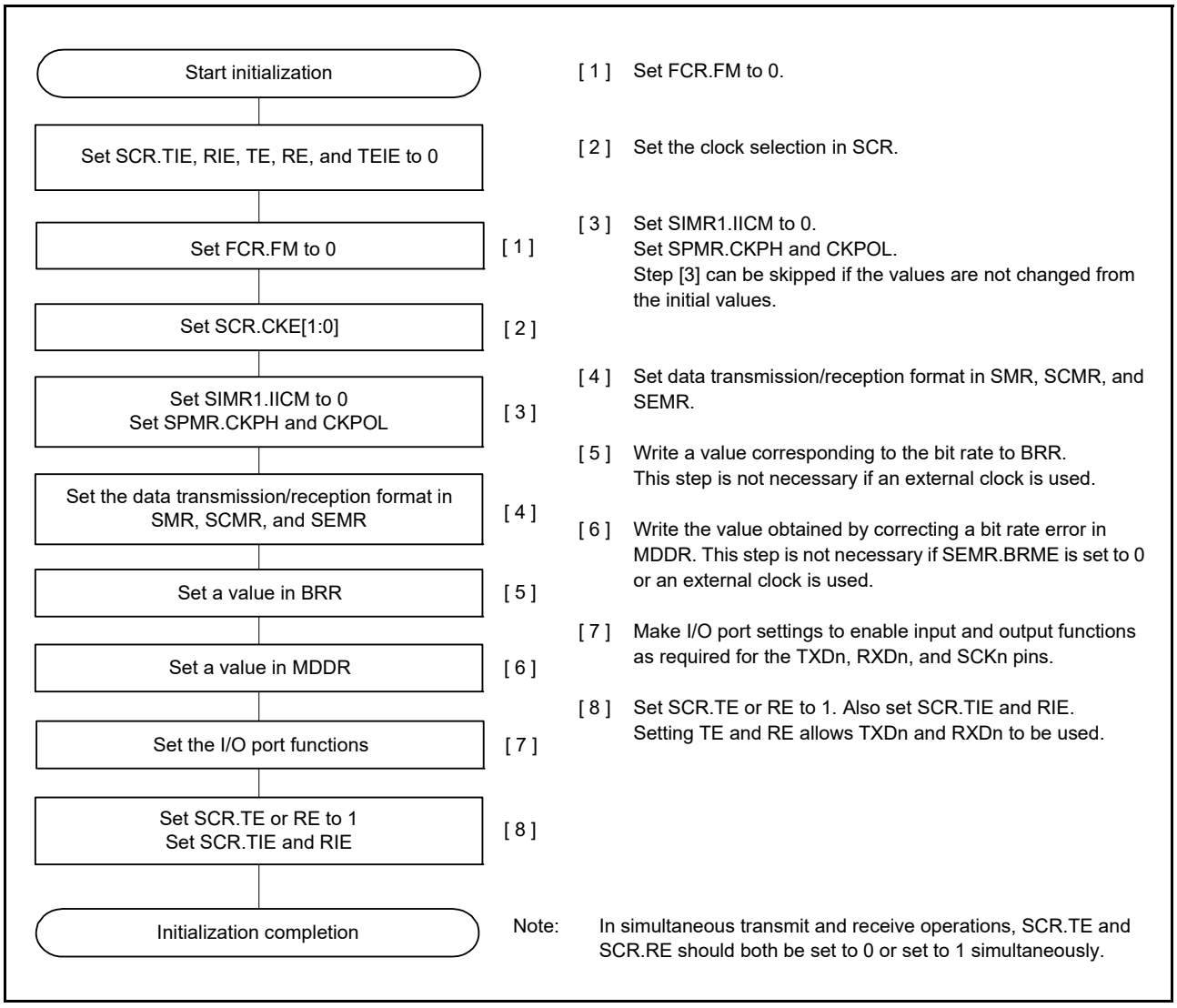


Figure 29.32 Example flow of SCI initialization in clock synchronous mode with non-FIFO selected

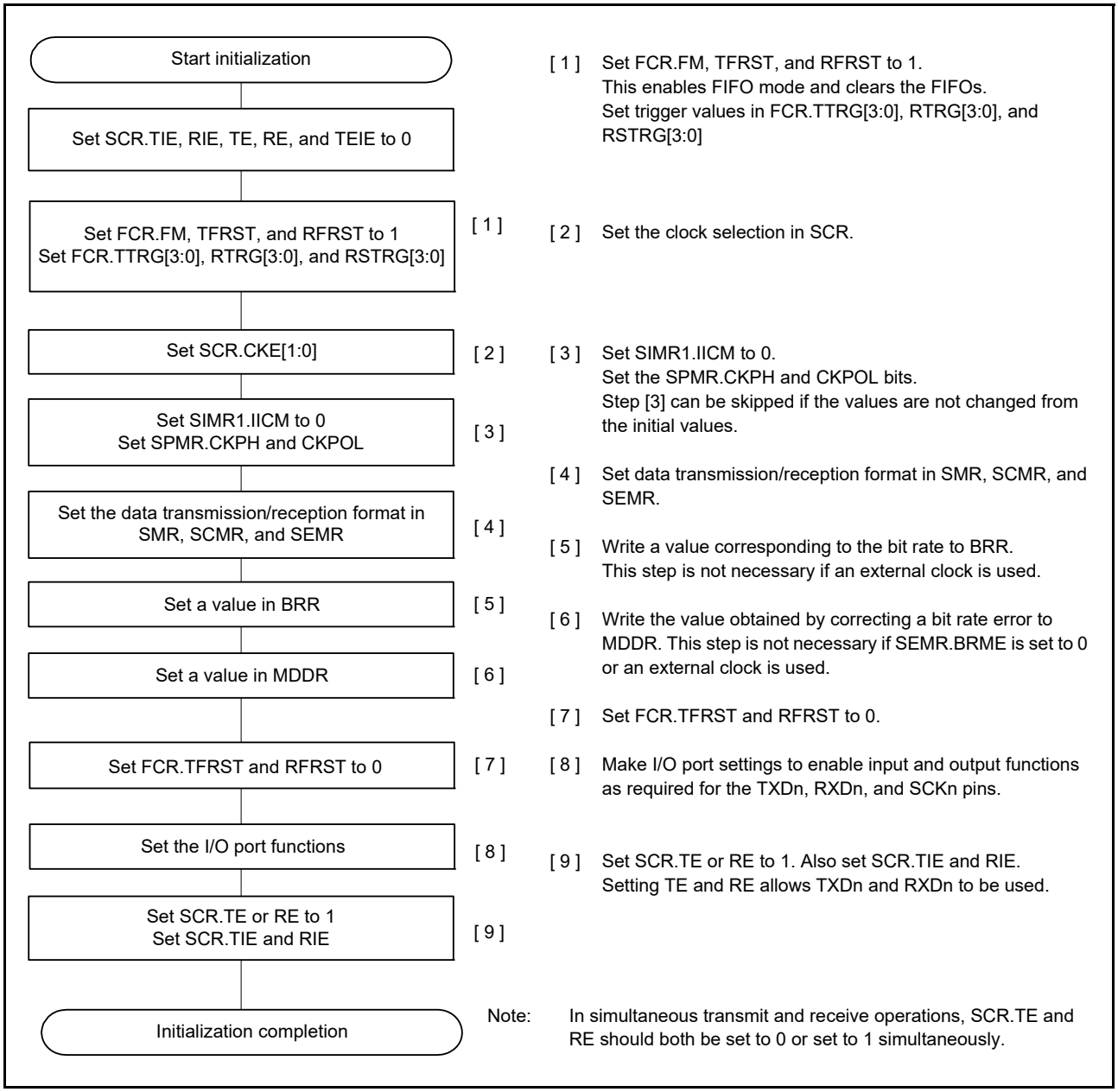


Figure 29.33 Example flow of SCI initialization in clock synchronous mode with FIFO selected

29.5.4 Serial Data Transmission in Clock Synchronous Mode

(1) Non-FIFO selected

Figure 29.34, Figure 29.35, and Figure 29.36 show examples of serial transmission in clock synchronous mode.

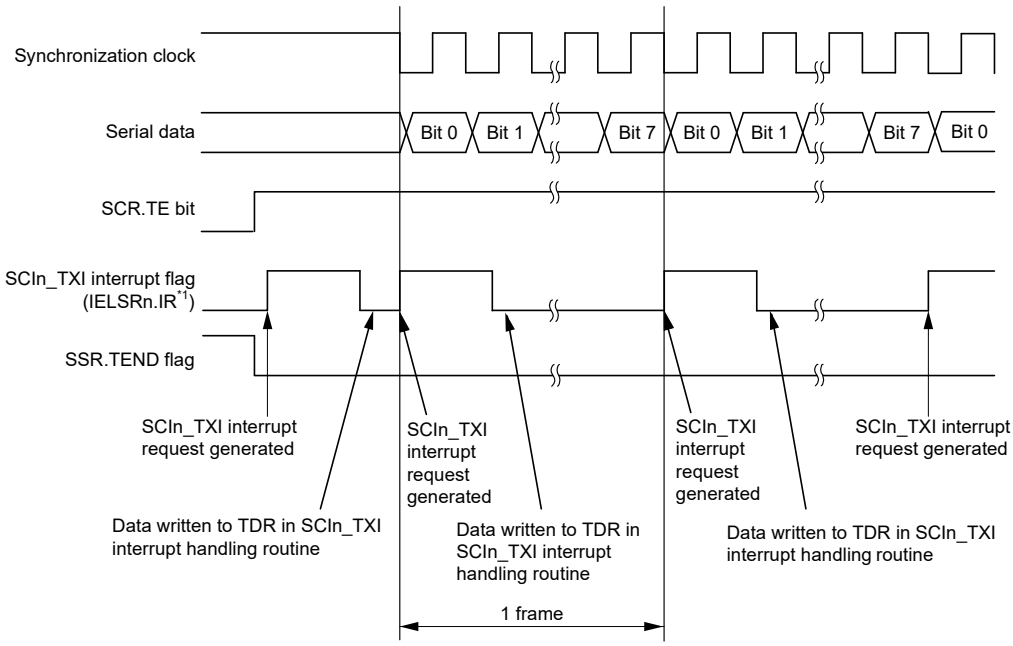
In serial data transmission, the SCI operates as follows:

1. The SCI transfers data from TDR to TSR 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 SCR.TE is set to 1 but only after SCR.TIE is also set to 1, or when SCR.TE and SCR.TIE are both set to 1 simultaneously by a single instruction.
2. After transferring data from TDR to TSR, the SCI starts transmission. When SCR.TIE 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 SCR.TIE to 0 and SCR.TEIE to 1 after the last of the data to be transmitted is written to TDR.
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 CTSn_RTSn input signal is low when SPMR.CTSE is 1.
4. The SCI checks for update to TDR on output of the last bit.
5. When TDR 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 SCR.TEIE is 1, an SCIn_TEI interrupt request is generated and the SCKn pin is held high.

Figure 29.34, Figure 29.35, and Figure 29.36 show example flows of serial data transmission.

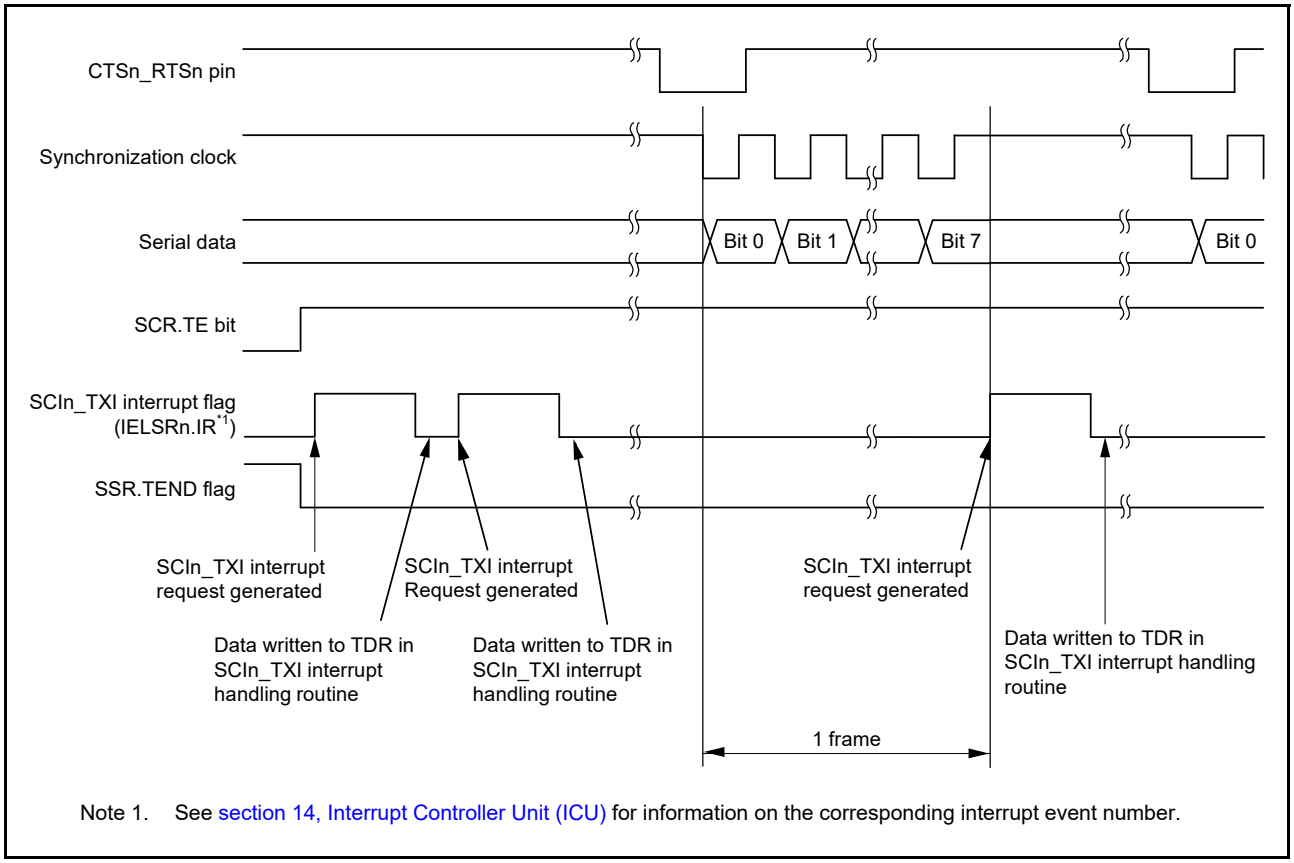
Transmission does not start while a receive error flag (ORER, FER, or PER in SSR) is set to 1. Be sure to set the receive error flags to 0 before starting transmission.

Note: Setting SCR.RE to 0 does not clear the receive error flags.



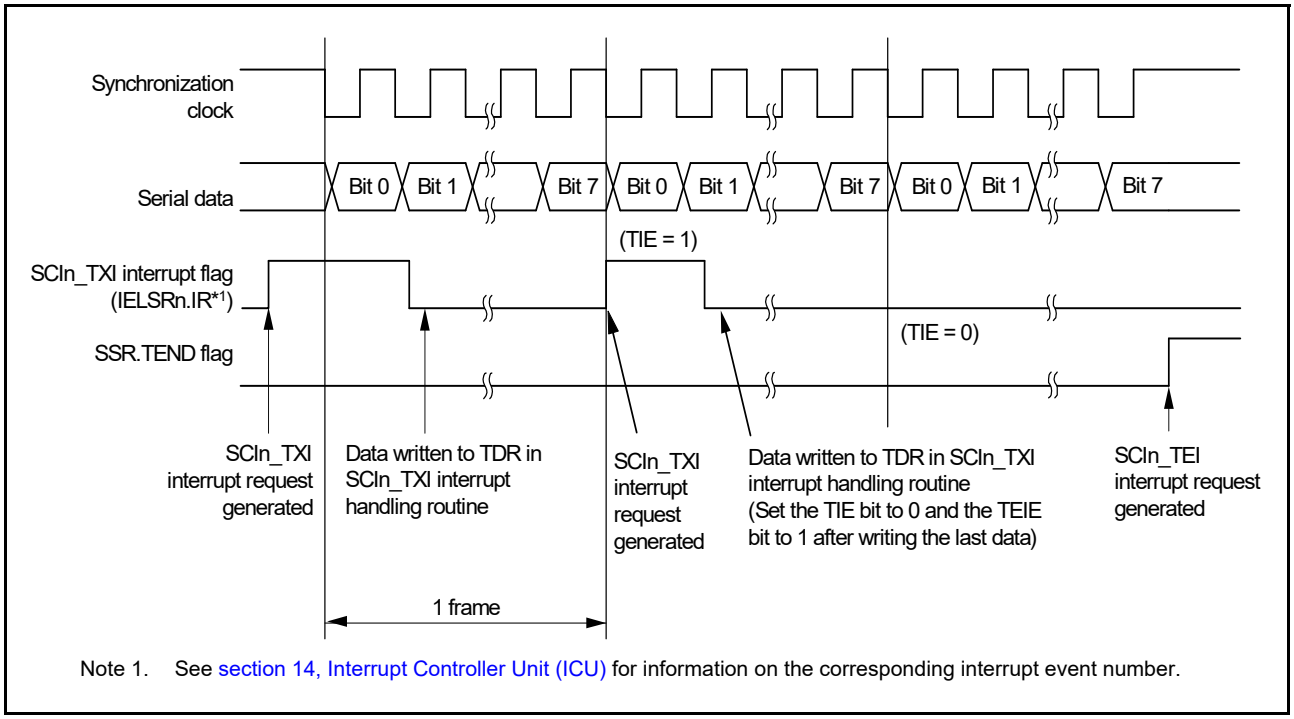
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 29.34 Example of serial data transmission in clock synchronous mode when the CTS function is not used at the beginning of transmission



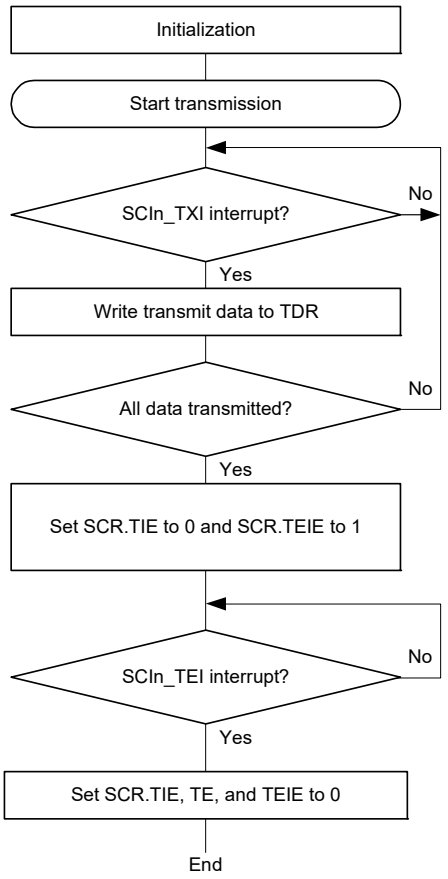
Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 29.35 Example of serial data transmission in clock synchronous mode when the CTS function is used at the beginning of transmission



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 29.36 Example of serial data transmission in clock synchronous mode from the middle of transmission until transmission completion



[1] SCI initialization:
Set data transmission.

[2] Writing transmit data 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] 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 DMAC or DTC by the SCIn_TXI interrupt request.
When SCIn_TEI interrupt requests are in use, set SCR.TIE to 0 and SCR.TEIE to 1 after the last of the data to be transmitted is written to TDR.

Note: When the external clock is in use (the value of SCR.CKE[1:0] is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR.TEND flag to 1. Setting SCR.TE to 0 immediately after this may lead to insufficient hold time for received data on the receiver side.

Figure 29.37 Example flow of serial transmission in clock synchronous mode with non-FIFO selected

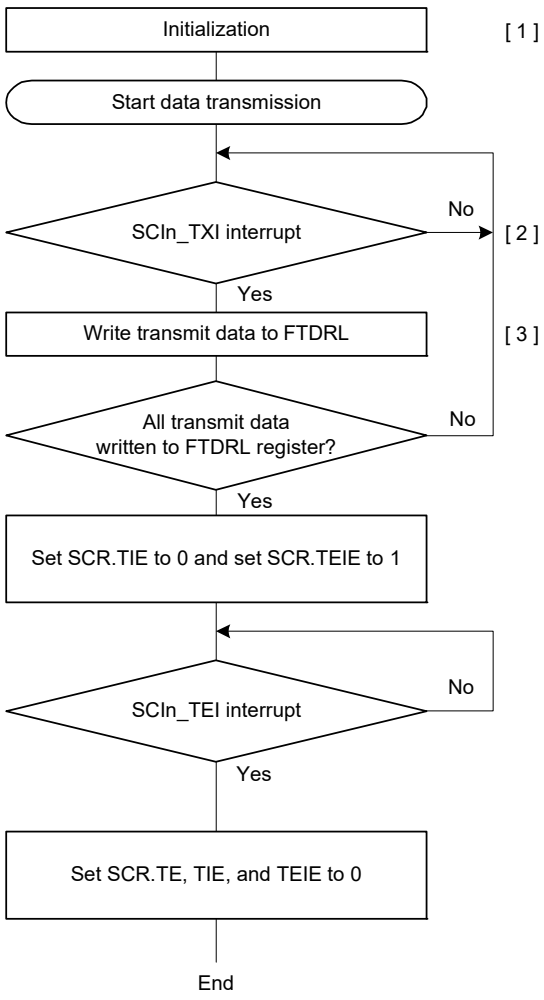
(2) FIFO selected

Figure 29.38 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 FTDRL*1 to TSR 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 SCR.TE is set to 1, but only after SCR.TIE is also set to 1, or when SCR.TE and SCR.TIE are both 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 flag 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 finishes. When SCIn_TEI interrupt requests are in use, set SCR.TIE to 0 and SCR.TEIE to 1 after the last of the data to be transmitted is written to FTDRL.
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 use of an external clock is specified. Output of the clock signal is suspended until the CTSn_RTSn input signal is low and SPMR.CTSE is 1.
4. The SCI checks whether non-transmitted data remains in FTDRL at the time of the 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 SCR.EIE 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]

[2]

[3]

[1] SCI initialization:
Set data transmission.
After SCR.TE 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: when transmit data is transferred from FTDRL to TSR, and the amount of transmit data written in FTDRL is less than or equal to 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 the 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 DMAC or 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 SCR.TIE to 0 and SCR.TEIE to 1 after the last of the data to be transmitted is written to FTDRL.

Note: When the external clock is in use (the value of SCR.CKE[1:0] is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR_FIFO.TEND flag to 1. Setting SCR.TE to 0 immediately after this may lead to insufficient hold time for receive data on the receiver side.

Figure 29.38 Example flow of serial transmission in clock synchronous mode with FIFO selected

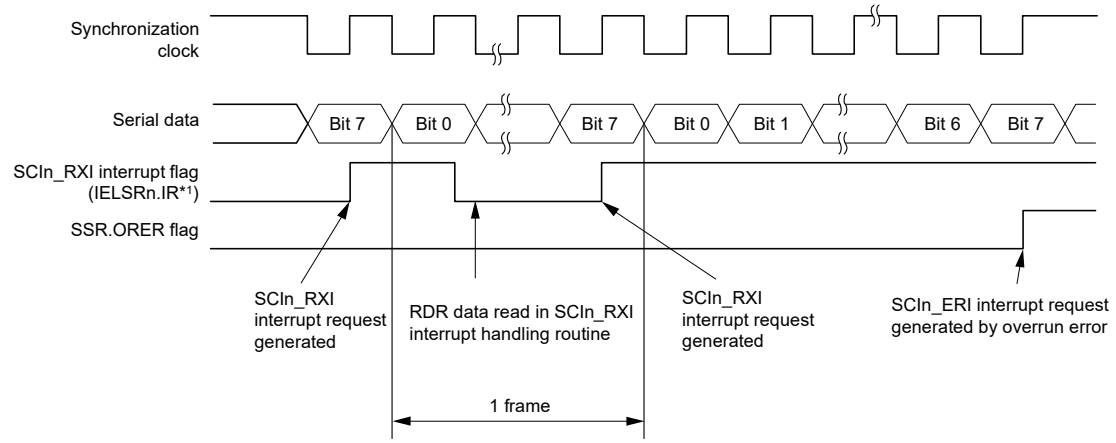
29.5.5 Serial Data Reception in Clock Synchronous Mode

(1) Non-FIFO selected

Figure 29.39 and Figure 29.40 show examples of SCI operation for serial reception in clock synchronous mode.

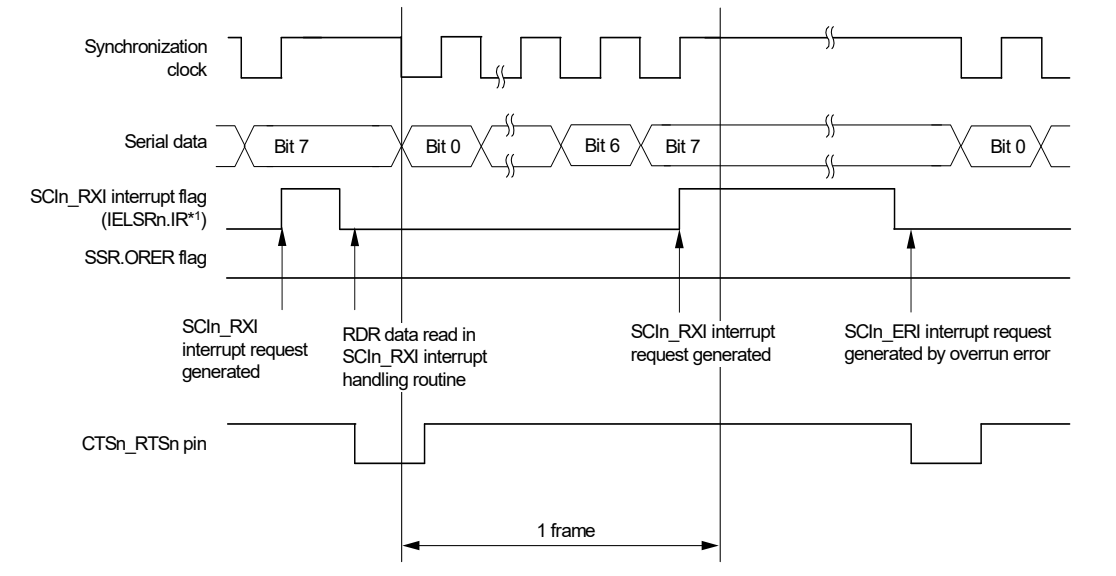
In serial data reception, the SCI operates as follows:

1. When the value of SCR.RE 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 RSR.
3. If an overrun error occurs, the SSR.ORER flag is set to 1. If SCR.RIE is 1, an SCIn_ERI interrupt request is generated. Receive data is not transferred to RDR.
4. When reception completes successfully, receive data is transferred to RDR. If SCR.RIE is 1, an SCIn_RXI interrupt request is generated. Continuous reception is enabled by reading the received data transferred to RDR in the SCIn_RXI interrupt handling routine before reception of the next receive data completes. Reading the received data from RDR causes the CTSn_RTSn pin to output low.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 29.39 Example operation for serial reception in clock synchronous mode (1) when RTS function is not used

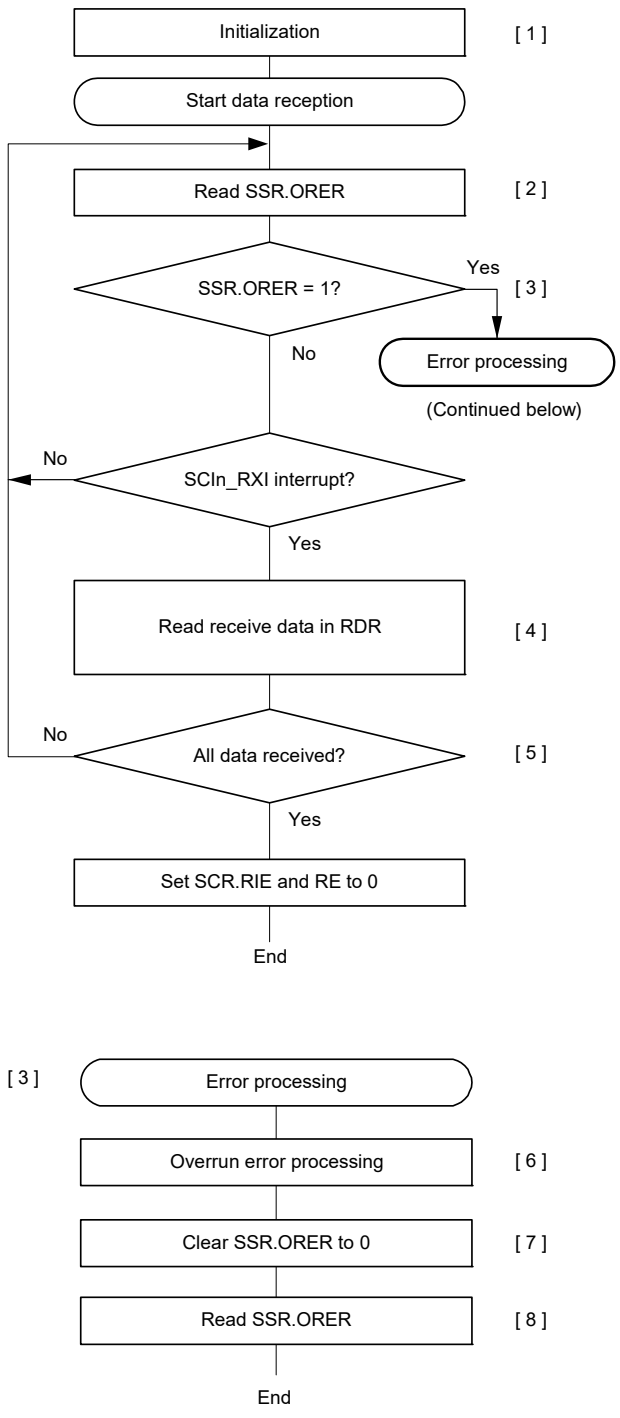


Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 29.40 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 bits in SSR to 0 before resuming data reception. Additionally, be sure to read the RDR during overrun error processing. When a data reception is forced to terminate by setting SCR.RE to 0 during operation, read the RDR because received data that is not yet read might be left in the RDR.

Figure 29.41 shows an example flow of serial data reception.



[1] SCI initialization:
Make input port settings for pins to be used as RXDn pins.

[2] [3] Receive error processing:
If a receive error occurs, read SSR.ORER, perform the relevant error processing, then set SSR.ORER to 0. Data reception cannot be resumed while SSR.ORER is 1.

[4] Read the receive data in RDR once in the receive data full interrupt (SCIn_RXI) request handling routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in RDR. The RDR data can also be read by activating the DMAC or DTC by an SCIn_RXI interrupt request.

[6] Processing in response to an overrun error:
Read RDR. In combination with step [7], this makes 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.

Figure 29.41 Example flow of serial reception in clock synchronous mode with non-FIFO selected

(2) FIFO selected

Figure 29.42 shows an example of serial reception in clock synchronous mode with FIFO operation selected. In serial data reception, the SCI operates as follows:

1. When the value of SCR.RE 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 RSR.
3. If an overrun error occurs, the ORER flag in SSR_FIFO is set to 1. If SCR.RIE is 1, an SCIn_ERI interrupt request is generated. Received data is not transferred to FRDRL*1.
4. When data reception completes successfully, the receive data is transferred to FRDRL*1. SSR_FIFO.RDF is set to 1 when the amount of the receive data is equal to or greater than the specified receive triggering number stored in FRDRHL. If SCR.RIE 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 has been transferred to FRDRL is less than the RTS trigger number, the CTSn_RTSn pin goes low.

Note 1. In clock synchronous mode, FTDRH is not used.
Note 2. Read data in order from FRDRH to FRDRL when RDF and ORER are read with receive data.

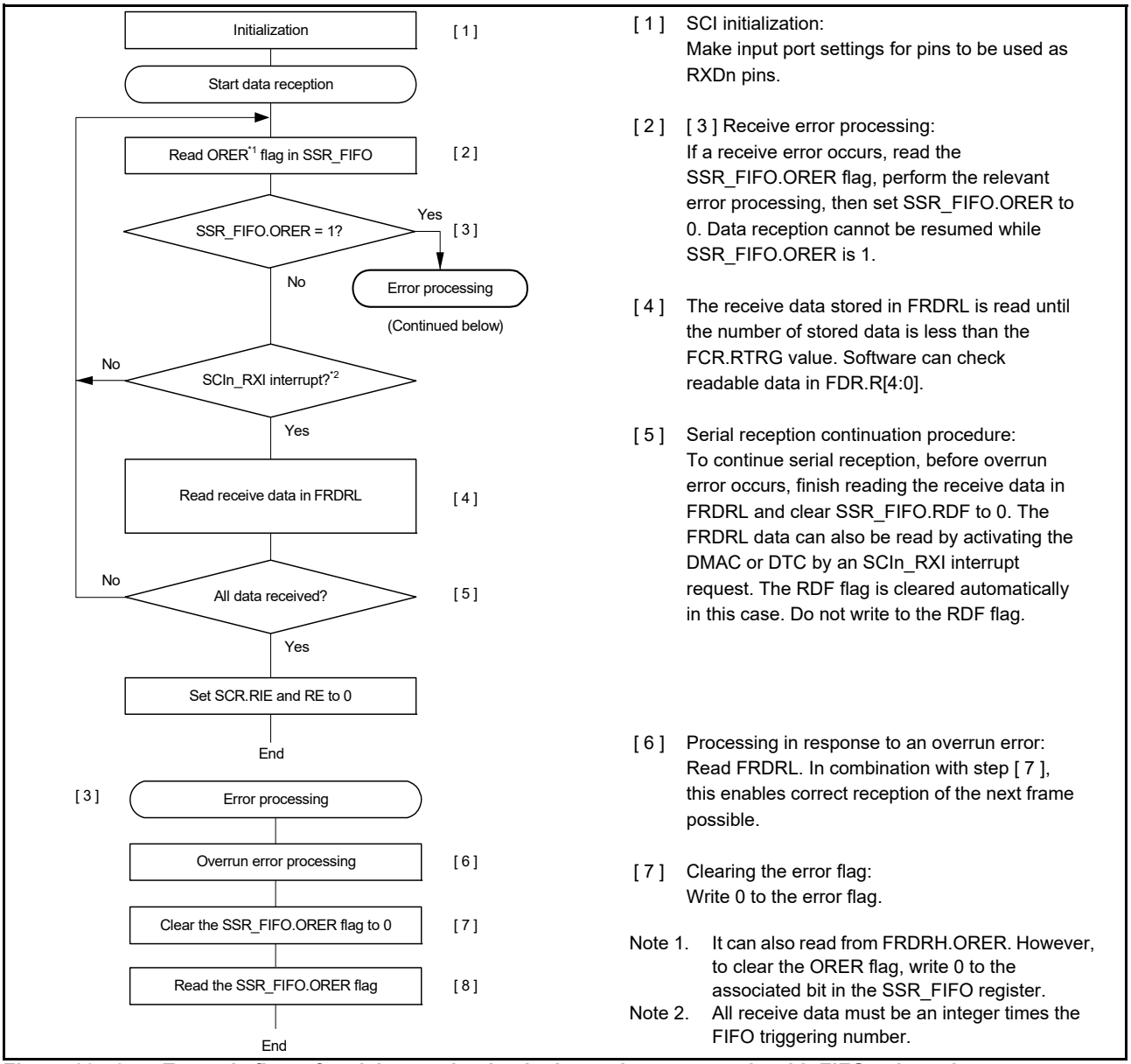


Figure 29.42 Example flow of serial reception in clock synchronous mode with FIFO selected

29.5.6 Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode

(1) Non-FIFO selected

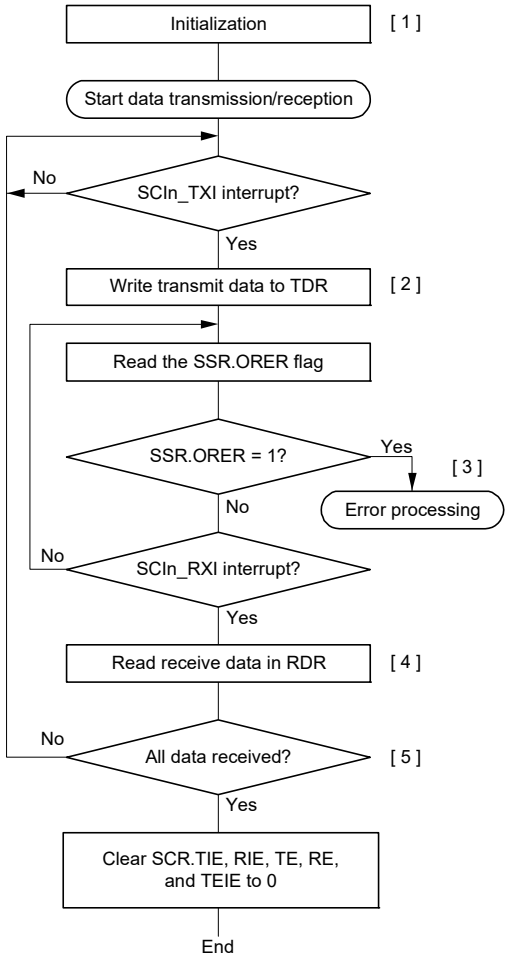
Figure 29.43 shows a sample flowchart for simultaneous serial transmit and receive operations in clock synchronous mode. After initializing the SCI, the following procedure should be used for simultaneous serial data transmit and receive operations.

To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the data transmission by verifying that the TEND flag in SSR is set to 1.
2. Initialize the SCR register and then set the TIE, RIE, TE, and RE bits in the SCR 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 then check that the receive error flags ORER in SSR are 0.
3. Set the TIE, RIE, TE, and RE bits in SCR 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 SSR.ORER flag, perform the relevant error processing, then set SSR.ORER to 0. Data reception cannot be resumed while SSR.ORER 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 DMAC or DTC by a transmit data empty interrupt (SCIn_TXI) request. Similarly, the RDR data can also be read by activating the DMAC or 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 the TIE, RIE, TE, and RE bits to 1, simultaneously.

Figure 29.43 Example flow of simultaneous serial transmission and reception in clock synchronous mode with non-FIFO selected

(2) FIFO selected

Figure 29.44 shows an example flow of simultaneous serial transmit and receive operations in clock synchronous mode with FIFO selected.

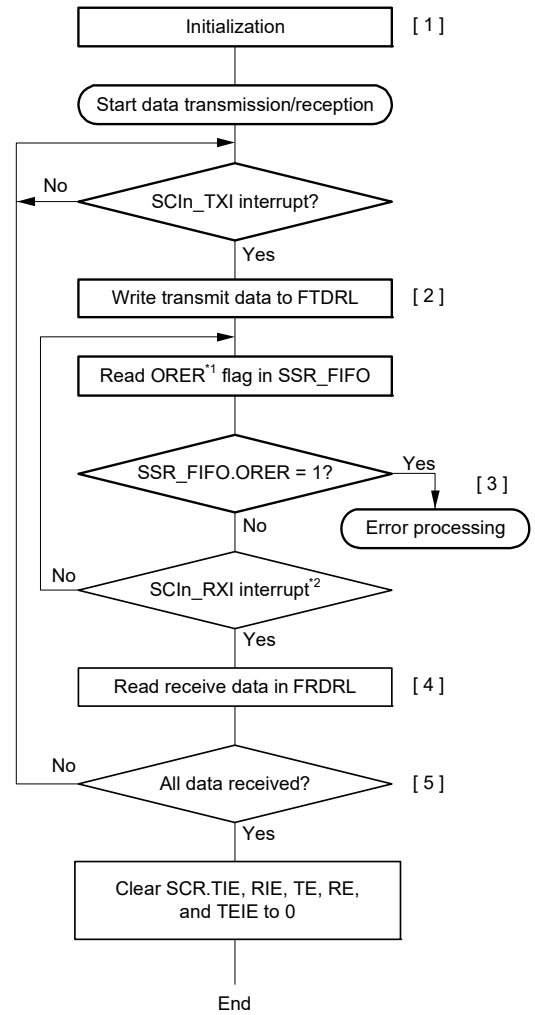
After initializing the SCI, use the following procedure for simultaneous serial data transmit and receive operations.

To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the transmission by verifying that the TEND flag in SSR_FIFO is set to 1.
2. Initialize the SCR register, then set the TIE, RIE, TE, and RE bits in the SCR 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 in SCR to 0 and then check that the receive error flag ORER in SSR_FIFO is 0.
3. Set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously using a single instruction.



Note 1. It can also read from FRDRH.ORER. To clear the ORER flag, write 0 to the associated bit in the SSR_FIFO register.

Note 2. Data must be all receive data of amount an integer times the FIFO triggering number.

[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 can be written is 16 - (the number of stored transmit data in the FIFO).

[3] Receive error processing:
If a receive error occurs, read the SSR_FIFO.ORER flag, perform the relevant error processing, then set SSR_FIFO.ORER to 0. Data reception cannot be resumed while SSR_FIFO.ORER 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 DMAC or DTC by a transmit FIFO data empty interrupt (SCIn_TXI) request. Similarly, the FRDRL data can also be read by activating the DMAC or 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.

Figure 29.44 Example flow of simultaneous serial transmission and reception in clock synchronous mode with FIFO selected

29.6 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.

29.6.1 Example Connection

Figure 29.45 shows an example connection between a smart card (IC card) and the MCU.

As shown in Figure 29.45, 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 TE and RE bits in SCR_SMCI 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.

The output port of the MCU can be used to output a reset signal.

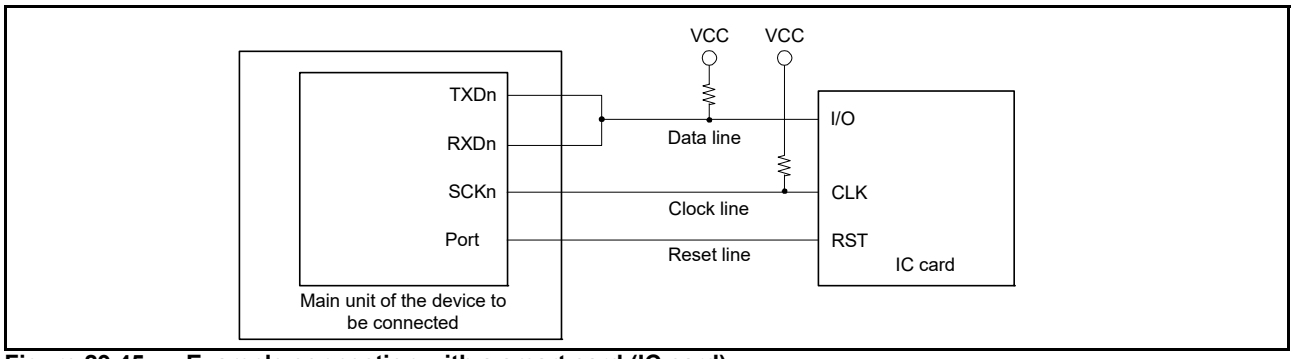


Figure 29.45 Example connection with a smart card (IC card)

29.6.2 Data Format (Except in Block Transfer Mode)

Figure 29.46 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.

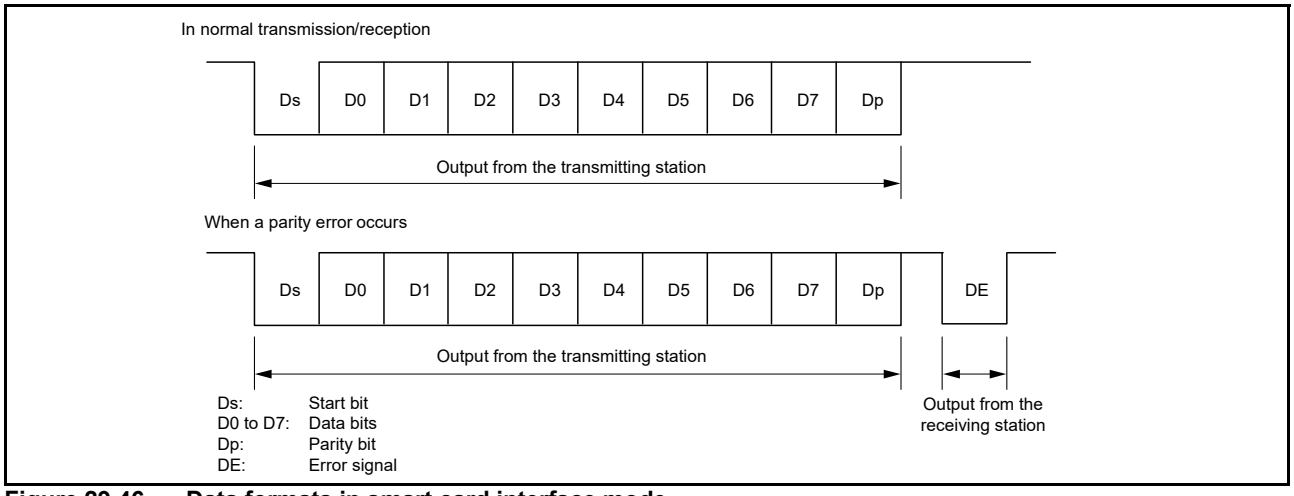


Figure 29.46 Data formats in smart card interface mode

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 as the start character, as Figure 29.47 shows. Therefore, data in the start character in the figure is 3Bh.

When using the direct convention type, write 0 to both the SDIR and SINV bits in SCMR. Write 0 to the SMR_SMCI.PM bit to use even parity, which is described by the smart card standard.

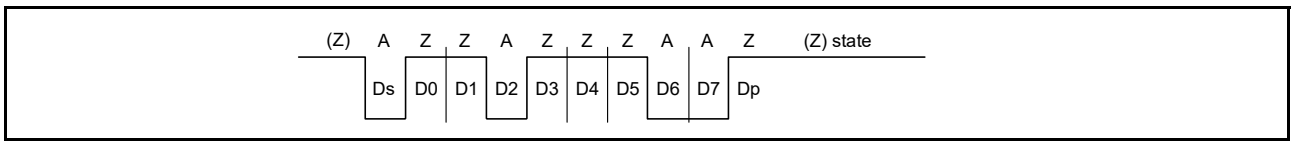


Figure 29.47 Direct convention with SDIR in SCMR = 0, SINV in SCMR = 0, and PM in SMR_SMCI = 0

(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 as the start character, as Figure 29.48 shows. Therefore, data in the start character in the figure is 3Fh.

When using the inverse convention type, write 1 to both the SDIR and SINV bits in SCMR. The parity bit is logic level 0 to produce even parity, which is described by the smart card standard, and corresponds to state Z. Because the SINV bit only inverts data bits D7 to D0, write 1 to the SMR_SMCI.PM bit to invert the parity bit for both transmission and reception.

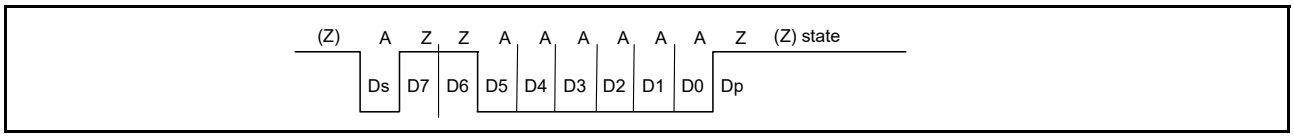


Figure 29.48 Inverse convention with SDIR in SCMR = 1, SINV in SCMR = 1, and PM in SMR_SMCI = 1

29.6.3 Block Transfer Mode

Block transfer mode differs from non-block transfer mode of smart card interface mode as follows:

- If a parity error is detected during reception, no error signal is output. Because SSR_SMCI.PER is set by error detection, clear SSR_SMCI.PER 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 SSR_SMCI.TEND flag is set to 11.5 ETUs after transmission starts
- In block transfer mode, the SSR_SMCI.ERS flag indicates the error signal status as in non-block transfer mode of the smart card interface mode, but the flag is read as 0 because no error signal is transferred.

29.6.4 Receive Data Sampling Timing and Reception Margin

Only the 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 according to the settings of SCMR.BCP2 and SMR_SMCI.BCP[1:0].

For data reception, the falling edge of the start bit is sampled with the base clock to perform 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 Figure 29.49 shows. The reception margin is determined by the following formula:

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{\left| D - 0.5 \right|}{N}(1 + F) \right| \times 100\ [\%]$$

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 by the following formula:

M = {0.5 - 1/(2 × 372)} × 100 [%] = 49.866%

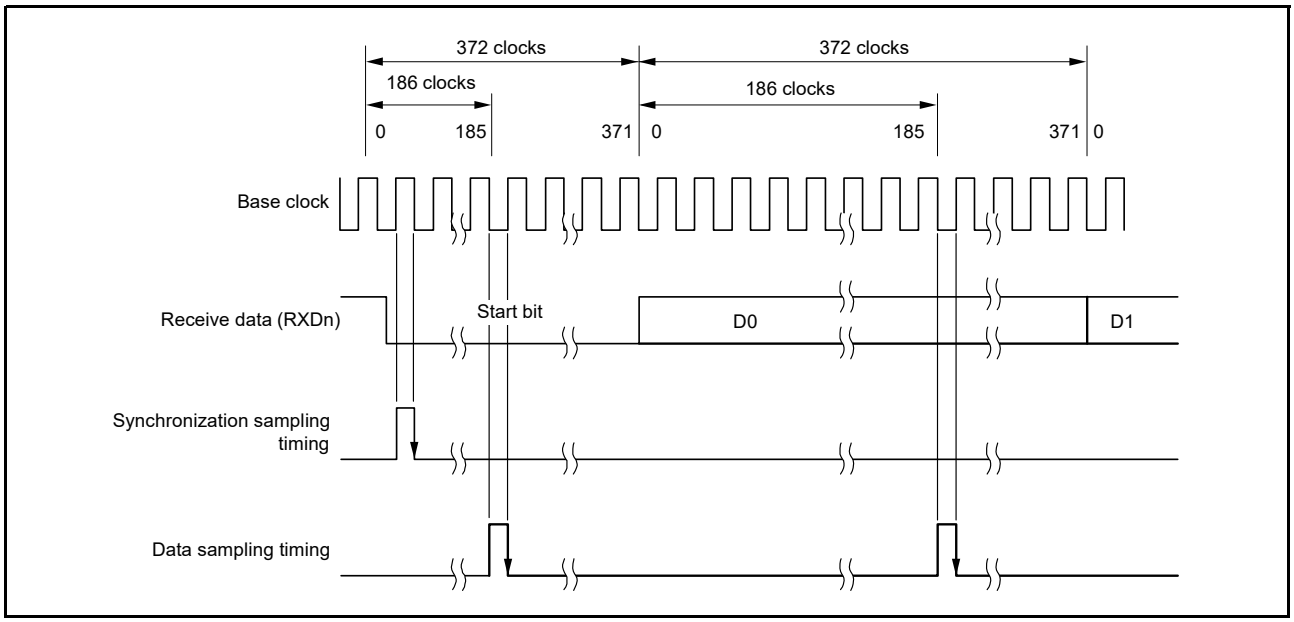


Figure 29.49 Receive data sampling timing in smart card interface mode when clock frequency is 372 times the bit rate

29.6.5 Initialization of the SCI

Before transmitting and receiving data, write the initial value 00h to SCR_SMCI and initialize the SCI following the flowchart example shown in Figure 29.50.

Be sure to set the initial value in the TIE, RIE, TE, RE, TEIE bits in SCR_SMCI 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 from 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 SSR_SMCI.TEND flag.

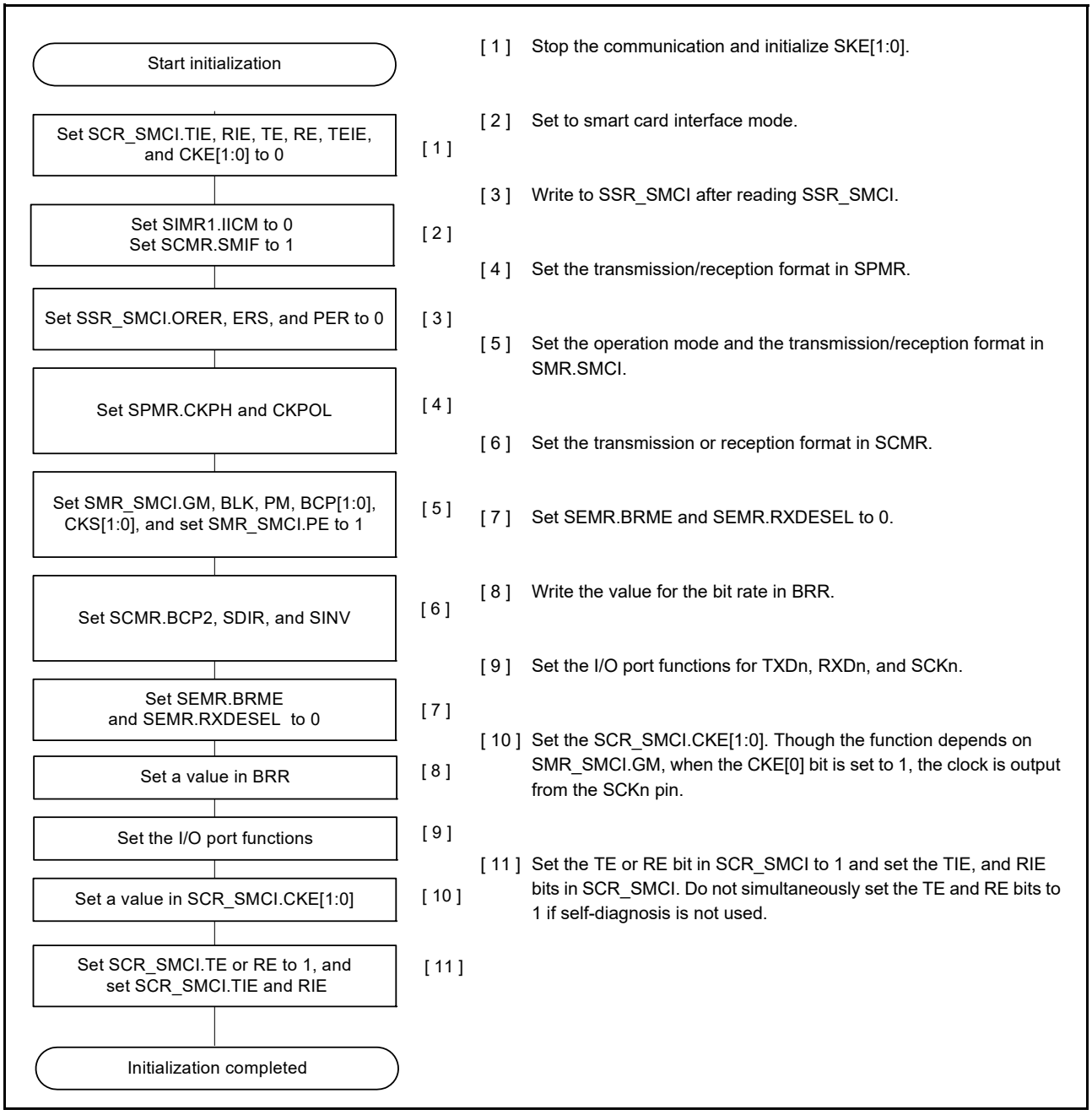


Figure 29.50 Example flow of SCI initialization in smart card interface mode

Figure 29.51 shows a timing chart when data transmission is performed by making a transition to the smart card interface mode. The figure shows the case when the GM bit in SMR_SMCI is set to 0. As shown in the figure, when the port is connected as SCKn pin and TXDn pin, they are in high-impedance state because SCR_SMCI.CKE [0] bit is 0.

Start clock output to the SCK pin by setting the clock output, SCR_SMCI.CKE [0] bit to 1, and start data transmission by writing transmit data after setting the SCR_SMCI.TE bit to 1. As soon as the SCR_SMCI.TE bit changes from 0 to 1, there is a preamble period for one frame, and data transmission follows. In the smart card interface mode, the TXDn pin is in high-impedance state in the preamble period. Due to the state of the SCKn pin and TXDn pins, pull-up or pull-down is required outside the MCU. In the smart card interface mode, even if communication with SCR_SMCI.TE bit = 0 and SCR_SMCI.RE bit = 0 is not performed, the clock is continuously output if the clock output setting is used.

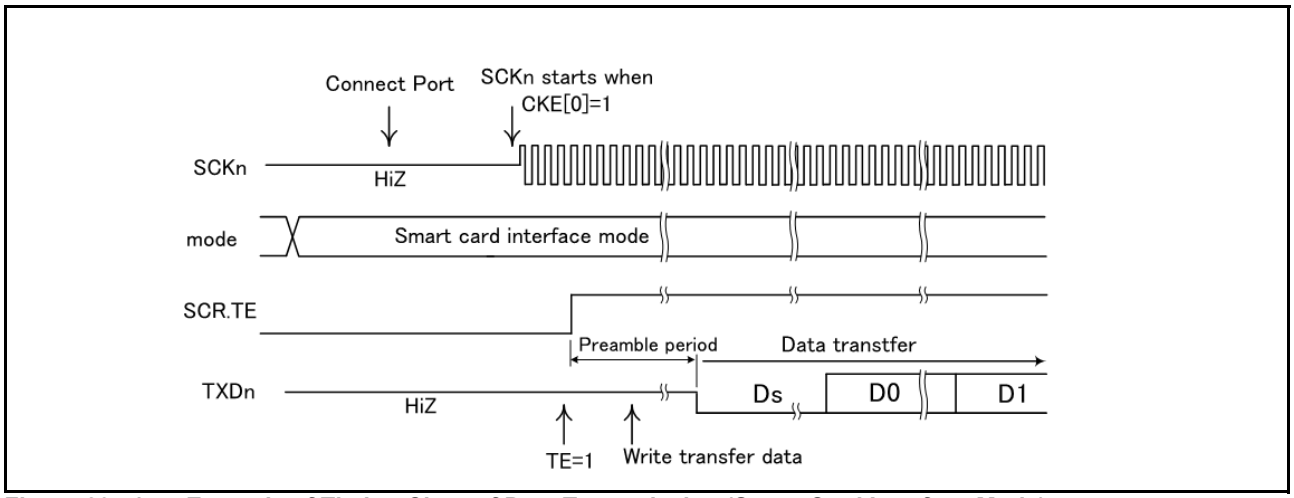


Figure 29.51 Example of Timing Chart of Data Transmission (Smart Card Interface Mode)

29.6.6 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 retransmitted in smart card mode. Figure 29.52 shows the data retransfer operation during transmission.

[1] When an error signal from the receiver end is sampled after 1-frame data is transmitted, the ERS flag in SSR_SMCI is set to 1. If SCR_SMCI.RIE is 1, an SCIn_ERI interrupt request is generated. Clear the SSR_SMCI.ERS flag to 0 before the next parity bit is sampled.

[2] For a frame in which an error signal is received, the TEND flag in SSR_SMCI is not set. Data is retransferred 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 retransfer, is complete, and the TEND flag is set. If SCR_SMCI.TIE is 1, an SCIn_TXI interrupt request is generated. Write transmit data to the TDR to start transmission of the next data.

Figure 29.54 shows an example flow of serial transmission. All the processing steps are automatically performed using an SCIn_TXI interrupt request to activate the DMAC or DTC.

When the SSR_SMCI.TEND flag is set to 1 in transmission and when SCR_SMCI.TIE is 1, an SCIn_TXI interrupt request is generated.

The DMAC or DTC is activated by an SCIn_TXI interrupt request if the SCIn_TXI interrupt request is specified as a source of DMAC or DTC activation beforehand, allowing the transfer of transmit data. The TEND flag is automatically set to 0 when the DMAC or DTC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept to 0 and the DMAC or DTC is not activated. Therefore, the SCI and DMAC or 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 DMAC or DTC, be sure to enable the DMAC or DTC before setting the SCI. For DMAC or DTC settings, see section 17, DMA Controller (DMAC) and section 18, Data Transfer Controller (DTC).

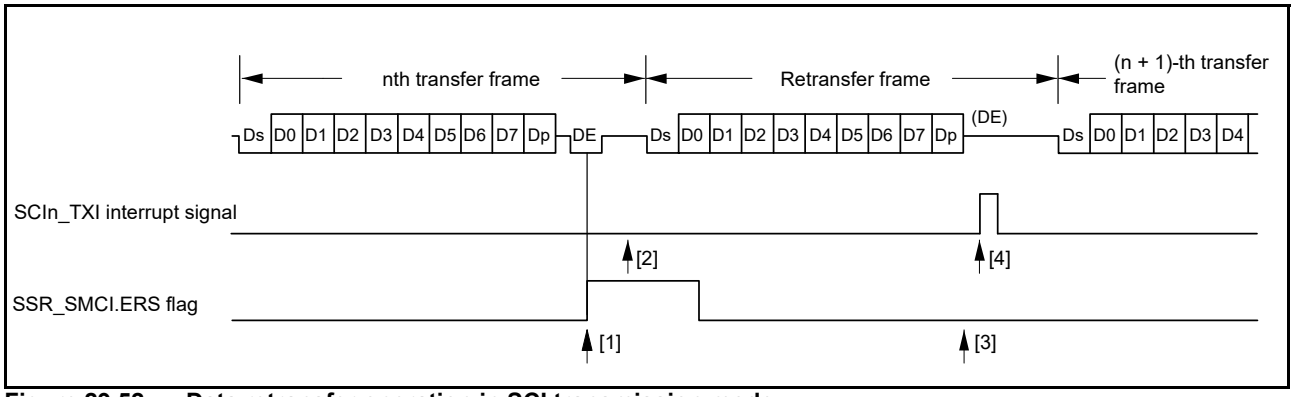


Figure 29.52 Data retransfer operation in SCI transmission mode

Note: The SSR_SMCI.TEND flag is set at different timings depending on the GM bit setting in SMR_SMCI.

Figure 29.53 shows the TEND flag generation timing.

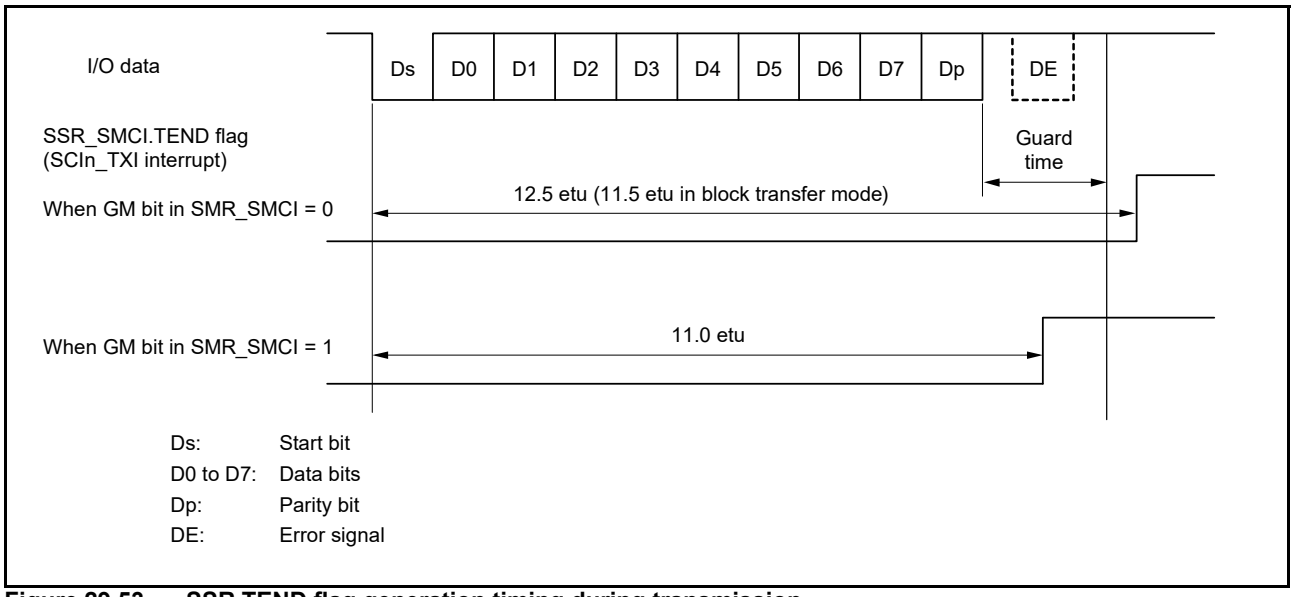


Figure 29.53 SSR.TEND flag generation timing during transmission

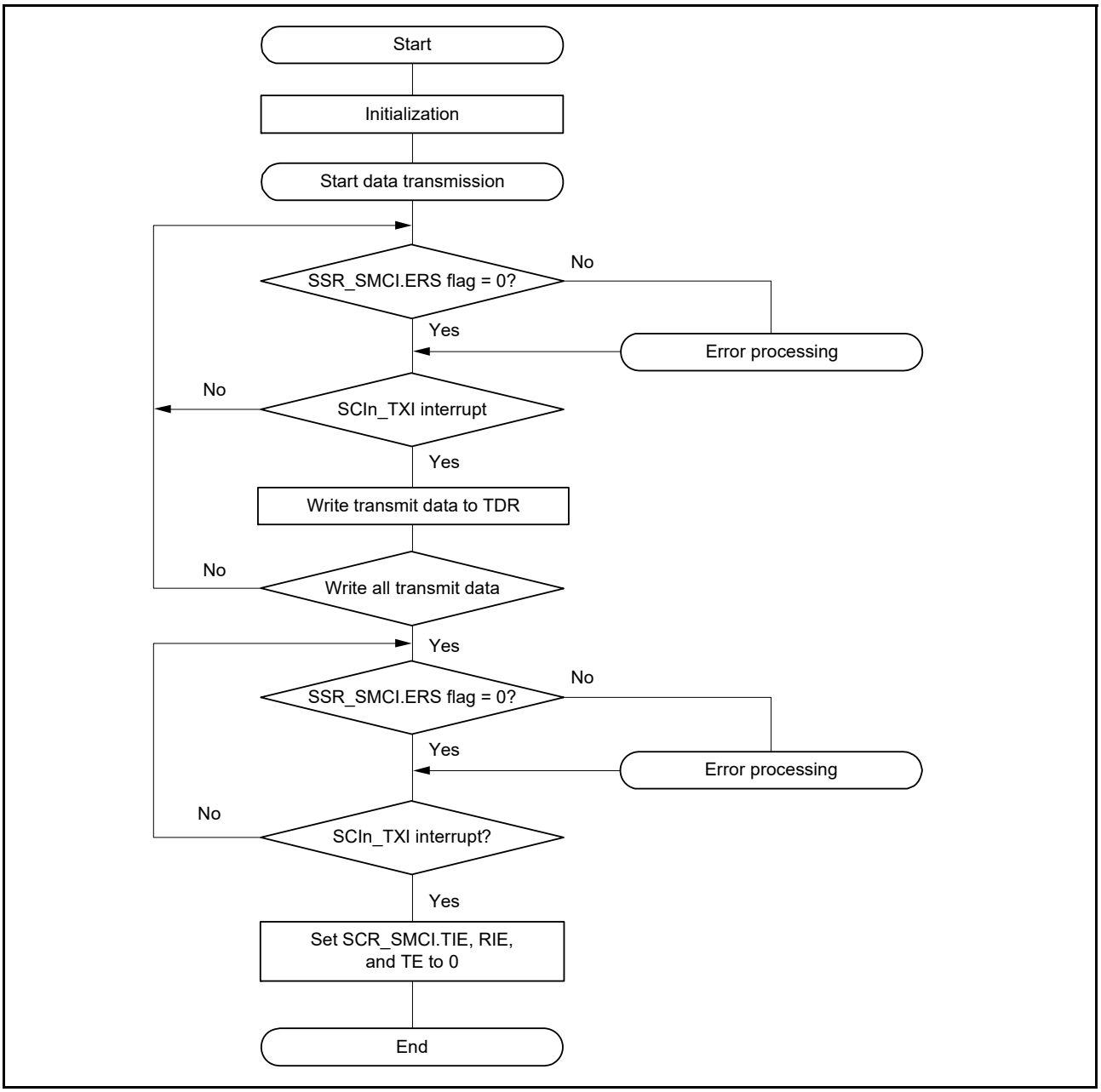


Figure 29.54 Example flow of smart card interface transmission

29.6.7 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 29.55 shows the data retransfer operation in reception mode.

[1] If a parity error is detected in the receive data, the PER flag in SSR_SMCI is set to 1. When the RIE bit in SCR_SMCI 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 SSR_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 29.56 shows an example flow for serial data reception. All the processing steps are automatically performed using an SCIn_RXI interrupt request to activate the DMAC or DTC.

In reception, setting the RIE bit to 1 allows an SCIn_RXI interrupt request to be generated. The DMAC or DTC is activated by an SCIn_RXI interrupt request if the SCIn_RXI interrupt request is specified as a source of DMAC or DTC activation beforehand, 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 DMAC or DTC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DMAC or 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, therefore 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 29.3.9, Serial Data Reception in Asynchronous Mode.

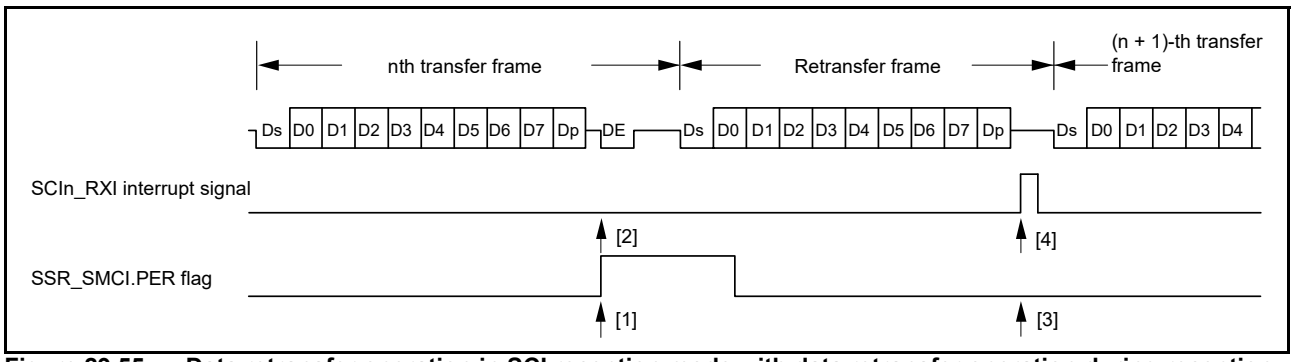


Figure 29.55 Data retransfer operation in SCI reception mode with data retransfer operation during reception

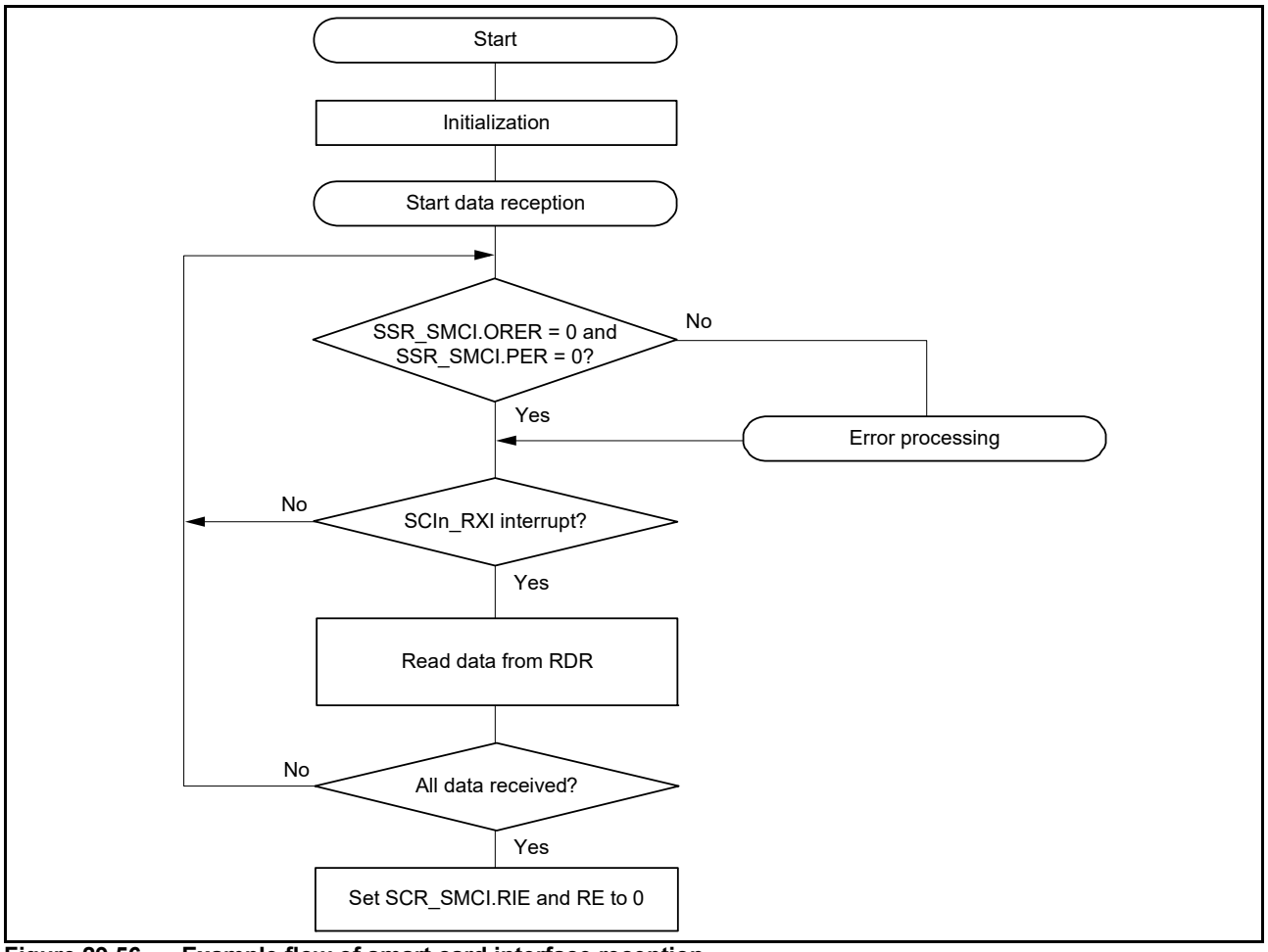


Figure 29.56 Example flow of smart card interface reception

29.6.8 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 29.2.12, Serial Control Register for Smart Card Interface Mode (SCR_SMCI) (SCMR.SMIF = 1). When setting the clock output, the base clock described in section 29.6.4, Receive Data Sampling Timing and Reception Margin is output, so the width of the clock pulse can be kept to the width specified by setting the bit rate. The bit rate is set by the CKS bit in SMR_SMCI, BCP bit in SMR_SMCI, BCP2 bit in SCMR, BRR register.

Figure 29.57 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 is immediately reflected on the SCK pin, so there is a possibility that pulses with an unintended width may be output from the SCK pin.

When the GM bit in SMR_SMCI is 1, the output pulse control by the CKE [0] bit in SCR_SMCI controls the pulse width set so as to be based on the state of the base clock.

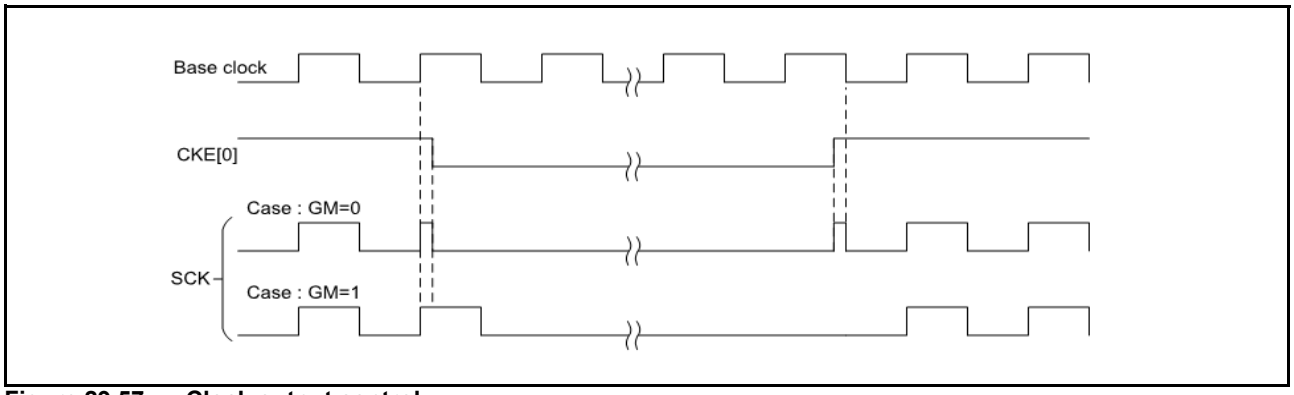


Figure 29.57 Clock output control

29.7 Operation in Simple IIC Mode

Simple I2C bus 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 I2C bus format and timing of the I2C bus are shown in Figure 29.58 and Figure 29.59.

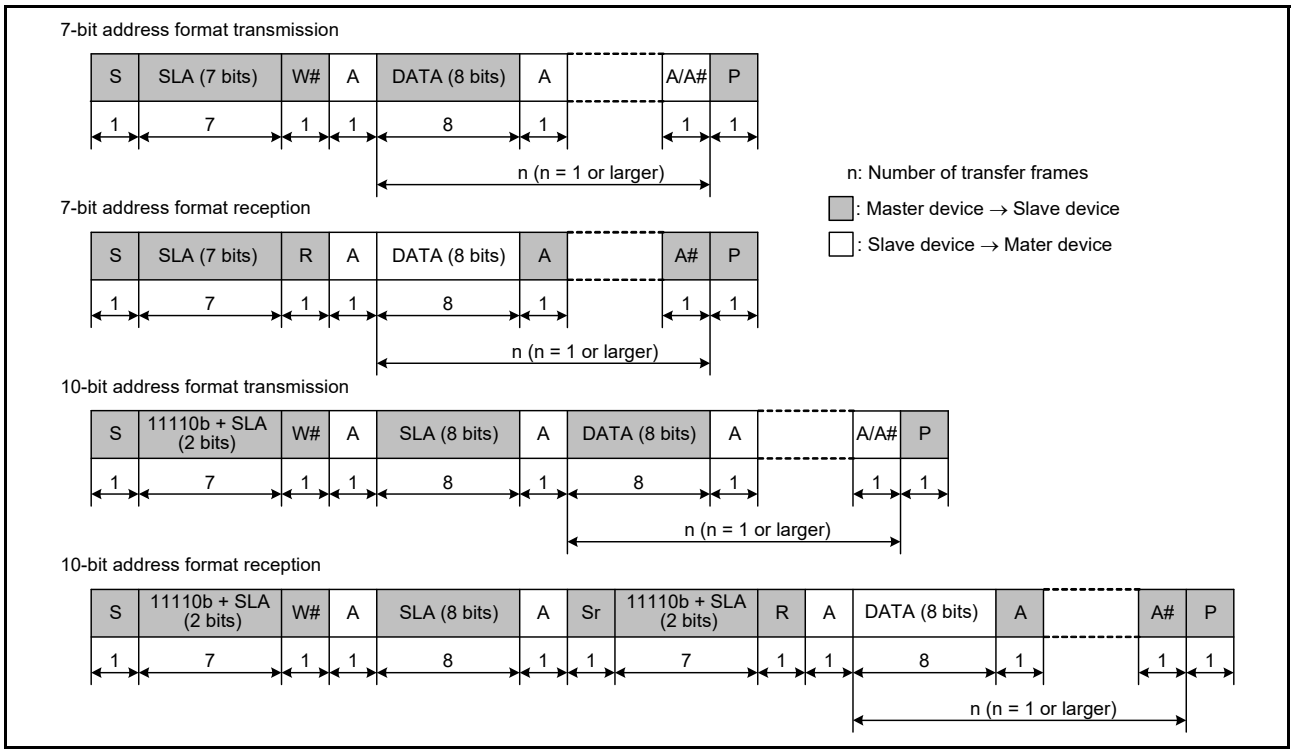


Figure 29.58 I2C bus format

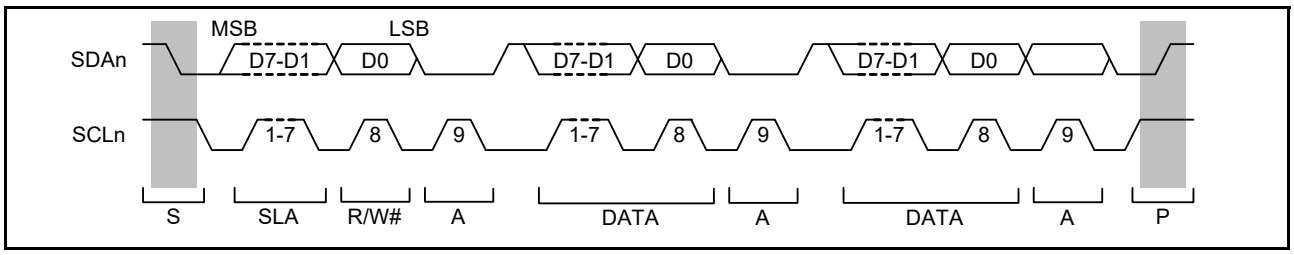


Figure 29.59 I2C bus timing when SLA is 7 bits

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 to transfer from the slave device to the master device and 0 indicates to 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.

29.7.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the IICSTAREQ bit in SIMR3 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 the high level on the SCLn line is detected, 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 IICRSTAREQ bit in SIMR3 is set to 0, and a restart-condition generated interrupt is output.

Writing 1 to the IICSTPREQ bit in SIMR3 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 setting of BRR
- The SCLn line is released (transition from the low to the high level)
- When the high level on the SCLn line is detected, 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 is released (transition from the low to the high level), the IICSTPREQ bit in SIMR3 is set to 0, and a stop-condition generated interrupt is output.

Figure 29.60 shows the timing of operations in the generation of start, restart, and stop conditions.

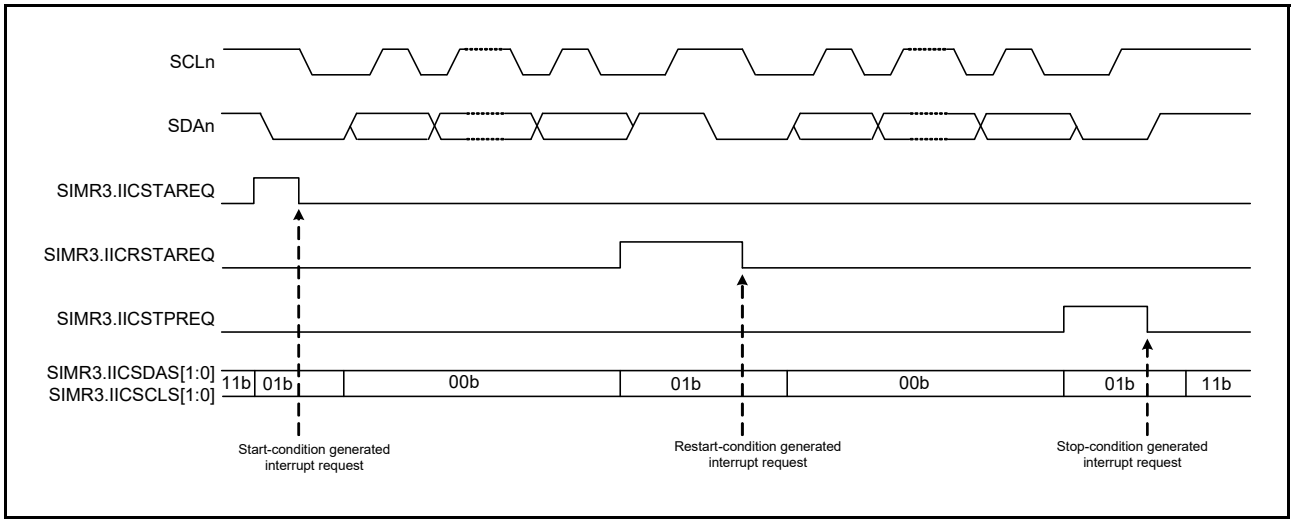


Figure 29.60 Timing of operations to generate start, restart, and stop conditions

29.7.2 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 IICCSC bit in SIMR2 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 IICCSC bit in SIMR2 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 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 the 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 IICCSC bit in SIMR2 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 IICCSC bit in SIMR2 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 29.61 shows an example of operations to synchronize the clocks.

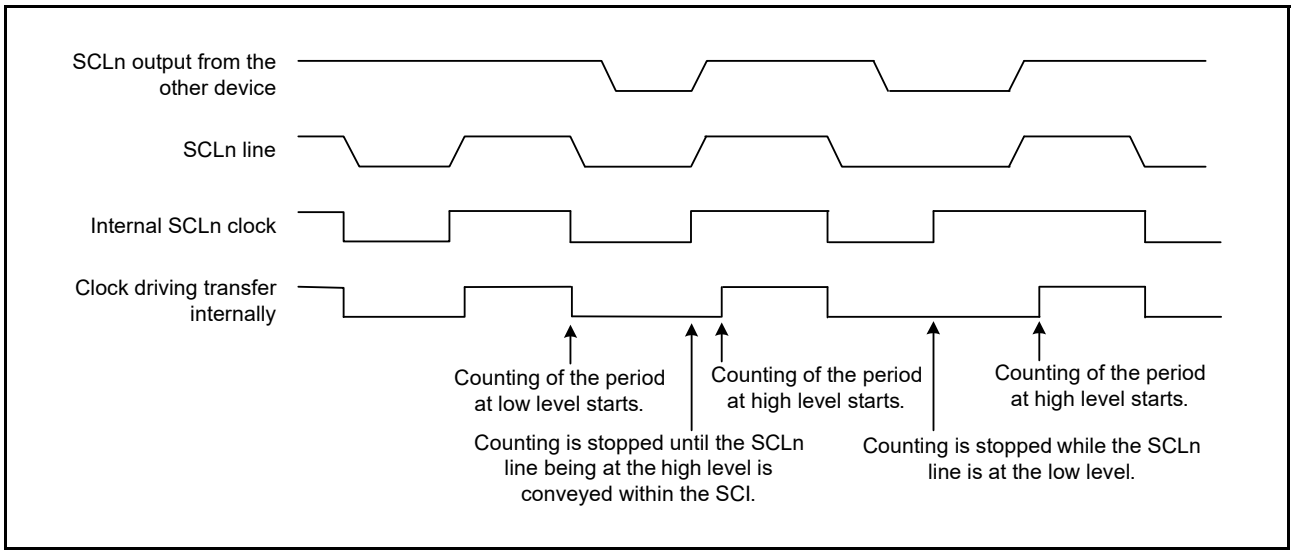


Figure 29.61 Example operations for clock synchronization

29.7.3 SDA Output Delay

The IICDL[4:0] bits in SIMR1 can be used to set a delay for output on the SDAn pin relative to the 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]). A delay for output on the SDAn pin applies to the start condition/restart condition/stop condition signal, 8-bit transmit data, and an 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 standard mode and fast mode).

Figure 29.62 shows the timing of delays in SDAn output.

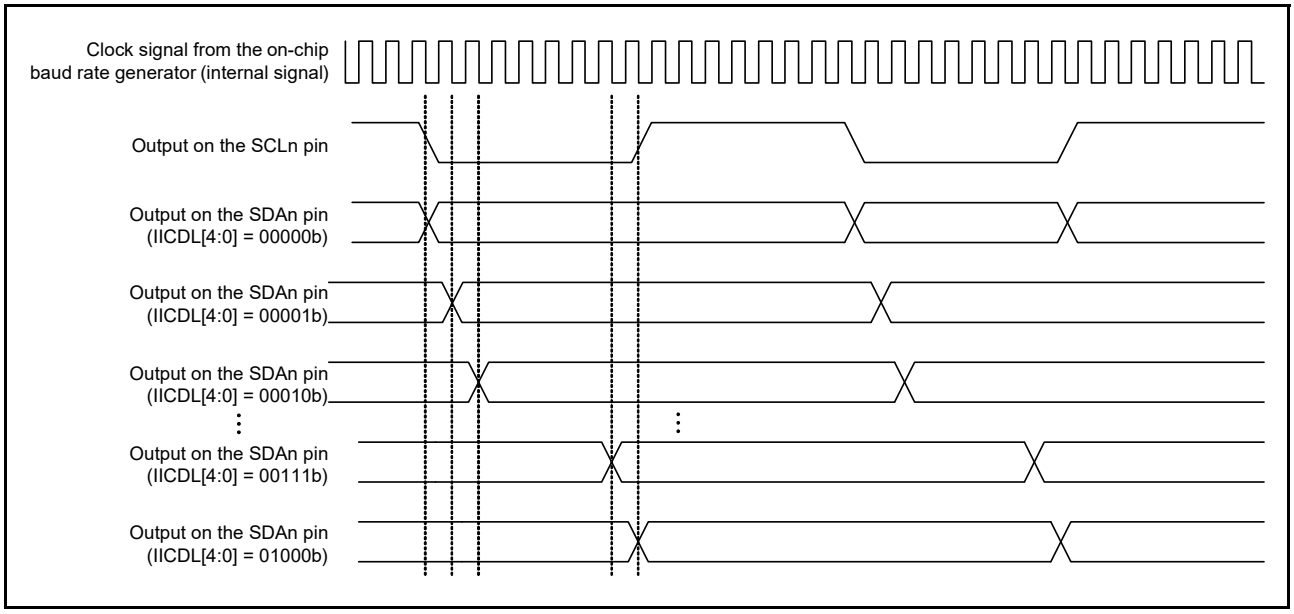


Figure 29.62 Timing of delays in SDAn output

29.7.4 SCI Initialization in Simple IIC Mode

Before transferring data, write the initial value 00h to SCR and initialize the interface following the example shown in Figure 29.63.

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.

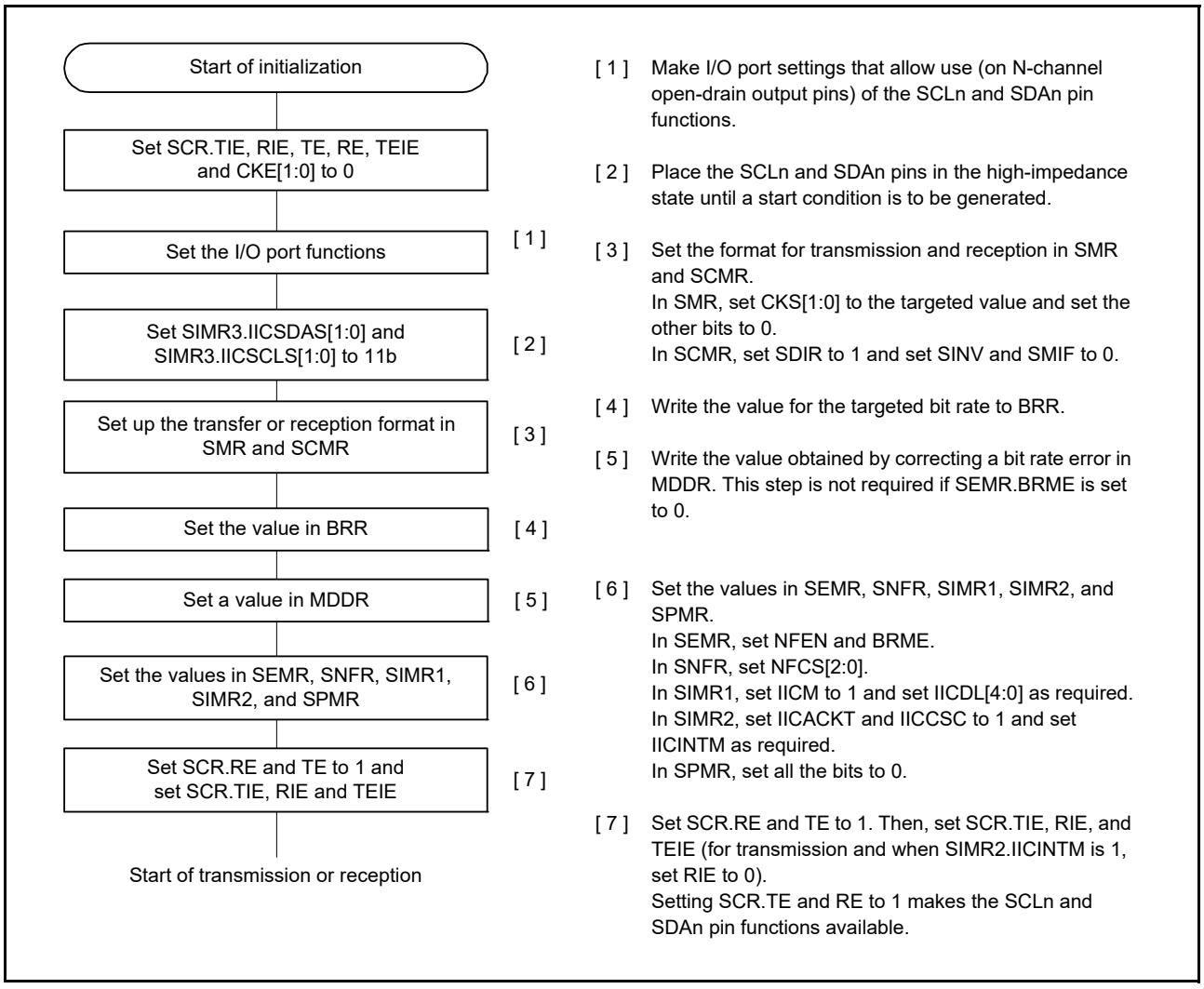


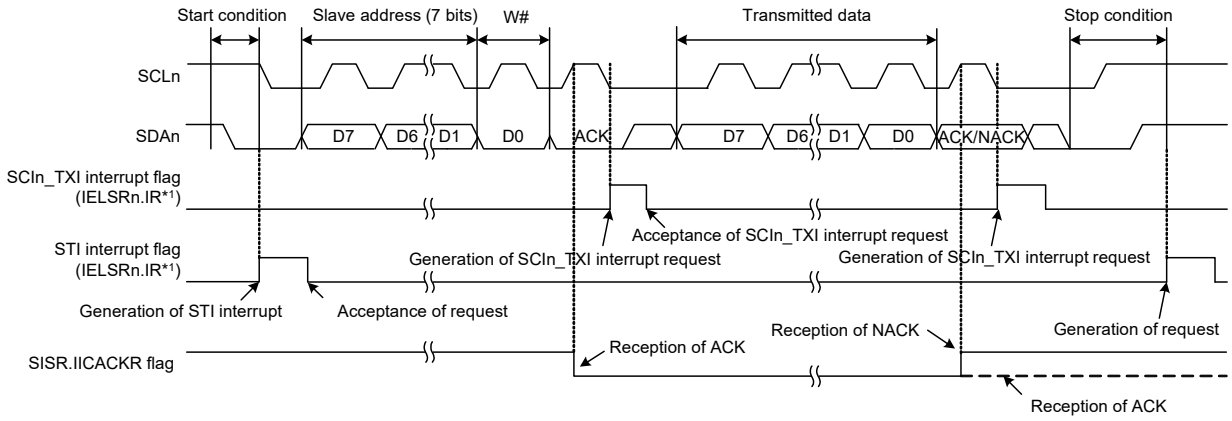
Figure 29.63 Example flow of SCI initialization in simple IIC mode

29.7.5 Operation in Master Transmission (Simple IIC Mode)

Figure 29.64 and Figure 29.65 show examples of master transmission and Figure 29.66 shows an example flow of data transmission. The value of SIMR2.IICINTM is assumed to be 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 29.28 for more information on the STI interrupt.

When 10-bit slave addresses are in use, steps [3] and [4] in Figure 29.66 are repeated twice.

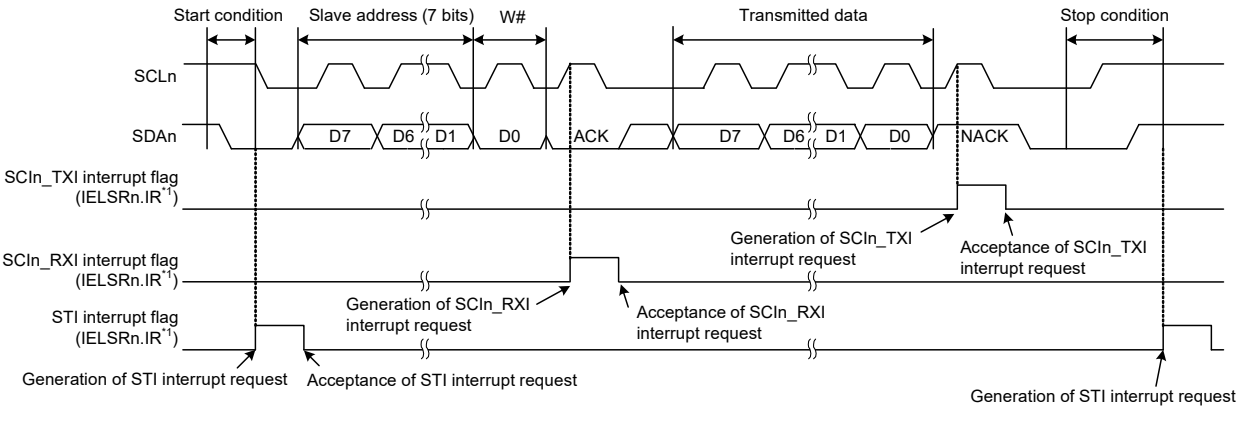
In simple IIC mode, the transmit data empty interrupt (SCIn_TXI) is generated when communication of one frame is complete, unlike the SCIn_TXI interrupt request generation timing during clock synchronous transmission.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 29.64 Example 1 of operations for master transmission in simple IIC bus 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 DMAC or DTC is activated by the ACK interrupt as the trigger and required number of data bytes are transmitted. When a NACK is received, error processing such as transmission stop and retransmission, is performed using the NACK interrupt as the trigger.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 29.65 Example 2 of operations for master transmission in simple IIC bus mode with 7-bit slave addresses, ACK interrupts, and NACK interrupts

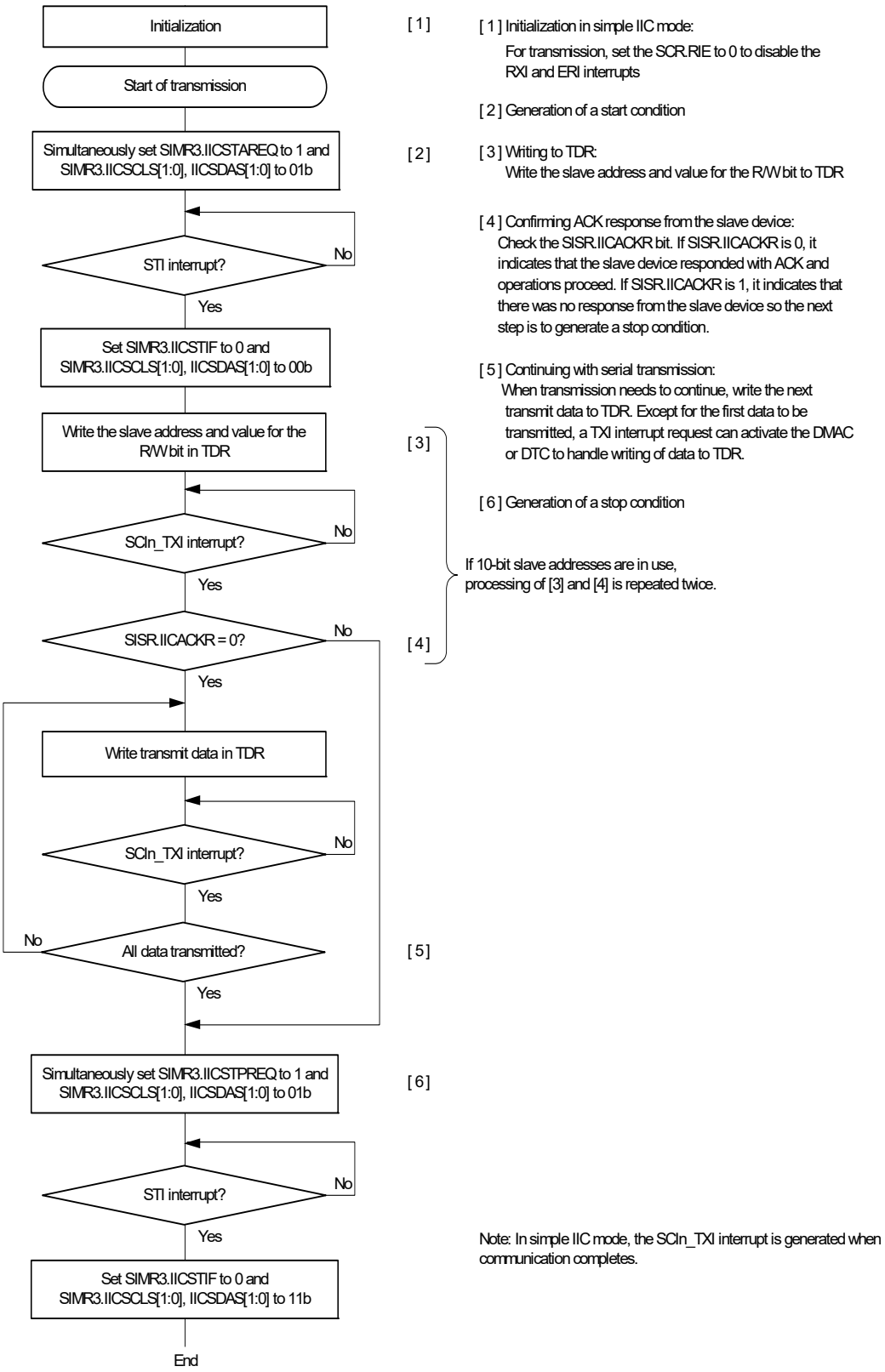


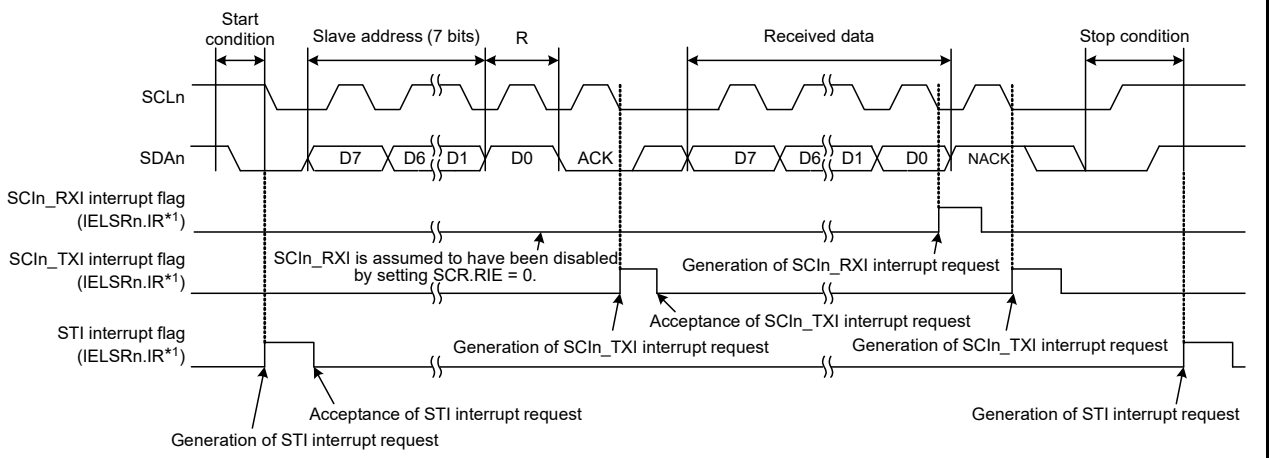
Figure 29.66 Example flow of master transmission in simple IIC mode with transmission interrupts and reception interrupts

29.7.6 Master Reception in Simple IIC Mode

Figure 29.67 shows an example operation in simple IIC mode master reception and Figure 29.68 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 completed, unlike the SCIn_TXI interrupt request generation timing during clock synchronous transmission.



Note 1. See section 14, Interrupt Controller Unit (ICU) for information on the corresponding interrupt event number.

Figure 29.67 Example operations for master reception in simple IIC bus mode with 7-bit slave addresses, transmission interrupts, and reception interrupts

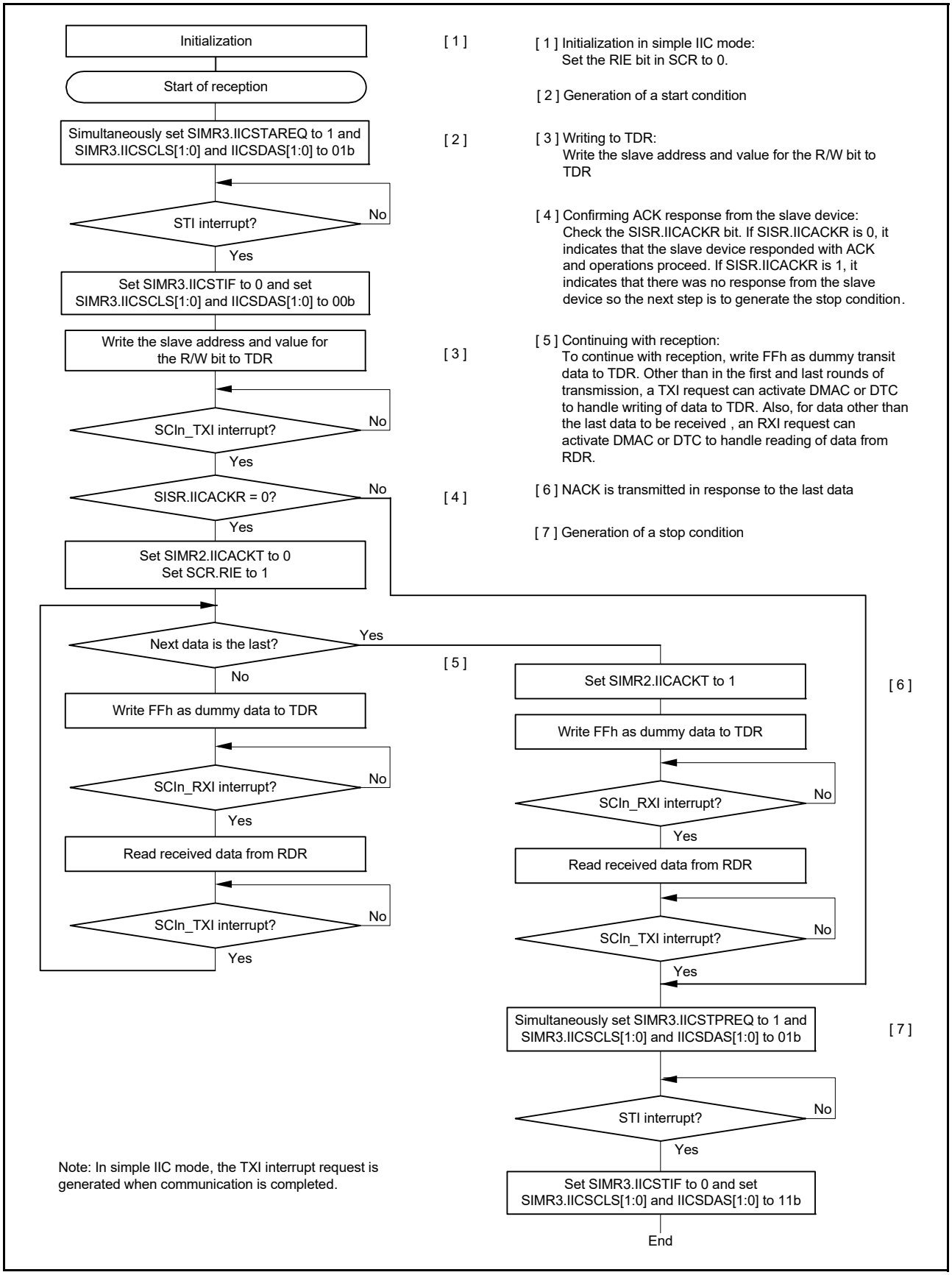


Figure 29.68 Example flow of master reception in simple IIC mode with transmission interrupts and reception interrupts

29.8 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) plus setting SPMR.SSE to 1 places 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 SSE bit in SPMR to 0 in such cases.

Figure 29.69 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 to this. The data can be inverted by setting SCMR.SINV to 1.

Because the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a common 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.

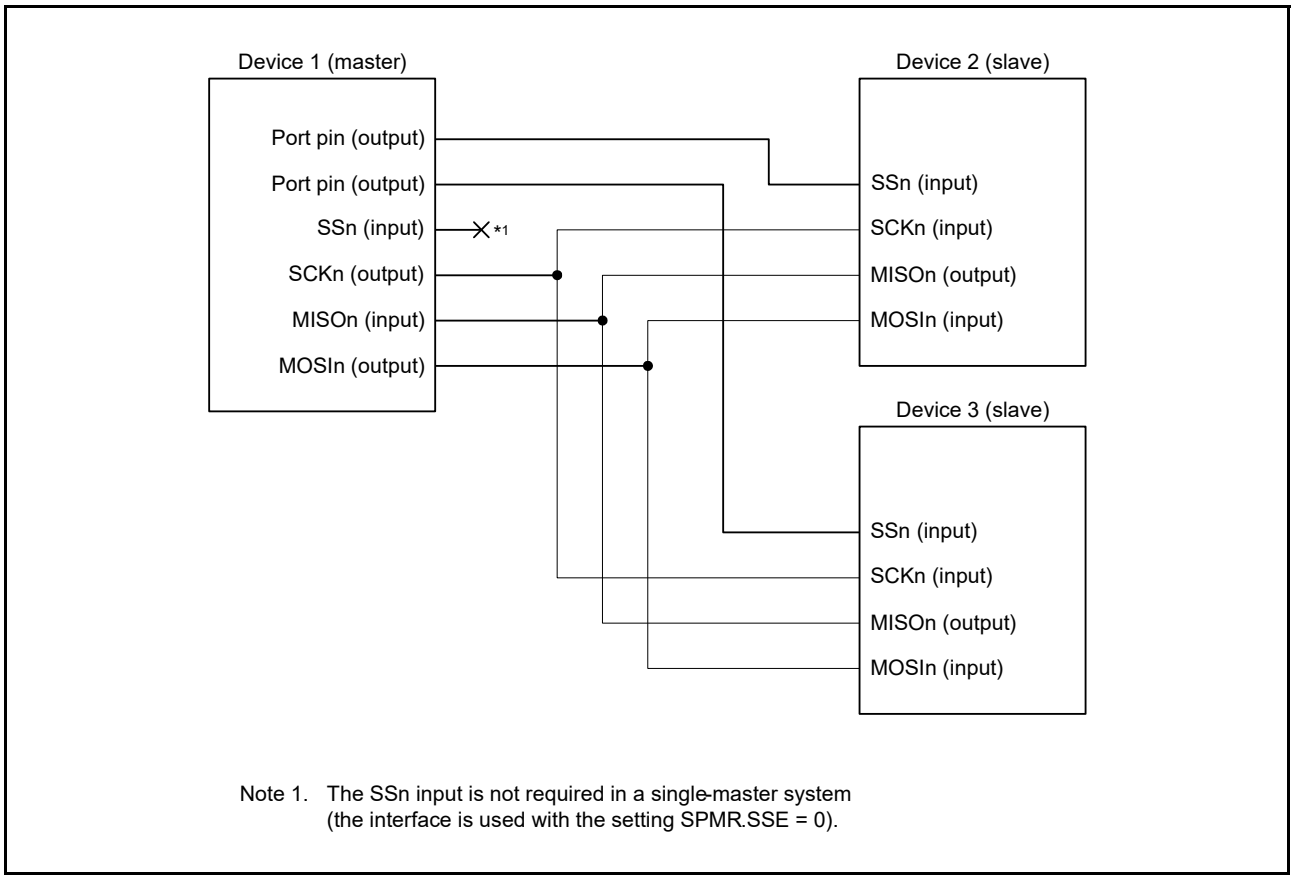


Figure 29.69 Example connections using simple SPI mode in single master mode with SPMR.SSE bit = 0

29.8.1 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 29.24 lists the relationship between the pin states, mode, and level on the SSn pin.

Table 29.24 States of pins by mode and input level on SSn Pin

Mode	Input on SSn pin	State of TXDn pin	State of RXDn 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. Because the SSn pin function is not required, the pin is available for other purposes.
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 and RE bits = 00b) in a multimaster configuration (SPMR.SSE = 1).

29.8.2 SS Function in Master Mode

Setting the SCR.CKE[1:0] bits to 00b and the SPMR.MSS bit to 0 selects master 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 is not stopped, but the MOSIn and SCKn outputs are in the high-impedance state after the completion of the transfer. Control a general port pin to produce the SS output signal from the master.

29.8.3 SS Function in Slave Mode

Setting the SCR.CKE[1:0] bits to 10b and the SPMR.MSS bit to 1 selects slave operation. When the level on 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 level on the SSn pin is low, clock input through the SCKn pin is effective and transmission or reception can proceed.

If the input on the SSn pin changes from low to high level 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 currently being transmitted or received completes, after which it stops, and the appropriate interrupt (SCIn_TXI, SCIn_RXI, or SCIn_TEI) is generated.

29.8.4 Relationship between Clock and Transmit/Receive Data

The CKPOL and CKPH bits in SPMR 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 29.70. 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.

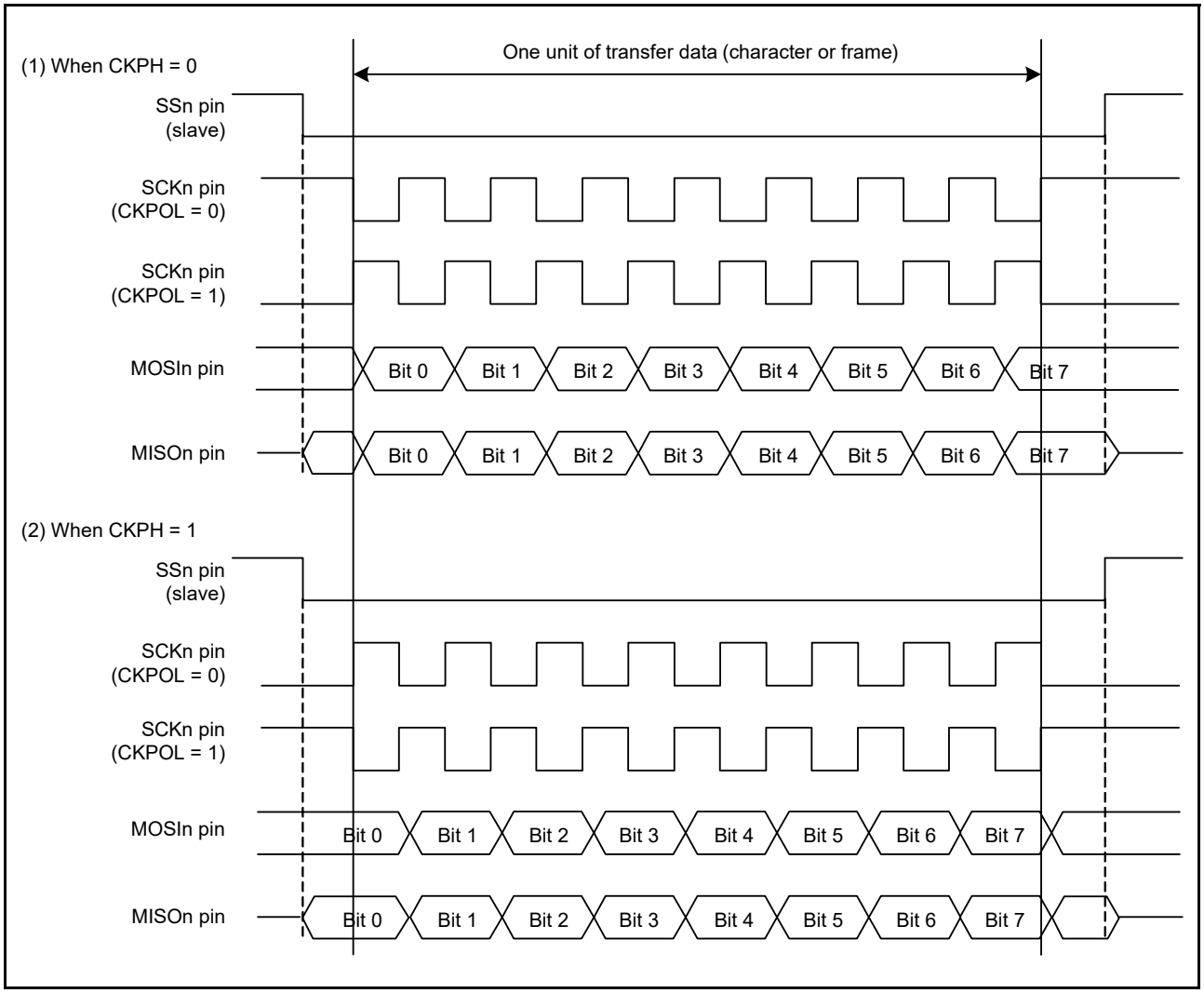


Figure 29.70 Relation between clock signal and transmit or receive data in simple SPI mode

29.8.5 SCI Initialization (Simple SPI Mode)

Initialization in simple SPI mode is the same as in clock synchronous mode. See Figure 29.32 for an example initialization flow. The CKPOL and CKPH bits in SPMR must be set to ensure that the clock signal configuration they select 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 1. Only the RE bit is set to 0. The SSR.ORER, FER, PER, and RDR flags are not initialized.
Note 2. Changing the value of the TE bit from 1 to 0 or from 0 to 1 leads to the generation of a transmit data empty interrupt (SCIn_TXI) if the value of the TIE bit in the SCR is 1 at the time.

29.8.6 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.

29.9 Bit Rate Modulation Function

Using the bit rate modulation function, the bit rate can be evenly corrected using the number specified in MDDR when PCLK is selected in the CKS[1:0] bits in SMR/SMR_SMCI.

Figure 29.71 shows an example where PCLK is selected by the CKS[1:0] bits in SMR/SMR_SMCI and BRR and MDDR are set to 0 and 160 respectively, 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. Contraction is 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).

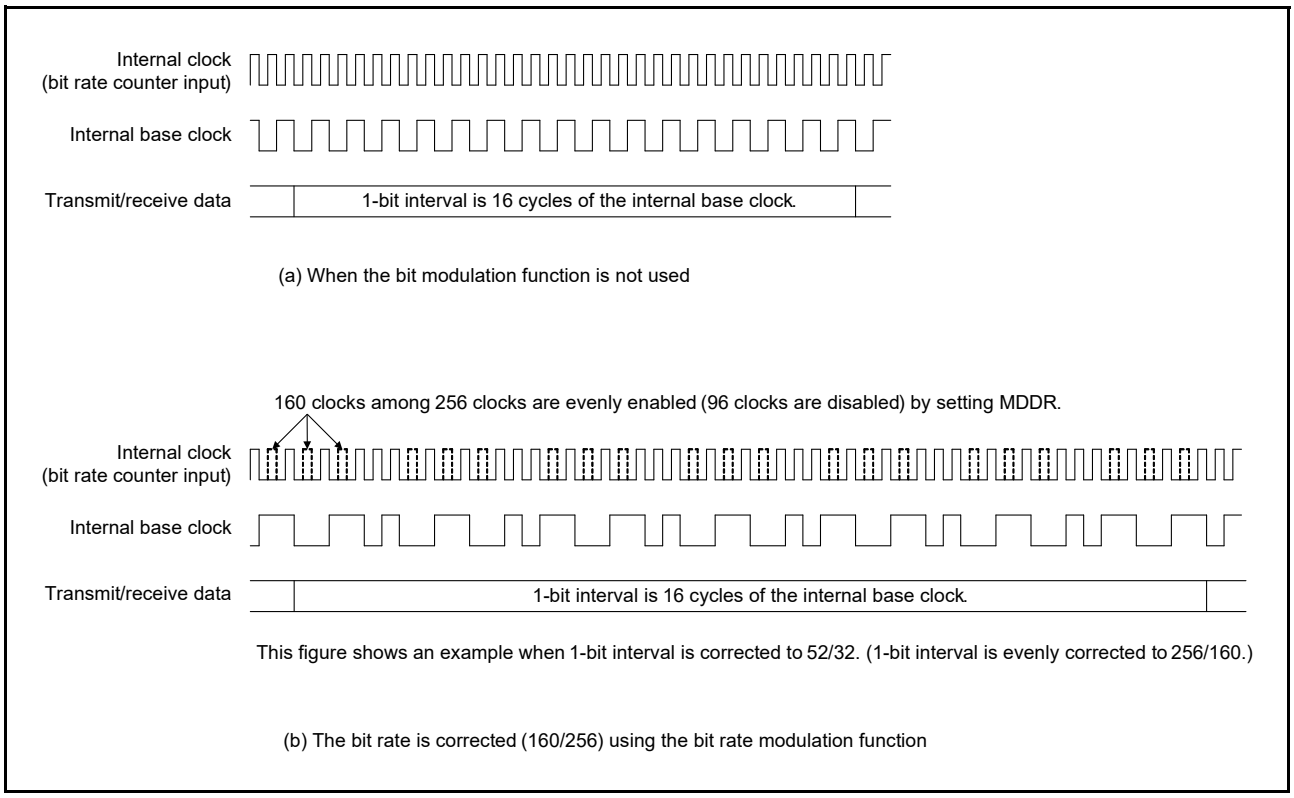


Figure 29.71 Example internal base clock when bit rate modulation function is used

29.10 Interrupt Sources

29.10.1 Buffer Operations 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 Interrupt Controller Unit is 1, the ICU does not output the interrupt request but retains it internally with a capacity for retention of one request per source.

When the interrupt status flag in the ICU is 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.

29.10.2 Buffer Operations for SCIn_TXI and SCIn_RXI Interrupts (FIFO selected)

When an interrupt status flag in the Interrupt Controller Unit is set to 1, the SCIn_TXI and SCIn_RXI interrupts do not output interrupt requests to the Interrupt Controller Unit. When an interrupt status flag of the Interrupt Controller Unit is set to 0, and if the conditions for an SCIn_TXI and SCIn_RXI interrupts are satisfied, an interrupt request is generated.

29.10.3 Interrupts in Asynchronous, Clock Synchronous, and Simple SPI Modes

(1) Non-FIFO selected

Table 29.25 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 SCR.

If the SCR.TIE bit is 1, an SCIn_TXI interrupt request is generated when transmit data is transferred from TDR or TDRHL*1 to the TSR. An SCIn_TXI interrupt request can also be generated using a single instruction to set the SCR.TE and SCR.TIE bits to 1 simultaneously. An SCIn_TXI interrupt request can activate the DMAC or DTC to handle data transfer.

An SCIn_TXI interrupt request is not generated by setting SCR.TE to 1 when SCR.TIE is 0 or by setting SCR.TIE to 1 when 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 becomes 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 are 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 SCR.RIE is 1, an SCIn_RXI interrupt request is generated when received data is stored in RDR. An SCIn_RXI interrupt request can activate the DMAC or DTC to handle data transfer.

Setting any of the ORER, FER, and PER flags in SSR to 1 when SCR.RIE is 1 leads to the generation of an SCIn_ERI interrupt request. An SCIn_RXI interrupt request is not generated at this time. Clearing all three flags (ORER, FER, and PER) leads to discarding of the SCIn_ERI interrupt request.

(2) FIFO selected

Table 29.26 lists interrupt sources in FIFO selected mode.

If SCR.TIE is 1, an SCIn_TXI interrupt request is generated when the stored number of data in the FTDRL register becomes the threshold value indicated in FCR.TTRG or less. An SCIn_TXI interrupt request can also be generated using a single instruction to set the SCR.TE and SCR.TIE bits to 1 simultaneously.

An SCIn_TXI interrupt request is not generated by setting SCR.TE to 1 when SCR.TIE is 0 or by setting SCR.TIE to 1 when SCR.TE is 1.

If SCR.TEIE is 1and 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, an SCIn_RXI interrupt request is generated when the stored number 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 SCR.RIE 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, an SCIn_ERI interrupt request is generated. When the amount of data stored in the FRDRL register is at the threshold value or above, an 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.

Note 1. In the case where 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.

Table 29.25 SCI interrupt sources with non-FIFO selected

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation	DMAC activation
SCIn_ERI	Receive error*1	ORER, FER, PER, DFER, DPER	RIE	Not possible	Not possible
SCIn_RXI	Receive data full	RDRF	RIE	Possible	Possible
	Address match	DCMF	RIE	Possible	Possible
SCIn_AM	Address match	DCMF	—	Possible	Possible
SCIn_TXI	Transmit data empty	TDRE	TIE	Possible	Possible
SCIn_TEI	Transmit end	TEND	TEIE	Not possible	Not possible

Note 1. The interrupt flag is only ORER when in clock synchronous and simple SPI mode.

Table 29.26 SCI interrupt sources with FIFO selected

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation	DMAC activation
SCIn_ERI	Receive error*1	ORER, FER, PER, DFER, DPER	RIE	Not possible	Not possible
		DR (when FCR.DRES = 1)	RIE	Not possible	Not possible
SCIn_RXI	Receive data full	RDF	RIE	Possible	Possible
	Receive data ready	DR (when FCR.DRES = 0)	RIE	Possible	Possible
	Address match	DCMF	RIE	Possible	Possible
SCIn_AM	Address match	DCMF	—	Possible	Possible
SCIn_TXI	Transmit data empty	TDFE	TIE	Possible	Possible
SCIn_TEI	Transmit end	TEND	TEIE	Not possible	Not possible

Note 1. The interrupt flag is only ORER when in clock synchronous and simple SPI mode.

29.10.4 Interrupts in Smart Card Interface Mode

Table 29.27 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 29.27 SCI interrupt sources: smart card interface mode

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation	DMAC activation
SCIn_ERI	Receive error or error signal detection	ORER, FER, ERS	RIE	Not possible	Not possible
SCIn_RXI	Receive data full	RDRF	RIE	Possible	Possible
SCIn_TXI	Transmit end	TEND	TIE	Possible	Possible

Data transmission or reception using the DMAC or DTC is also possible in smart card interface mode. In transmission, when the TEND flag in SSR_SMCI is set to 1, an SCIn_TXI interrupt request is generated. The SCIn_TXI interrupt request activates the DMAC or DTC allowing transfer of transmit data if the SCIn_TXI request is specified beforehand as a source of DMAC or DTC activation. The TEND flag is automatically set to 0 when the DMAC or DTC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept to 0 and the DMAC or DTC is not activated. Therefore, the SCI and DMAC or DTC automatically transmit the specified

number of bytes, including retransmission when errors occur. However, the ERS flag in SSR_SMCI is not automatically set to 0 at error occurrence. Therefore, the ERS flag must be cleared by setting the RIE bit in SCR_SMCI to 1 to enable an SCIn_ERI interrupt request to be generated at error occurrence.

When transmitting or receiving data using the DMAC or DTC, be sure to enable the DMAC or DTC before setting the SCI. For DMAC or DTC settings, see section 17, DMA Controller (DMAC) and section 18, Data Transfer Controller (DTC).

In reception, an SCIn_RXI interrupt request is generated when receive data is set to RDR. This SCIn_RXI interrupt request activates the DMAC or DTC allowing transfer of receive data if the SCIn_RXI request is previously specified as a source of DMAC or DTC activation. If an error occurs, the error flag is set. Therefore, the DMAC or DTC is not activated and an SCIn_ERI interrupt request is issued to the CPU instead. The error flag must be cleared.

29.10.5 Interrupts in Simple IIC Mode

Table 29.28 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 DMAC or 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 DMAC or DTC, the SCIn_RXI request activates the DMAC or 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 DMAC or DTC, the SCIn_TXI request activates the DMAC or 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 at the low level on the rising edge of the SCLn signal for the 9th bit (acknowledge bit). If the SCIn_RXI is previously set up as an activation source for the DMAC or DTC, the SCIn_RXI request activates the DMAC or DTC to handle transfer of the received data.
- An SCIn_TXI request (NACK detection) is generated if the input on the SDAn pin is at the high level on the rising edge of the SCLn signal for the 9th bit (acknowledge bit).

If the DMAC or DTC is used for data transfer in reception or transmission, be sure to set up and enable the DMAC or 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 29.28 SCI interrupt sources: simple IIC mode

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC activation	DMAC activation
SCIn_RXI	Reception, ACK detection	—	RIE	Possible	Possible
SCIn_TXI	Transmission, NACK detection	—	TIE	Possible	Possible
STIn	Completion of generation of a start, restart, or stop condition	IICSTIF	TEIE	Not possible	Not possible

Note: Activation of the DTC is only possible when the SIMR2.IICINTM bit is 1 (use reception and transmission interrupts).

29.11 Event Linking

By using interrupt request signals as event signals, the SCI can provide linked operation through the event link controller (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)

- 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
- Indicates that when SSR_FIFO.FER and PER flags are 0, and receive data less than the receive FIFO data trigger number is in the receive FIFO buffer, 15 ETUs elapse when FIFO selected and FCR.DRES is 1.

(2) Receive data full event output

- Indicates that ACK is detected if SIMR2.IICINTM is 0 in simple IIC mode
- Indicates that the 8th bit SCLn falling edge is detected if SIMR2.IICINTM 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.

(a) Non-FIFO selected

- Indicates that received data is in the Receive Data Register (RDR or RDRHL).

(b) FIFO selected

- Using this event output is prohibited.

(3) Transmit data empty event output

- Indicates that the SCR/SCR_SMCI.TE bit changed from 0 to 1
- Indicates that transmission is complete in smart card interface mode
- Indicates that NACK is detected if SIMR2.IICINTM is 0 in simple IIC mode
- Indicates that the 9th bit SCLn falling edge is detected if SIMR2.IICINTM is 1 in simple IIC mode.

(a) Non-FIFO selected

- Indicates that transmit data is transferred from the Transmit Data Register (TDR or TDRHL) to the Transmit Shift Register (TSR).

(b) FIFO selected

- Using this event output is prohibited.

(4) Transmit end event output

- 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

- Indicates a match of the comparison data (CDR.CMPD) with one frame of receive data when DCCR.DCME is 1 in asynchronous mode, including multi-processor mode.

29.12 Address non-match event output (SCI0_DCUF)

SCI0_DCUF indicates the non-match of comparison data (CDR.CMPD) with one frame of receive data when DCCR.DCME is set to 1 in asynchronous mode, including multi-processor mode.

29.13 Noise Cancellation Function

Figure 29.72 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 a match occurs, 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.

When SEMR.ABCS = 0 and SEMR.ABCSE = 0, the cycle is 1/16 the period of 1 transfer bit.

When SEMR.ABCS = 1 and SEMR.ABCSE = 0, the cycle is 1/8 the period of 1 transfer bit.

When SEMR.ABCSE = 1, the cycle is 1/6 the period of 1 transfer bit.

In asynchronous mode, the noise cancellation function can be applied to the receive signal input on the RXDn pin. The receive level of RXDn is sampled in the flip-flop circuit of the noise filter on the base clock of asynchronous mode.

In simple IIC mode, this function can be used for each input on SDAn and SCLn. The sampling clock for the noise cancellation function is selected in the SNFR.NFCS bit by dividing the baud rate generator source clock by 1, 2, 4, or 8.

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.

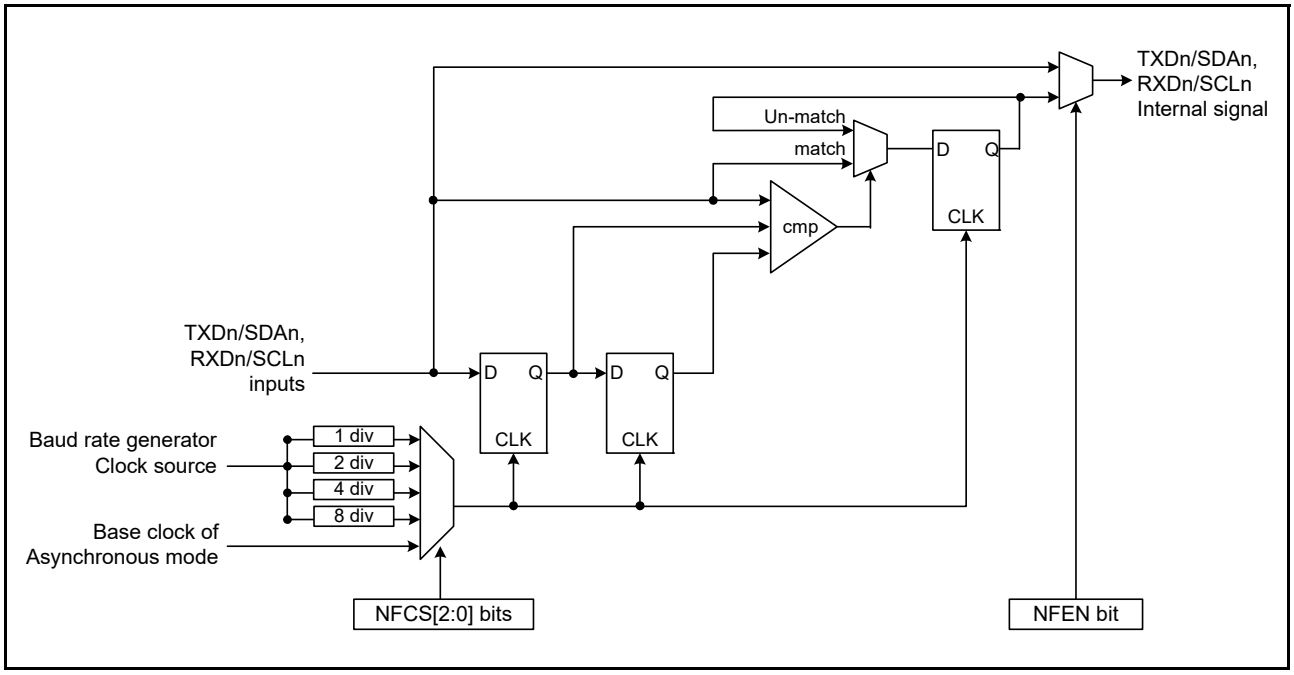


Figure 29.72 Digital noise filter circuit block diagram

29.14 Usage Notes

29.14.1 Settings for the Module-Stop State

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 11, Low Power Modes.

29.14.2 SCI Operations 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 the I/O port as an SCI function, the SPTR register can control the state of the TXDn pin. Setting the TE bit to 0 initializes TSR. The TEND bit in the SSR/SSR_SMCI is initialized to 1 with non-FIFO selected. The value is kept 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 consumption 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 consumption 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 29.73 shows an example flow of transition to Software Standby mode during transmission. Figure 29.74 and Figure 29.75 show the port pin states during transition to Software Standby mode.

Before specifying the module-stop state or making a transition 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

(a) When address match function is not used as wakeup condition

Before specifying the module-stop state or making a transition 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 29.76 shows an example flow of transition to Software Standby mode during reception.

(b) When address match function is used as wakeup condition

Before specifying the module-stop state or making a transition to Software Standby mode:

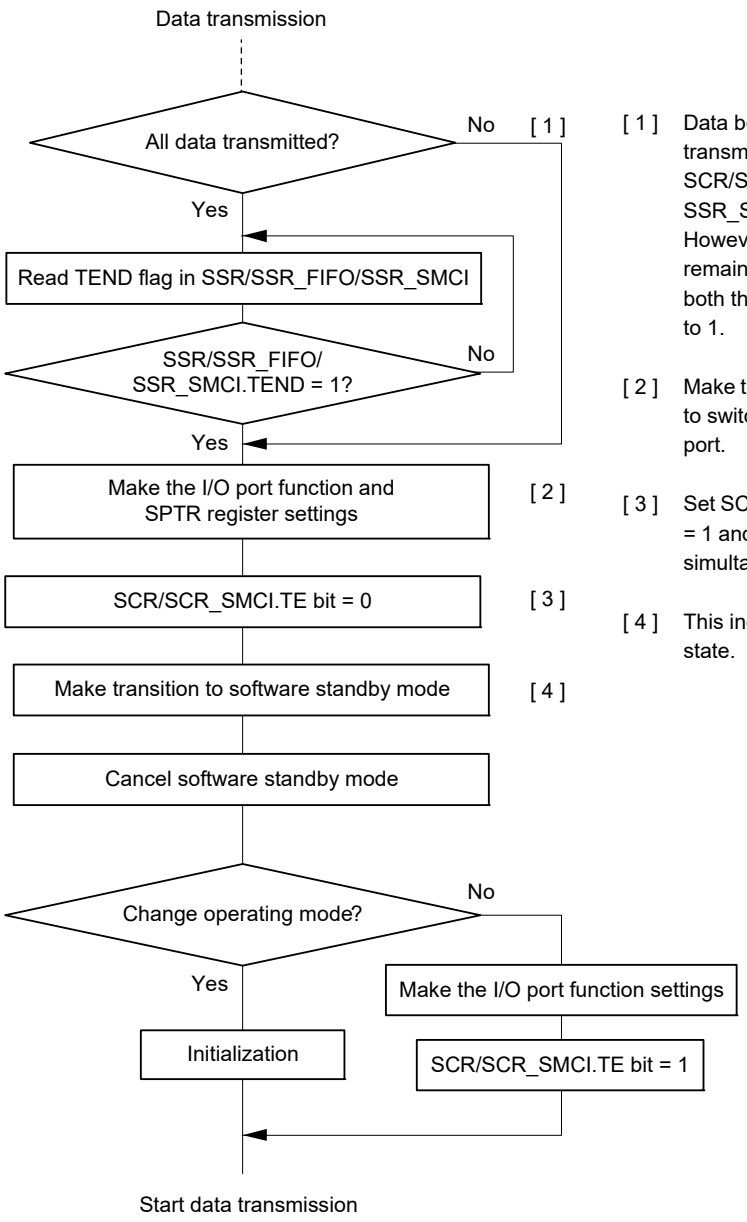
1. Set the operations after cancellation of the low power state.
2. Set CDR.CMPD and DCCR.DCME to 1.
3. Set the receive operations (RE = 1 in SCR/SCR_SMCI).
4. Set the module-stop state or Software Standby mode.

When the SCI transfers to the low power mode, if the receive data pin (RXD) is at the low level, set SEMR.RXDESEL to 0. If SEMR.RXDESEL is set to 1, there is a possibility that a start bit (fall edge of RXDn pin) cannot be detected on release of the low power mode.

Figure 29.77 shows an example flow of transition to Software Standby mode during reception with address match.

(c) When using SCI0 in Snooze mode

When using SCI0 in Snooze mode, some restrictions, including the maximum bit rates, exist. For details, see section 11, 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 Software Standby mode. However, if the DMAC or DTC is activated, the data remaining in the DMAC or DTC is transmitted when both the TE and TIE bits in SCR/SCR_SMCI are set to 1.

[2] Make 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 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 stopped state.

Figure 29.73 Example flow of transition to Software Standby mode during transmission

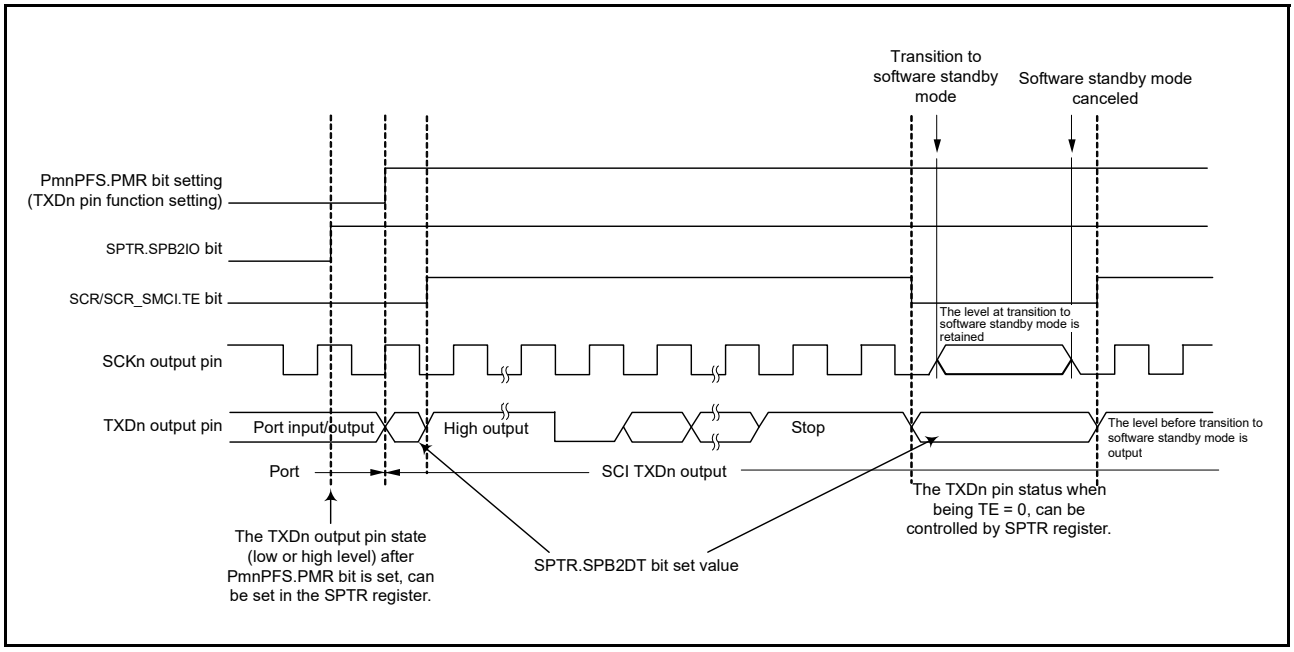


Figure 29.74 Port pin states during transition to Software Standby mode with internal clock and asynchronous transmission

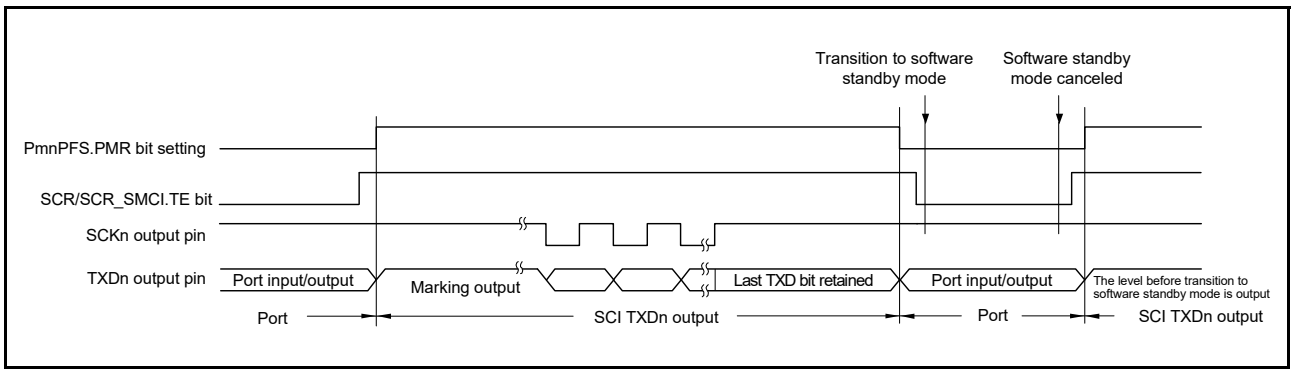


Figure 29.75 Port pin states during transition to Software Standby mode with internal clock and clock synchronous transmission

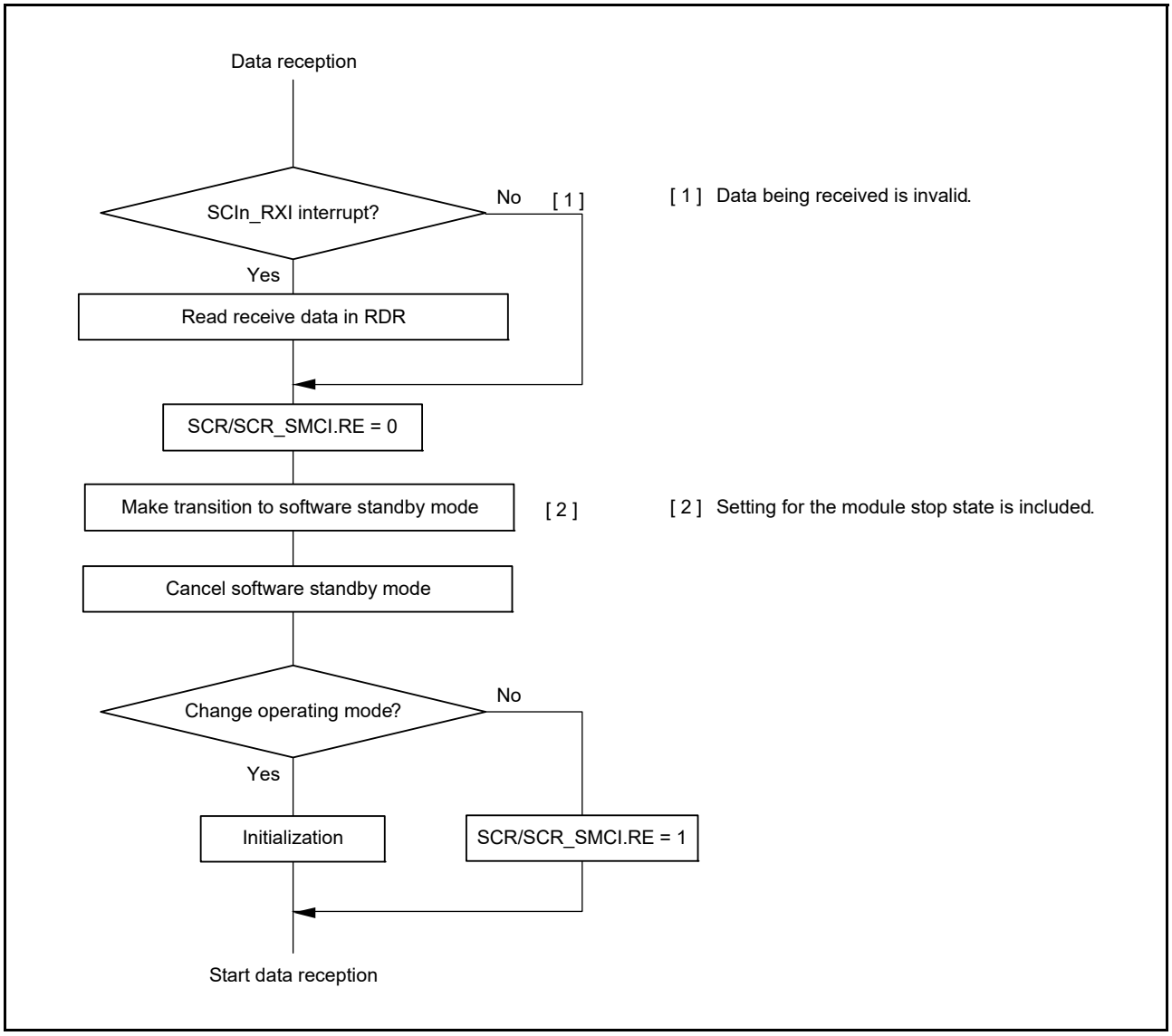


Figure 29.76 Example flow of transition to Software Standby mode during reception

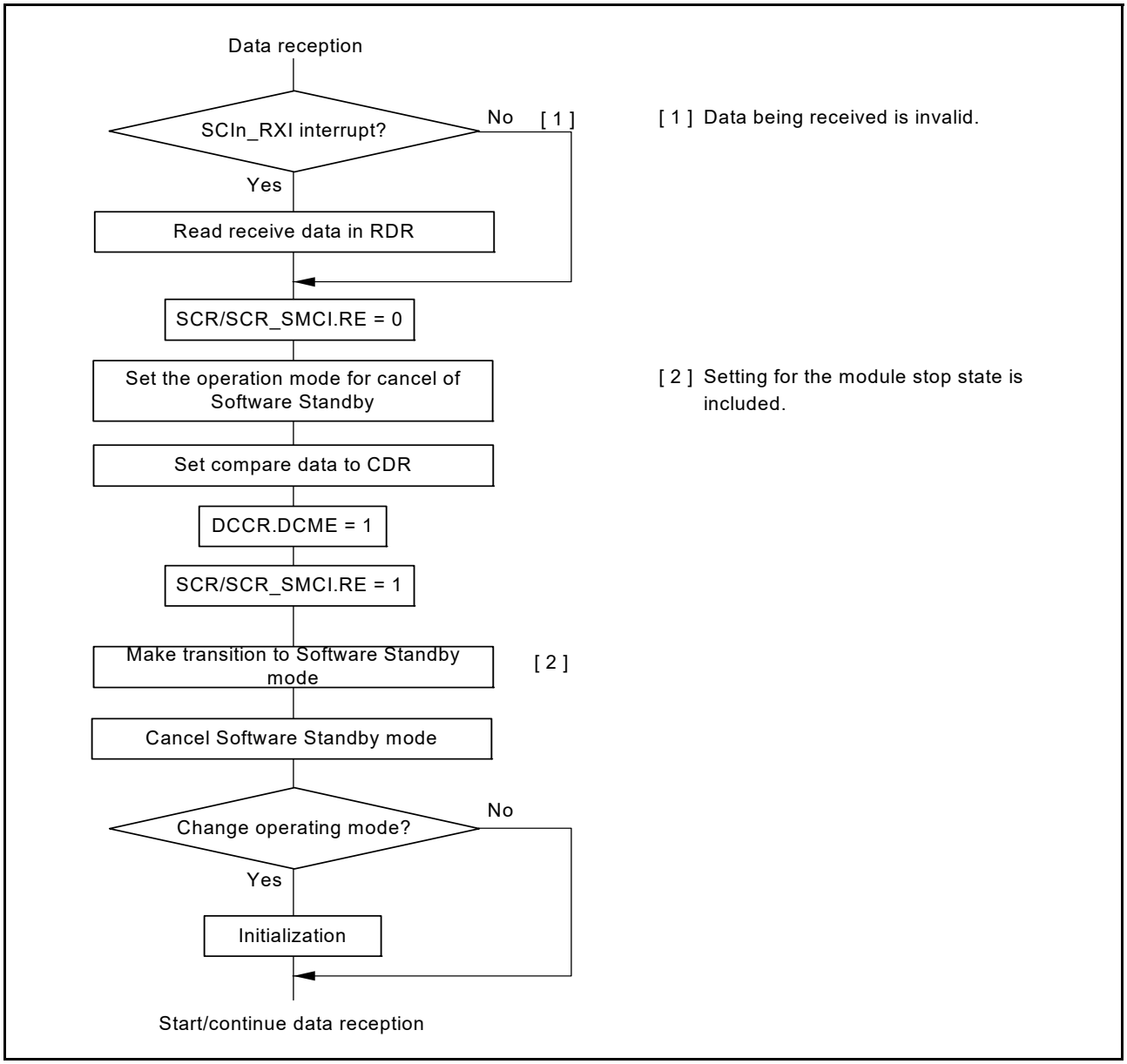


Figure 29.77 Example flow of transition to Software Standby mode during reception with address match

29.14.3 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 FER flag in SSR is set to 1 to indicate a framing error. The PER flag in SSR might also be set to 1 to indicate a parity error. The SCI continues the receive operation even after a break is received. Therefore, if the FER flag is set to 0, indicating 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 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 bit value. After the RXD signal is in the mark state and the break is finished, reception of data to FRDRHL resumes.

29.14.4 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 bit and SPTR.SPB2DT bit. 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 a 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.

29.14.5 Receive Error Flags and Transmit Operations 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*2. Be sure to set the receive error flags to 0 before starting transmission.

Note 1. The receive error flags cannot be set to 0 if the RE bit in SCR/SCR_SMCI is set to 0 (serial reception is disabled).
Note 2. Do not use the FTDRH register in simple SPI mode.

29.14.6 Restrictions on Clock Synchronous Transmission in Clock Synchronous and Simple SPI Modes

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 (tDO) + setup time for the master (tSU). See Figure 29.78.

(2) Continuous transmission

Write the next transmit data to TDR or TDRHL before the falling edge of the transmit clock, bit [7], see Figure 29.78.

When updating TDR after bit [7] starts 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 29.78.

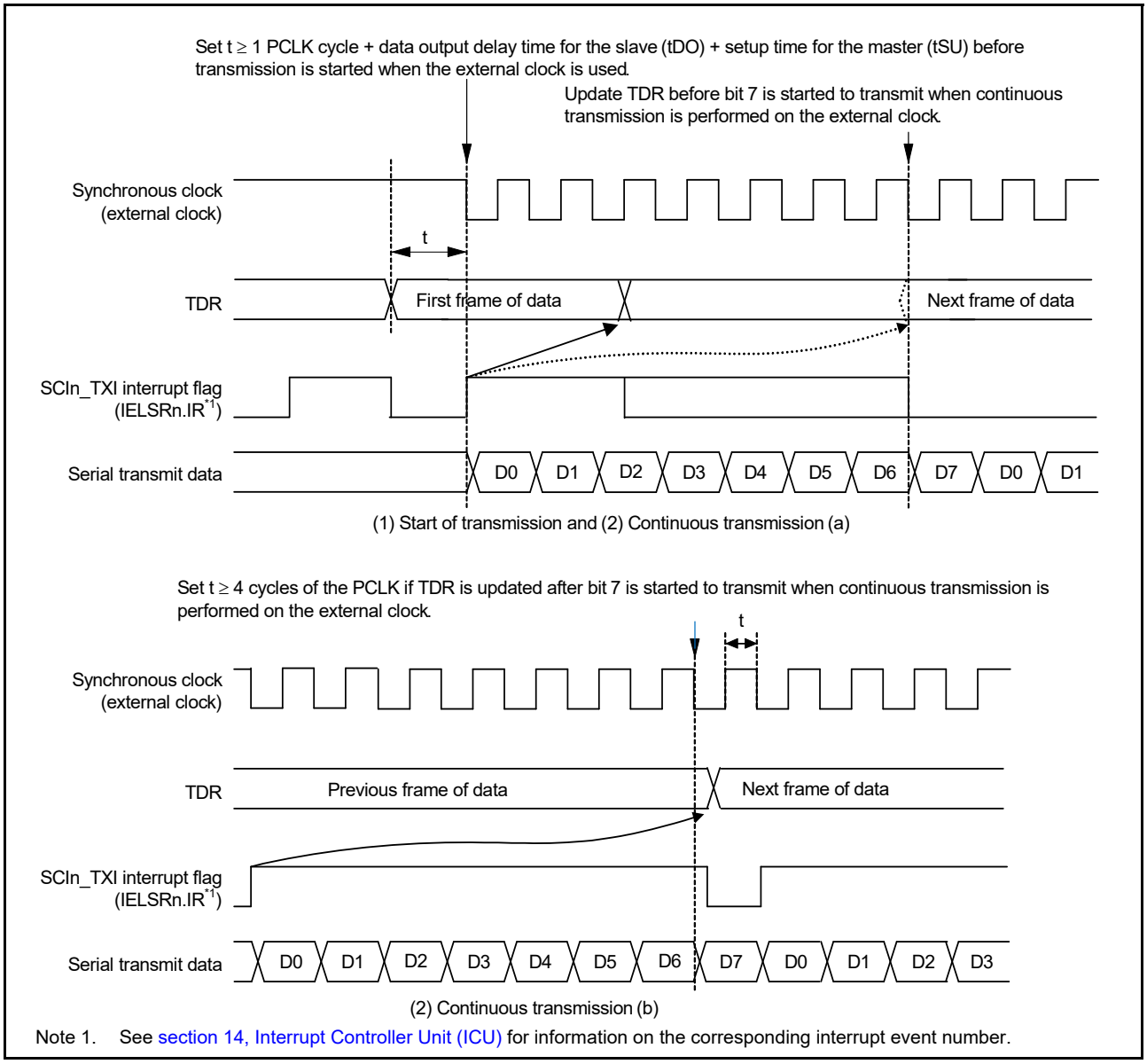


Figure 29.78 Restrictions on the use of external clock in clock synchronous transmission

29.14.7 Restrictions on Using DMAC or DTC

During transmission/reception operations using the DMAC or DTC, do not transfer data for the DMAC or DTC.

(1) Writing data to TDR (FTDRHL)

(a) 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 DMAC or DTC, be sure to write transmit data to TDR or TDRHL in the SCIn_TXI interrupt request handling routine.

(b) 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 DMAC or DTC to read RDR and RDRHL, be sure to set the receive data full interrupt (SCIn_RXI) as the activation source of the relevant SCI channel.

29.14.8 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 14, Interrupt Controller Unit (ICU).

1. Confirm that transfer 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) 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.

29.14.9 External Clock Input in Clock Synchronous and Simple SPI Modes

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.

29.14.10 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 by 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 is changed 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 29.79. If the TE and RE bits in the SCR 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 the current 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.

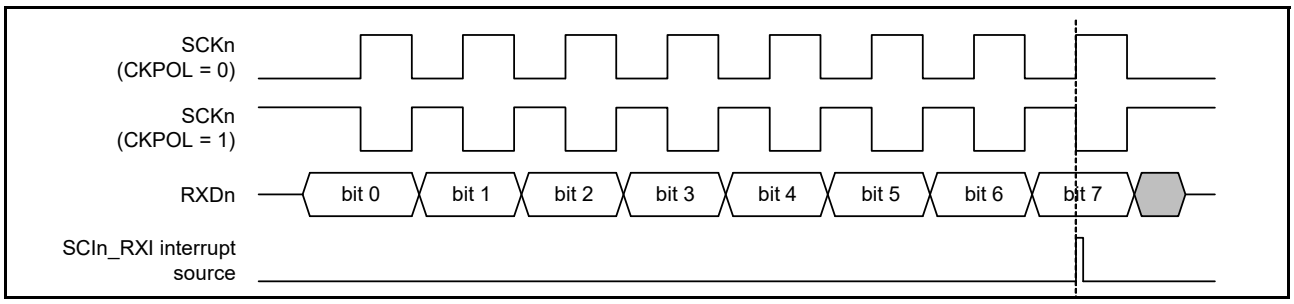


Figure 29.79 Timing of SCIn_RXI interrupt in simple SPI mode with clock delay

(2) Slave Mode

- Wait at least the following time from writing transmit data to TDR to the start of the external clock input:
  1 PCLK cycle + data output delay time for the slave (tDO) + setup time for the master (tSU).
  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 for 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 changes from low to high while a character is being transferred, set the TE and RE bits in SCR to 0 and, after restoring the settings, restart transfer of the first byte.

30. IrDA Interface

30.1 Overview

The IrDA interface sends and receives IrDA data communication waveforms in cooperation with the SCI1 based on the IrDA (Infrared Data Association) standard 1.0.

Enabling the IrDA function in the IRE bit in the IRCR register allows encoding and decoding of the TXD1 and RXD1 signals of the SCI1 to the waveforms conforming to the IrDA standard 1.0 (IRTXD1 and IRRXD1 pins). Connecting the waveforms to an infrared transmitter/receiver implements infrared data communication conforming to the IrDA standard 1.0 system.

With the IrDA standard 1.0 system, data transfer can be started at 9600 bps and the transfer rate can be changed whenever necessary. Because the IrDA interface cannot change the transfer rate automatically, the transfer rate must be changed through software.

Figure 30.1 shows interaction between the IrDA and SCI1 and Table 30.1 lists the I/O pins.

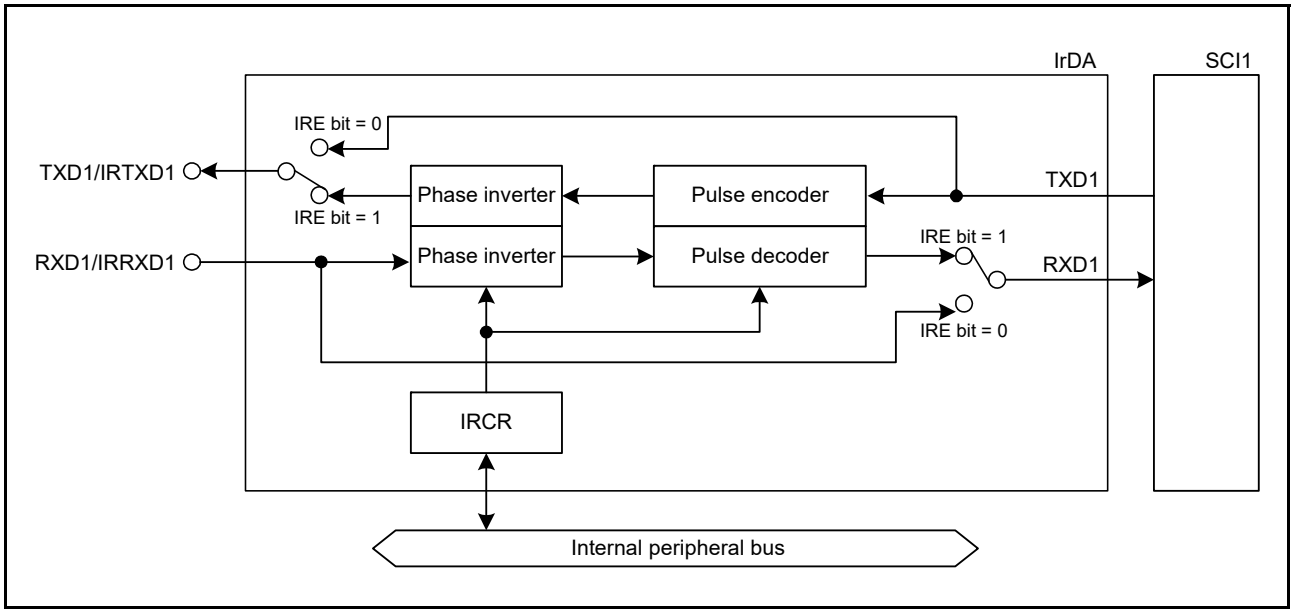


Figure 30.1 Interaction between IrDA and SCI1

Table 30.1 IrDA interface I/O pins

Pin Name	I/O	Function
IRTXD1	Output	Data to be transmitted
IRRXD1	Input	Received data

30.2 Register Descriptions

30.2.1 IrDA Control Register (IRCR)

Address(es): IRDA.IRCR 4007 0F00h

	b7	b6	b5	b4	b3	b2	b1	b0
	IRE	—	—	—	IRTXINV	IRRXINV	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b2	IRRXINV	IRRXD Polarity Switching	0: IRRXD input is used as received data as is 1: IRRXD input is used as received data after the polarity is inverted.	R/W
b3	IRTXINV	IRTXD Polarity Switching	0: Data to be transmitted is output to IRTXD as is 1: Data to be transmitted is output to IRTXD after the polarity is inverted.	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	IRE	IrDA Enable	0: Serial I/O pins are used for normal serial communication 1: Serial I/O pins are used for IrDA data communication.	R/W

Note: The IRCR register values are retained in Sleep and Software Standby modes.

IRRXINV bit (IRRXD Polarity Switching)

The IRRXINV bit inverts the logic level of the IRRXD input. When inverted, the high-level pulse width is applied to the low-level pulse width.

IRTXINV bit (IRTXD Polarity Switching)

The IRTXINV bit inverts the logic level of the IRTXD output. When inverted, the high-level pulse width is applied to the low-level pulse width.

IRE bit (IrDA Enable)

The IRE bit configures I/O pins for normal communication mode or IrDA data communication mode.

30.3 Operation

30.3.1 IrDA Interface Setup Procedure

To set up the IrDA interface operation:

1. Set the associated pins to IRTXD1 and IRRXD1 in the Pin Function Control Register (PmnPFS.PSEL[4:0] = 00101b) of the I/O port function.
2. Specify the peripheral function in the Pin Function Control Register (PmnPFS.PMR = 1) of the I/O port function.
3. Specify the IrDA function in the IRCR register.
4. Set the SCI1-related registers for the Serial Communications Interface (SCI).

30.3.2 Transmission

During transmission, the signals output from the SCI1 (UART frames) are converted to the IR frame data through the IrDA interface (see Figure 30.2). When the IRCR.IRTXINV bit is 0 and serial data is 0, high-level pulses with 3/16 the width of the bit rate (1-bit width period) are output (initial setting). The standard prescribes that the minimum high-level pulse width should be 1.41 µs and the maximum high-level pulse width should be (3/16 + 2.5%) × bit rate or (3/16 × bit rate) + 1.08 µs. When the serial data is 1, no pulses are output.

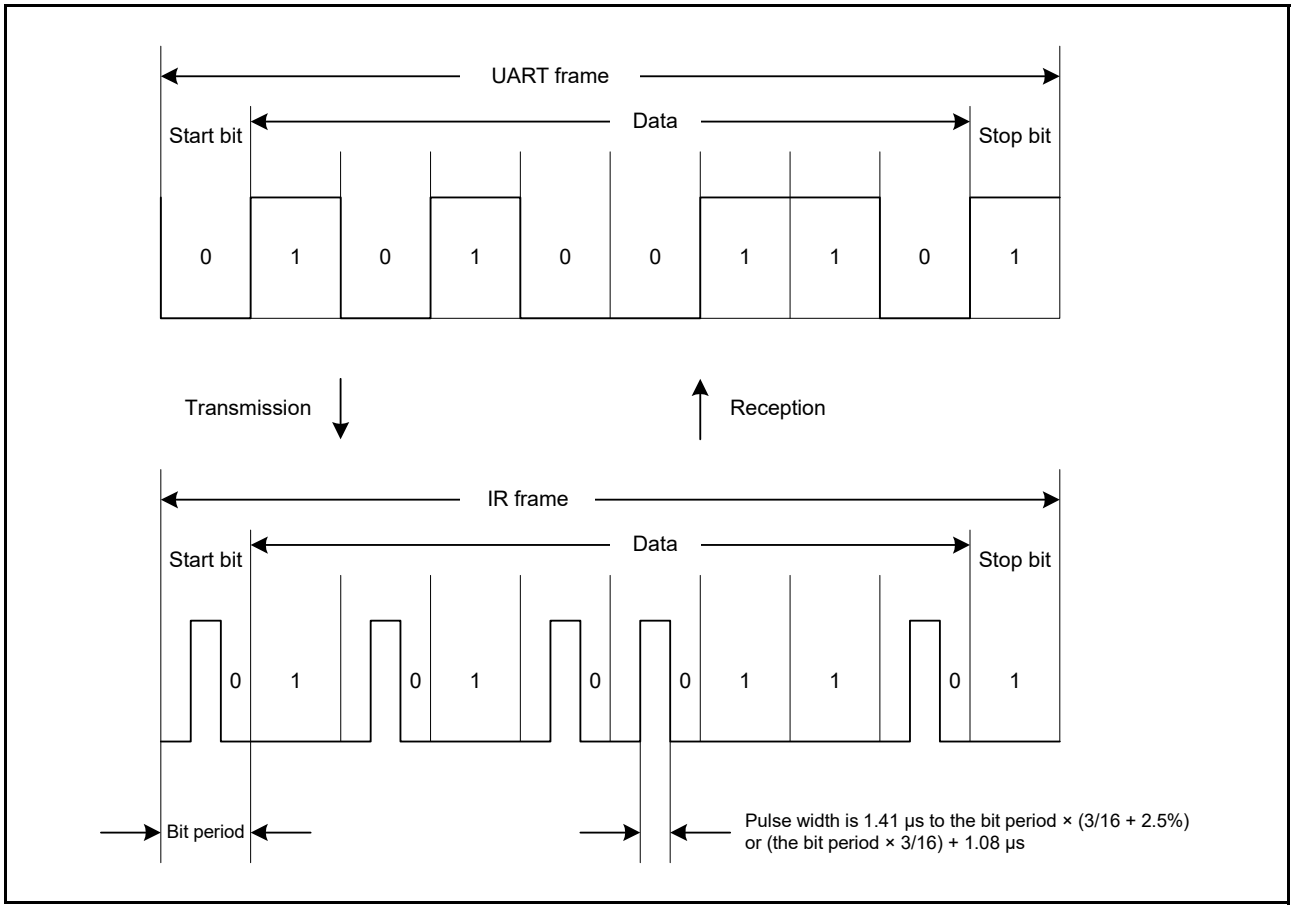


Figure 30.2 IrDA transmission and reception

30.3.3 Reception

During reception, the IR frame data is converted to the UART frame data through the IrDA interface and is input to the SCI1. Low-level data is input to SCI1 when the IRCR.IRRXINV bit is 0 and a high-level pulse is detected. High-level data is input to SCI1 when no pulse is detected for a 1-bit period.

30.4 Usage Notes

30.4.1 Module Stop Function Setting

The IrDA is stopped after a reset. Registers can be accessed by releasing the module-stop state. For details, see section 11, Low Power Modes.

30.4.2 Asynchronous Reference Clock for SCI1

The IrDA receives a clock with a frequency 16 times the bit rate from SCI1 and operates in conjunction with SCI1. When using the IrDA, set the SCI1.SEMR.ABCS bit to 0.

31. I2C Bus Interface (IIC)

31.1 Overview

The MCU has a three-channel I2C Bus Interface (IIC). The IIC module conforms with and provides a subset of the NXP I2C bus (Inter-Integrated Circuit bus) interface functions. Table 31.1 lists the IIC specifications, Figure 31.1 shows the block diagram, and Figure 31.2 shows an example of I/O pin connections to external circuits, with an I2C bus configuration example. Table 31.2 lists the I/O pins.

Table 31.1 IIC specifications (1 of 2)

Parameter	Description
Communications format	• I2C bus format or SMBus format • Master mode 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-bit 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, 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 8th and 9th clock cycles • Waiting between the 9th 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 would create conflict on the bus, loss of arbitration is detected by testing for non-matching 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 non-matching 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 occurrence of events: arbitration detection, 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 31.1 IIC specifications (2 of 2)

Parameter	Description
Event link function (output)	• Transfer error or occurrence of events: arbitration detection, 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*1	• CPU can return from a Software Standby mode using a wakeup event

Note 1. This function is only available for IIC channel IIC0.

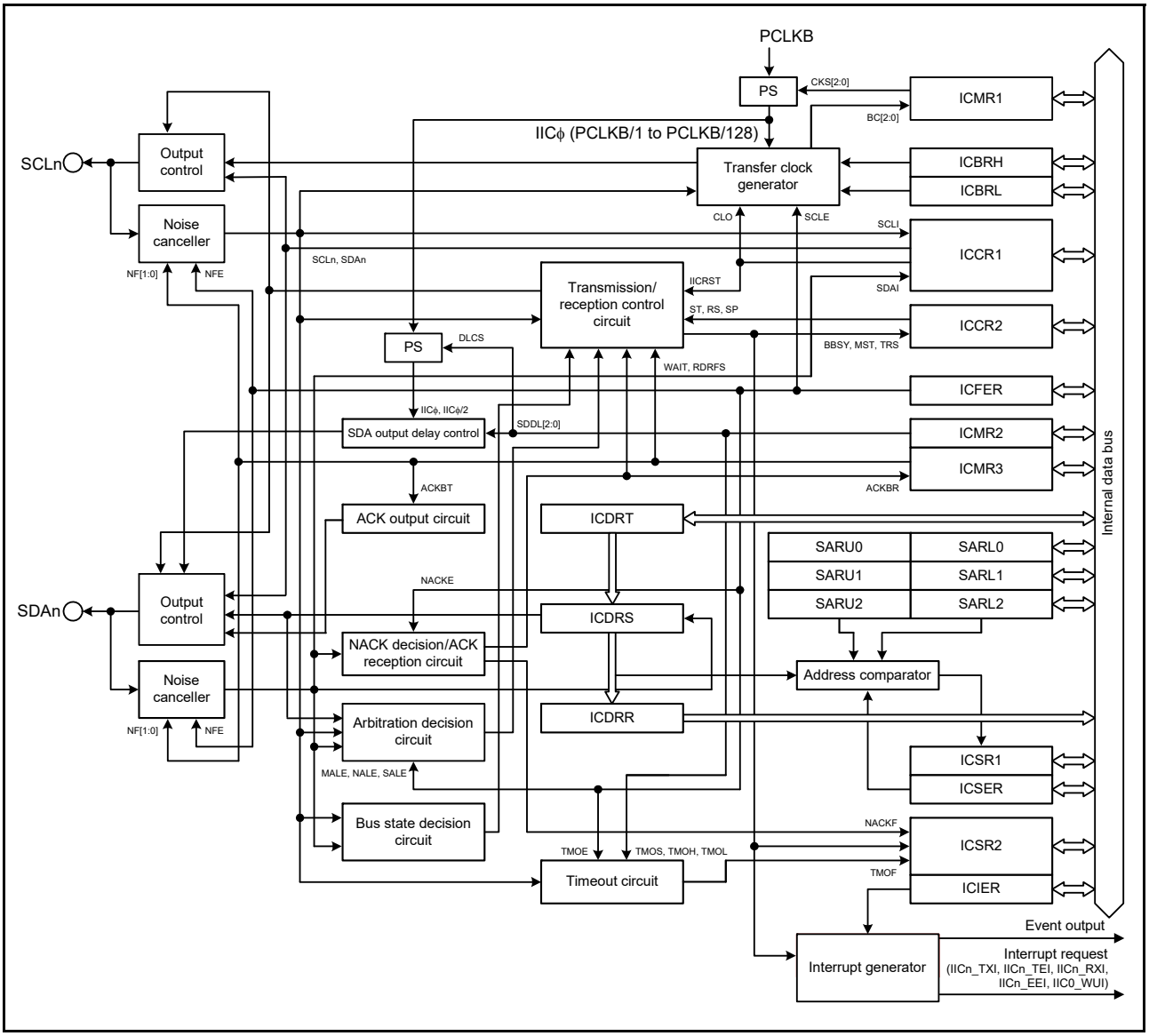


Figure 31.1 IIC block diagram

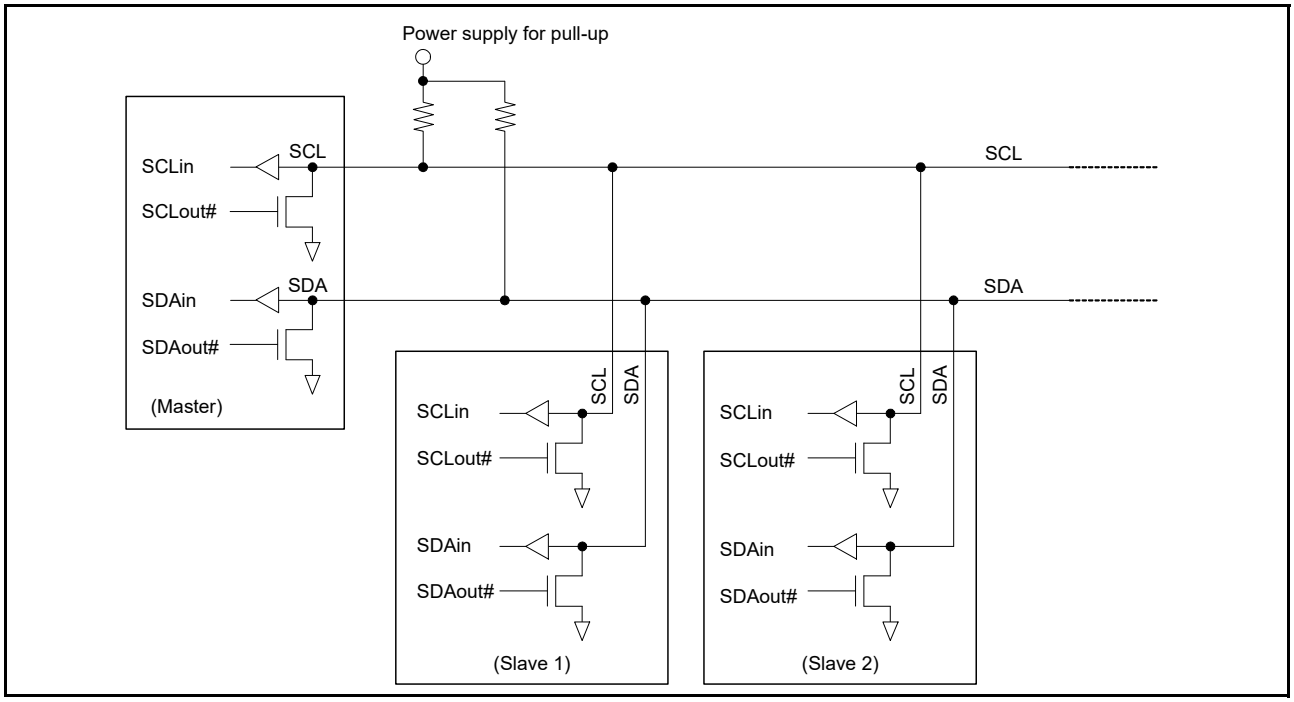


Figure 31.2 I/O pin connection to the external circuit (I2C bus configuration example)

The input level of the signals for IIC is CMOS when I2C bus is selected (ICMR3.SMBS = 0), or TTL when SMBus is selected (ICMR3.SMBS = 1).

Table 31.2 IIC pin configuration

Channel	Pin name	I/O	Function
IIC0	SCL0	I/O	IIC0 serial clock I/O pin
	SDA0	I/O	IIC0 serial data I/O pin
IIC1	SCL1	I/O	IIC1 serial clock I/O pin
	SDA1	I/O	IIC1 serial data I/O pin
IIC2	SCL2	I/O	IIC2 serial clock I/O pin
	SDA2	I/O	IIC2 serial data I/O pin

31.2 Register Descriptions

31.2.1 I2C Bus Control Register 1 (ICCR1)

Address(es): IIC0.ICCR1 4005 3000h, IIC1.ICCR1 4005 3100h, IIC2.ICCR1 4005 3200h

	b7	b6	b5	b4	b3	b2	b1	b0
	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b0	SDAI	SDA Line Monitor	0: SDAn line is low 1: SDAn line is high.	R
b1	SCLI	SCL Line Monitor	0: SCLn line is low 1: SCLn line is high.	R
b2	SDAO	SDA Output Control/Monitor	• Read: 0: IIC drove the SDAn pin low 1: IIC released the SDAn pin. • Write: 0: Drive SDAn pin low 1: Release SDAn pin.	R/W
b3	SCLO	SCL Output Control/Monitor	• Read: 0: IIC drove the SCLn pin low 1: IIC released the SCLn pin. • Write: 0: Drive SCLn pin low 1: Release SCLn pin. Use an external pull-up resistor to drive the signal high.	R/W
b4	SOWP	SCLO/SDAO Write Protect	0: Write enable SCLO and SDAO bits 1: Write protect SCLO and SDAO bits. This bit is read as 1.	R/W
b5	CLO	Extra SCL Clock Cycle Output	0: Do not output extra SCL clock cycle (default) 1: Output extra SCL clock cycle. This bit clears automatically after one clock cycle is output.	R/W
b6	IICRST	IIC Bus Interface Internal Reset	0: Release IIC reset or internal reset 1: Initiate IIC reset or internal reset. This clears the bit counter and the SCLn/SDAn output latch.	R/W
b7	ICE	IIC Bus Interface Enable	0: Disable (SCLn and SDAn pins in inactive state) 1: Enable (SCLn and SDAn pins in active state). Used in combination with the IICRST bit to select either IIC or internal reset.	R/W

SDAO bit (SDA Output Control/Monitor) and SCLO bit (SCL Output Control/Monitor)

The SDAO and SCLO bits directly control the SDAn and SCLn 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, or during transmission or reception. Operation after rewriting under the above 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 this bit to 1 in a normal communication state causes a communication error. For details on this function, see section 31.12.2, Extra SCL Clock Cycle Output Function.

IICRST bit (IIC 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 setting this bit in combination with the ICE bit. Table 31.3 lists the IIC resets.

The IIC reset initializes all registers except ICCR1.ICE and ICCR1.ICCRST 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.SCLO and ICCR1.SDAO bits)
- I2C Bus Control Register 2 (except ICCR2.BBSY bit).

For the reset conditions for each register, see section 31.15, Register States 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 up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the SCLn pin and SDAn 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 device and the master device 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 necessary because the IIC hangs with the SCLn line in a low level output state in slave mode, initiate an internal reset, and 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 condition or restart condition from the master device, synchronization is lost because the master and slave devices operate asynchronously.

Table 31.3 IIC resets

IICRST	ICE	State	Specifications
1	0	IIC reset	Resets all registers except ICCR1.ICE and ICCR1.ICCRST bits and internal states of the IIC
	1	Internal reset	Resets the ICMR1.BC[2:0] bits, the ICSR1, ICSR2, and ICDRS registers, the ICCR1.SCLO and ICCR1.SDAO bits, the ICCR2 register (except ICCR2.BBSY bit), and the internal states of the IIC

ICE bit (IIC Bus Interface Enable)

The ICE bit selects the active or inactive state of the SCLn and SDAn pins. It can also be combined with the IICRST bit to initiate one of the two resets. See Table 31.3 for the reset types.

Set the ICE bit to 1 when using the IIC. The SCLn and SDAn 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 SCLn and SDAn pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCLn or SDAn pin to the IIC when setting up the pin function control. Slave address comparison is performed if the pins are assigned to the IIC.

31.2.2 I2C Bus Control Register 2 (ICCR2)

Address(es): IIC0.ICCR2 4005 3001h, IIC1.ICCR2 4005 3101h, IIC2.ICCR2 4005 3201h

	b7	b6	b5	b4	b3	b2	b1	b0
	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	ST	Start Condition Issuance Request	0: Do not issue a start condition request 1: Issue a start condition request.	R/W
b2	RS	Restart Condition Issuance Request	0: Do not issue a restart condition request 1: Issue a restart condition request.	R/W
b3	SP	Stop Condition Issuance Request	0: Do not issue a stop condition request 1: Issue a stop condition request.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	TRS	Transmit/Receive Mode	0: Receive mode 1: Transmit mode.	R/W*1
b6	MST	Master/Slave Mode	0: Slave mode 1: Master mode.	R/W*1
b7	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 issues 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 issuing a start condition, see section 31.11, 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 initiate an IIC reset or an internal reset.

Note: 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 issuing a restart condition, see section 31.11, 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 initiate 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 RS bit being cleared, the 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 issuing a stop condition, see section 31.11, 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 initiate 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 set to 0 and in transmit mode when the TRS bit is set to 1. The combination of the TRS bit and the MST bit indicates the operating mode of the IIC.

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 the TRS 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 start 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 initiate 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 set to 0 and is in master mode when the MST bit is set to 1. The combination of the MST bit and the TRS bit indicates the operating mode of the IIC.

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 when 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 the MST 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 initiate an IIC reset or an internal reset.

BBSY flag (Bus Busy Detection Flag)

The BBSY flag indicates whether the I2C bus is occupied (bus busy state) or released (bus free state).

This flag is set to 1 when the SDAn line changes from high to low when the SCLn line is high, assuming that a start condition was issued.

This flag is set to 0 if the bus free time (ICBRL setting) start condition is not detected, assuming that a stop condition was issued.

[Setting condition]

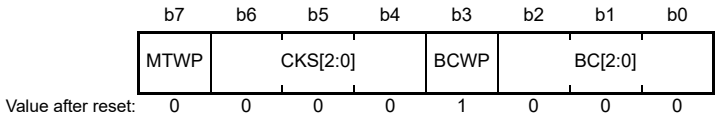
- When a start condition is detected.

[Clearing conditions]

- When the bus free time (ICBRL setting) start condition is not detected 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).

31.2.3 I2C Bus Mode Register 1 (ICMR1)

Address(es): IIC0.ICMR1 4005 3002h, IIC1.ICMR1 4005 3102h, IIC2.ICMR1 4005 3202h



Bit	Symbol	Bit name	Description	R/W
b2 to b0	BC[2:0]	Bit Counter	b2 b0 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
b3	BCWP	BC Write Protect	0: Write enable BC[2:0] bits 1: Write protect BC[2:0] bits. This bit is read as 1.	R/W*1
b6 to b4	CKS[2:0]	Internal Reference Clock Select	Select the internal reference clock source (IICϕ) for the IIC. b6 b4 0 0 0: PCLKB clock 0 0 1: PCLKB/2 clock 0 1 0: PCLKB/4 clock 0 1 1: PCLKB/8 clock 1 0 0: PCLKB/16 clock 1 0 1: PCLKB/32 clock 1 1 0: PCLKB/64 clock 1 1 1: PCLKB/128 clock.	R/W
b7	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 that indicates the number of bits remaining to be transferred on detection of a rising edge on the SCLn line. Although the BC[2:0] bits are writable and readable, it is not necessary 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 SCLn line is at a low level.

The values of the BC[2:0] bits return to 000b at the end of a data transfer including the acknowledge bit, or when a start or restart condition is detected.

31.2.4 I2C Bus Mode Register 2 (ICMR2)

Address(es): IIC0.ICMR2 4005 3003h, IIC1.ICMR2 4005 3103h, IIC2.ICMR2 4005 3203h

	b7	b6	b5	b4	b3	b2	b1	b0
	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Bit name	Description	R/W
b0	TMOS	Timeout Detection Time Select	0: Select long mode 1: Select short mode.	R/W
b1	TMOL	Timeout L Count Control	0: Disable count while the SCLn line is low 1: Enable count while the SCLn line is low.	R/W
b2	TMOH	Timeout H Count Control	0: Disable count while the SCLn line is high 1: Enable count while the SCLn line is high.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	SDDL[2:0]	SDA Output Delay Counter	• When ICMR2.DLCS = 0 (IICϕ) b6 b4 0 0 0: No output delay 0 0 1: 1 IICϕ cycle 0 1 0: 2 IICϕ cycles 0 1 1: 3 IICϕ cycles 1 0 0: 4 IICϕ cycles 1 0 1: 5 IICϕ cycles 1 1 0: 6 IICϕ cycles 1 1 1: 7 IICϕ cycles. • When ICMR2.DLCS = 1 (IICϕ/2) b6 b4 0 0 0: No output delay 0 0 1: 1 or 2 IICϕ cycles 0 1 0: 3 or 4 IICϕ cycles 0 1 1: 5 or 6 IICϕ cycles 1 0 0: 7 or 8 IICϕ cycles 1 0 1: 9 or 10 IICϕ cycles 1 1 0: 11 or 12 IICϕ cycles 1 1 1: 13 or 14 IICϕ cycles.	R/W
b7	DLCS	SDA Output Delay Clock Source Select	0: Internal reference clock (IICϕ) selected as the clock source for the SDA output delay counter 1: Internal reference clock divided by 2 (IICϕ/2) selected as the clock source for the 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 setting DLCS = 1 is invalid and the clock source becomes the internal reference clock (IICϕ).

TMOS bit (Timeout Detection Time Select)

The TMOS bit selects long mode or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE bit = 1). When this bit is set to 0, long mode is selected. When the TMOS bit 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 SCLn 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 the timeout function, see section 31.12.1, 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 SCLn line is held low and the timeout function is enabled (ICFER.TMOE bit = 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 SCLn line is held high and the timeout function is enabled (ICFER.TMOE bit = 1).

SDDL[2:0] bits (SDA Output Delay Counter)

The SDA output can be delayed using the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. This function setting can be used for all types of SDA output, including the transmission of the acknowledge bit.

Set the SDA output delay time 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 31.5, 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)

31.2.5 I2C Bus Mode Register 3 (ICMR3)

Address(es): IIC0.ICMR3 4005 3004h, IIC1.ICMR3 4005 3104h, IIC2.ICMR3 4005 3204h

	b7	b6	b5	b4	b3	b2	b1	b0
	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	NF[1:0]	Noise Filter Stage Select	b1 b0 0 0: Noise of up to one IICϕ cycle filtered out (single-stage filter) 0 1: Noise of up to two IICϕ cycles filtered out (2-stage filter) 1 0: Noise of up to three IICϕ cycles filtered out (3-stage filter) 1 1: Noise of up to four IICϕ cycles filtered out (4-stage filter).	R/W
b2	ACKBR	Receive Acknowledge	0: 0 received as the acknowledge bit (ACK reception) 1: 1 received as the acknowledge bit (NACK reception).	R
b3	ACKBT	Transmit Acknowledge	0: 0 sent as the acknowledge bit (ACK transmission) 1: 1 sent as the acknowledge bit (NACK transmission).	R/W*1
b4	ACKWP	ACKBT Write Protect	0: Write protect the ACKBT bit 1: Write enable the ACKBT bit.	R/W*1
b5	RDRFS	RDRF Flag Set Timing Select	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. Low-hold is released by writing to ACKBT.	R/W*2
b6	WAIT	WAIT	0: No WAIT SCLn is not held low during the period between 9th clock cycle and 1st clock cycle. 1: WAIT SCLn is held low during the period between 9th clock cycle and 1st clock cycle. Low-hold is released by reading ICDRR.	R/W*2
b7	SMBS	SMBus/I2C Bus Select	0: I2C bus is selected 1: SMBus is selected.	R/W

Note 1. Write to the ACKBT bit only when the ACKWP bit is already 1. If the application writes 1 to both 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 valid only 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 the digital noise filter function, see section 31.6, Digital Noise Filter Circuits.

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCLn line high-level period or low-level period. If the noise range is set to a value of [SCL clock width: high-level period 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 issuance is detected (when a stop condition 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 SCLn line low on the falling edge of the 8th SCL clock cycle.

When the RDRFS bit is 0, the SCLn 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 SCLn line is held low on the falling edge of the 8th SCL clock cycle. The low-hold of the SCLn line is released by a write to the ACKBT bit.

After data is received with this setting, the SCLn 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 hold the period between the 9th SCL clock cycle and the 1st SCL clock cycle low 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 is continued without holding the period between the 9th and the 1st SCL clock cycle low. 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 SCLn line is held low from the falling edge of the 9th clock cycle until the ICDRR value is read each time a single-byte of data is received. This enables receive operation in byte units.

Note: When the WAIT bit value is to be read, be sure to first read the ICDRR.

SMBS bit (SMBus/I2C Bus Select)

Setting the SMBS bit to 1 selects the SMBus and enables the HOAE bit in ICSER.

31.2.6 I2C Bus Function Enable Register (ICFER)

Address(es): IIC0.ICFER 4005 3005h, IIC1.ICFER 4005 3105h, IIC2.ICFER 4005 3205h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Bit name	Description	R/W
b0	TMOE	Timeout Function Enable	0: Timeout function disabled 1: Timeout function enabled.	R/W
b1	MALE	Master Arbitration-Lost Detection Enable	0: Master arbitration-lost detection disabled. Also disables automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost. 1: Master arbitration-lost detection enabled. Also enables automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost.	R/W
b2	NALE	NACK Transmission Arbitration-Lost Detection Enable	0: NACK transmission arbitration-lost detection is disabled 1: NACK transmission arbitration-lost detection is enabled.	R/W
b3	SALE	Slave Arbitration-Lost Detection Enable	0: Slave arbitration-lost detection disabled 1: Slave arbitration-lost detection enabled.	R/W
b4	NACKE	NACK Reception Transfer Suspension Enable	0: Transfer operation not suspended during NACK reception (transfer suspension disabled) 1: Transfer operation suspended during NACK reception (transfer suspension enabled).	R/W
b5	NFE	Digital Noise Filter Circuit Enable	0: No digital noise filter circuit used 1: A digital noise filter circuit used.	R/W
b6	SCLE	SCL Synchronous Circuit Enable	0: No SCL synchronous circuit used 1: An SCL synchronous circuit used.	R/W
b7	—	Reserved	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 the timeout function, see section 31.12.1, Timeout Function.

MALE bit (Master Arbitration-Lost Detection Enable)

The MALE bit specifies whether to use the arbitration-lost detection function in master mode. For normal operation, 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 instance when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with 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 from the slave device in transmit mode. For normal operation, 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 is continued regardless of the received acknowledge content.

For details on the NACK reception transfer suspension function, see section 31.9.2, 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. For normal operation, 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. In this setting, the IIC outputs the SCL clock with the transfer rate set in ICBRH and ICBRL regardless of the SCLn line state. For this reason, if the bus load of the I2C 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.

The SCLE bit must not be set to 0 except for checking the output of the set transfer rate.

31.2.7 I2C Bus Status Enable Register (ICSER)

Address(es): IIC0.ICSER 4005 3006h, IIC1.ICSER 4005 3106h, IIC2.ICSER 4005 3206h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Bit name	Description	R/W
b0	SAR0E	Slave Address Register 0 Enable	0: Slave address in SARL0 and SARU0 disabled 1: Slave address in SARL0 and SARU0 enabled.	R/W
b1	SAR1E	Slave Address Register 1 Enable	0: Slave address in SARL1 and SARU1 disabled 1: Slave address in SARL1 and SARU1 enabled.	R/W
b2	SAR2E	Slave Address Register 2 Enable	0: Slave address in SARL2 and SARU2 disabled 1: Slave address in SARL2 and SARU2 enabled.	R/W
b3	GCAE	General Call Address Enable	0: General call address detection disabled 1: General call address detection enabled.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DIDE	Device ID Address Detection Enable	0: Device ID address detection disabled 1: Device ID address detection enabled.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOAE	Host Address Enable	0: Host address detection disabled 1: Host address detection enabled.	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 the SARyE bit is set to 1, the slave address set in SARLy and SARUy is enabled and is compared with the received slave address. When the SARyE 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 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 the DIDE 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 subsequent R/W# bit is 0 [W], the IIC recognizes the second and the subsequent frames as slave addresses and continues the receive operation. When the DIDE bit is set to 0, the IIC ignores the received first frame even if it matches the device ID address and recognizes the first frame as a normal slave address.

For details on the device ID address detection, see section 31.7.3, Device ID Address Detection.

HOAE bit (Host Address Enable)

The HOAE bit specifies whether to ignore 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.

31.2.8 I2C Bus Interrupt Enable Register (ICIER)

Address(es): IIC0.ICIER 4005 3007h, IIC1.ICIER 4005 3107h, IIC2.ICIER 4005 3207h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TMOIE	Timeout Interrupt Request Enable	0: Timeout interrupt (TMOIn) request disabled 1: Timeout interrupt (TMOIn) request enabled.	R/W
b1	ALIE	Arbitration-Lost Interrupt Request Enable	0: Arbitration-lost interrupt (ALIn) request disabled 1: Arbitration-lost interrupt (ALIn) request enabled.	R/W
b2	STIE	Start Condition Detection Interrupt Request Enable	0: Start condition detection interrupt (STIn) request disabled 1: Start condition detection interrupt (STIn) request enabled.	R/W
b3	SPIE	Stop Condition Detection Interrupt Request Enable	0: Stop condition detection interrupt (SPIn) request disabled 1: Stop condition detection interrupt (SPIn) request enabled.	R/W
b4	NAKIE	NACK Reception Interrupt Request Enable	0: NACK reception interrupt (NAKIn) request disabled 1: NACK reception interrupt (NAKIn) request enabled.	R/W
b5	RIE	Receive Data Full Interrupt Request Enable	0: Receive data full interrupt (IICn_RXI) request disabled 1: Receive data full interrupt (IICn_RXI) request enabled.	R/W
b6	TEIE	Transmit End Interrupt Request Enable	0: Transmit end interrupt (IICn_TEI) request disabled 1: Transmit end interrupt (IICn_TEI) request enabled.	R/W
b7	TIE	Transmit Data Empty Interrupt Request Enable	0: Transmit data empty interrupt (IICn_TXI) request disabled 1: Transmit data empty interrupt (IICn_TXI) request enabled.	R/W

TMOIE bit (Timeout Interrupt Request Enable)

The TMOIE bit enables or disables timeout interrupt (TMOIn) requests when the TMOF flag in ICSR2 is set to 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 (ALIn) requests when the AL flag in ICSR2 is set to 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 (STIn) requests when the START flag in ICSR2 is set to 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 (SPIn) requests when the STOP flag in ICSR2 is set to 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 (NAKIn) requests when the NACKF flag in ICSR2 is set to 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 (IICn_RXI) requests when the RDRF flag in ICSR2 is set to 1.

TEIE bit (Transmit End Interrupt Request Enable)

The TEIE bit enables or disables transmit end interrupt (IICn_TEI) requests when the TEND flag in ICSR2 is set to 1. To cancel an IICn_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 (IICn_TXI) requests when the TDRE flag in ICSR2 is set to 1.

31.2.9 I2C Bus Status Register 1 (ICSR1)

Address(es): IIC0.ICSR1 4005 3008h, IIC1.ICSR1 4005 3108h, IIC2.ICSR1 4005 3208h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	AAS0	Slave Address 0 Detection Flag	0: Slave address 0 not detected 1: Slave address 0 detected.	R/(W) *1
b1	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 not detected 1: Slave address 1 detected.	R/(W) *1
b2	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 not detected 1: Slave address 2 detected.	R/(W) *1
b3	GCA	General Call Address Detection Flag	0: General call address not detected 1: General call address detected.	R/(W) *1
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DID	Device ID Address Detection Flag	0: Device ID command not detected 1: Device ID command detected. 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]).	R/(W) *1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOA	Host Address Detection Flag	0: Host address not detected 1: Host address detected. This bit is set to 1 when the received slave address matches the host address (0001 000b).	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)

[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). This flag is set to 1 on the rising edge of the 9th 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). This flag is set to 1 on the rising edge of the 9th SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy bit after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or 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). This flag is set to 0 on the rising edge of the 9th 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). This flag is set to 0 on the rising edge of the 9th 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). This flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.

GCA flag (General Call Address Detection Flag)

[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 is enabled). This flag is set to 1 on the rising edge of the 9th SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA bit 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 is enabled). This flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

DID flag (Device ID Address Detection Flag)

[Setting condition]

- When the first frame received immediately after a start condition 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 is enabled). This flag is set to 1 on the rising edge of the 9th SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID bit 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 is enabled). This flag is set to 0 on the rising edge of the 9th 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 is enabled). This flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

HOA flag (Host Address Detection Flag)

[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 is enabled). This flag is set to 1 on the rising edge of the 9th SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA bit 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 is enabled). This flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

31.2.10 I2C Bus Status Register 2 (ICSR2)

Address(es): IIC0.ICSR2 4005 3009h, IIC1.ICSR2 4005 3109h, IIC2.ICSR2 4005 3209h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TMOF	Timeout Detection Flag	0: Timeout not detected 1: Timeout detected.	R/(W)*1
b1	AL	Arbitration-Lost Flag	0: Arbitration not lost 1: Arbitration lost.	R/(W)*1
b2	START	Start Condition Detection Flag	0: Start condition not detected 1: Start condition detected.	R/(W)*1
b3	STOP	Stop Condition Detection Flag	0: Stop condition not detected 1: Stop condition detected.	R/(W)*1
b4	NACKF	NACK Detection Flag	0: NACK not detected 1: NACK detected.	R/(W)*1
b5	RDRF	Receive Data Full Flag	0: ICDRR contains no receive data 1: ICDRR contains receive data.	R/(W)*1
b6	TEND	Transmit End Flag	0: Data being transmitted 1: Data transmit complete.	R/(W)*1
b7	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 after the SCLn line state remains unchanged for the set period.

[Setting condition]

- When the SCLn line state remains unchanged for the period specified in the TMOH, TMOL, and TMOS bits in ICMR2 while the ICFER.TMOE bit is 1 (the timeout function is enabled) in master mode, or in slave mode, and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

AL flag (Arbitration-Lost Flag)

The AL flag indicates that bus mastership is lost in arbitration because of a bus conflict or some other reason when a start condition is issued or an address and data are transmitted. The IIC monitors the level on the SDAn line during transmission and, if the level on the line does not match the value of the bit being output, sets 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 in master mode or during data transmission in slave mode.

[Setting conditions]

When master arbitration-lost detection is enabled (ICFER.MALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of 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 issue requested) or the internal

SDA output state does not match the SDAn line level

- When the ST bit in ICCR2 is set to 1 (start condition issue 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 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 SDAn line level on the rising edge of 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 initiate an IIC reset or an internal reset.

Table 31.4 Relationship between arbitration-lost generation sources and arbitration-lost enable functions

ICFER			ICSR2		
MALE	NALE	SALE	AL	Error	Arbitration-lost generation source
1	×	×	1	Start condition issuance error	When internal SDA output state does not match SDAn line level when a start condition is detected while the ST bit in ICCR2 is 1
					When ST in ICCR2 is set to 1 with BBSY in ICCR2 set to 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
×	1	×	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master or slave receive mode
×	×	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

×: Don't care

START flag (Start Condition Detection Flag)

[Setting condition]

- When a start condition (or a restart condition) is detected.

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

STOP flag (Stop Condition Detection Flag)

[Setting condition]

- When a stop condition is detected.

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

NACKF flag (NACK Detection Flag)

[Setting condition]

- When an acknowledge is not received (NACK is 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 bit after reading NACKF = 1
- When 1 is written to the IICRST bit in ICCR1 to initiate 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)

[Setting conditions]

- When receive data is transferred from ICDRS to ICDRR. The RDRF flag is set to 1 on the rising edge of the 8th or 9th SCL clock cycle (selected by 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 bit after reading RDRF = 1
- When data is read from ICDRR
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

TEND flag (Transmit End Flag)

[Setting condition]

- On the rising edge of the 9th SCL clock cycle while the TDRE flag is 1.

[Clearing conditions]

- When 0 is written to the TEND bit 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 initiate an IIC reset or an internal reset.

TDRE flag (Transmit Data Empty Flag)

[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 initiate 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 is not set to 1.

31.2.11 I2C Bus Wakeup Unit Register (ICWUR)

Address(es): IIC0.ICWUR 4005 3016h

	b7	b6	b5	b4	b3	b2	b1	b0
	WUE	WUIE	WUF	WUACK	—	—	—	WUAFA
Value after reset:	0	0	0	1	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	WUAFA	Wakeup Analog Filter Additional Selection	0: Do not add the wakeup analog filter 1: Add the wakeup analog filter.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	WUACK	ACK bit for Wakeup Mode	Choice of four response modes in combination with ICCR1.IICRST and WUACK. See Table 31.5.	R/W
b5	WUF	Wakeup Event Occurrence Flag	0: Slave address not matching during wakeup 1: Slave address matching during wakeup.	R/W
b6	WUIE	Wakeup Interrupt Request Enable	0: Wakeup Interrupt Request (IIC0_WUI) disabled 1: Wakeup Interrupt Request (IIC0_WUI) enabled.	R/W
b7	WUE	Wakeup Function Enable	0: Wakeup function disabled 1: Wakeup function enabled.	R/W

Table 31.5 Wakeup mode

IICRST	WUACK	Operation mode	Description
0	0	Normal wakeup mode 1	ACK response at 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 return mode	ACK response on 9th SCL and no SCL low hold
1	1	EEP response mode	NACK response on 9th SCL and no SCL low hold

WUF flag (Wakeup Event Occurrence Flag)

[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 bit after reading WUF = 1
- ICE = 0, IICRST = 1.

31.2.12 Reserved (ICWUR2)

Address(es): IIC0.ICWUR2 4005 3017h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	WUSYF	WUASYF	WUSEN
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b0	WUSEN	Reserved	This bit is read as 1. Writing to this bit has no effect.	R/W
b1	WUASYF	Reserved	This bit is read as 1	R
b2	WUSYF	Reserved	This bit is read as 1	R
b7 to b3	—	Reserved	These bits are read as 1	R

31.2.13 Slave Address Register Ly (SARLy) (y = 0 to 2)

Address(es): IIC0.SARL0 4005 300Ah, IIC1.SARL0 4005 310Ah, IIC2.SARL0 4005 320Ah, IIC0.SARL1 4005 300Ch, IIC1.SARL1 4005 310Ch, IIC2.SARL1 4005 320Ch, IIC0.SARL2 4005 300Eh, IIC1.SARL2 4005 310Eh, IIC2.SARL2 4005 320Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	SVA[6:0]							SVA0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SVA0	10-Bit Address LSB	Slave address setting	R/W
b7 to b1	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), this 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.

When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, this bit is valid. When the SARUy.FS bit or SARyE bit is 0, the setting of 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), these bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS = 1), these bits combined with the SVA0 bit to form the lower 8 bits of a 10-bit address. While the SARyE bit in ICSER is 0, the setting of these bits is ignored.

31.2.14 Slave Address Register Uy (SARUy) (y = 0 to 2)

Address(es): IIC0.SARU0 4005 300Bh, IIC1.SARU0 4005 310Bh, IIC2.SARU0 4005 320Bh, IIC0.SARU1 4005 300Dh, IIC1.SARU1 4005 310Dh, IIC2.SARU1 4005 320Dh, IIC0.SARU2 4005 300Fh, IIC1.SARU2 4005 310Fh, IIC2.SARU2 4005 320Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SVA[1:0]		FS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	FS	7-Bit/10-Bit Address Format Select	0: The 7-bit address format is selected 1: The 10-bit address format is selected.	R/W
b2, b1	SVA[1:0]	10-Bit Address Upper Bits	Slave address setting	R/W
b7 to b3	—	Reserved	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 the 7-bit address or 10-bit address for the 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 settings of the SVA[1:0] bits and the SVA0 bit 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 settings of the SVA[1:0] bits and SARLy are valid.

When the SARyE bit in ICSER is 0 (SARLy and SARUy disabled), the setting in the SARUy.FS bit is invalid.

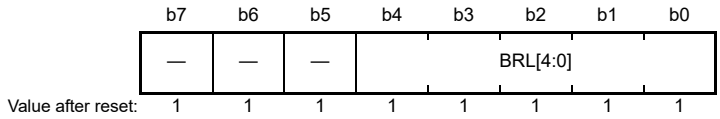
SVA[1:0] bits (10-Bit Address Upper Bits)

When the 10-bit address format is selected (FS = 1), these bits function as the upper 2 bits of a 10-bit address.

When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, these bits are valid. When the SARUy.FS bit or SARyE bit is 0, the setting of these bits is ignored.

31.2.15 I2C Bus Bit Rate Low-Level Register (ICBRL)

Address(es): IIC0.ICBRL 4005 3010h, IIC1.ICBRL 4005 3110h, IIC2.ICBRL 4005 3210h



Bit	Symbol	Bit name	Description	R/W
b4 to b0	BRL[4:0]	Bit Rate Low-Level Period	Low-level period of SCL clock	R/W
b7 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ICBRL is a 5-bit register that sets the low-level period of the SCL clock. ICBRL also works to generate the data setup time for automatic SCL low-hold operation (see section 31.9, Automatic Low-Hold Function for SCL). When the IIC is used only in slave mode, this register must be set to a value longer than the data setup time*1.

ICBRL counts the low-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 the ICBRL register to a value at least one greater than the number of stages in the noise filter. For this number, see the description of the ICMR3.NF[1:0] bits.

Note 1. Data setup time (tSU: DAT)
250 ns for up to 100 kbps: Standard-mode (Sm)
100 ns for up to 400 kbps: Fast-mode (Fm)

31.2.16 I2C Bus Bit Rate High-Level Register (ICBRH)

Address(es): IIC0.ICBRH 4005 3011h, IIC1.ICBRH 4005 3111h, IIC2.ICBRH 4005 3211h

Bit	Symbol	Bit name	Description	R/W
b4 to b0	BRH[4:0]	Bit Rate High-Level Period	High-level period of SCL clock	R/W
b7 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ICBRH is a 5-bit register that sets the high-level period of the SCL clock. ICBRH is valid in master mode. If the IIC is used only in slave mode, no setting is required in this register.

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 the ICBRH register to a value at least one greater than the number of stages in the noise filter. For this number, see the description of the ICMR3.NF[1:0] bits.

The IIC transfer rate and the SCL clock duty are calculated using the following expression:

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 rising time [tr] and SCLn line falling 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 filter stages selected in the ICMR3.NF bits.

Table 31.6 Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 0

Transfer rate (kbps)	CKS[2:0]	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	PCLKB[MHz]	NF[1:0]	Computation expression
100	011	15 (EFh)	18 (F2h)	32	-	1)
400	001	9 (E9h)	20 (F4h)	32	-	1)

Table 31.7 Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 1 and NFE = 0

Transfer rate (kbps)	CKS[2:0]	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	PCLKB[MHz]	NF[1:0]	Computation expression
100	011	14 (EEh)	17 (F1h)	32	-	4)
400	001	8 (E8h)	19 (F3h)	32	-	4)

Table 31.8 Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 1 and NFE = 1

Transfer rate (kbps)	CKS[2:0]	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	PCLKB[MHz]	NF[1:0]	Computation expression
100	011	12 (ECh)	15 (EFh)	32	01b	5)
400	001	6 (E6h)	17 (F1h)	32	01b	5)

Note: SCLn line rising time (tr): ≤ 100 kbps, Sm: 1000 ns, ≤ 400 kbps, Fm: 300 ns
SCLn line falling time (tf): ≤ 400 kbps, Sm/Fm: 300 ns

31.2.17 I2C Bus Transmit Data Register (ICDRT)

Address(es): IIC0.ICDRT 4005 3012h, IIC1.ICDRT 4005 3112h, IIC2.ICDRT 4005 3212h

	b7	b6	b5	b4	b3	b2	b1	b0
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 is 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 from and written to. Write transmit data to ICDRT once when a transmit data empty interrupt (IICn_TXI) request is generated.

31.2.18 I2C Bus Receive Data Register (ICDRR)

Address(es): IIC0.ICDRR 4005 3013h, IIC1.ICDRR 4005 3113h, IIC2.ICDRR 4005 3213h

	b7	b6	b5	b4	b3	b2	b1	b0
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 (IICn_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 one cycle before the RDRF flag is set to 1 again.

31.2.19 I2C bus Shift Register (ICDRS)

	b7	b6	b5	b4	b3	b2	b1	b0
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 sent from the SDAn pin. During reception, data is transferred from ICDRS to ICDRR after 1 byte of data is received. ICDRS cannot be accessed directly.

31.3 Operation

31.3.1 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 31.3 shows the I2C bus format; Figure 31.4 shows the I2C bus timing.

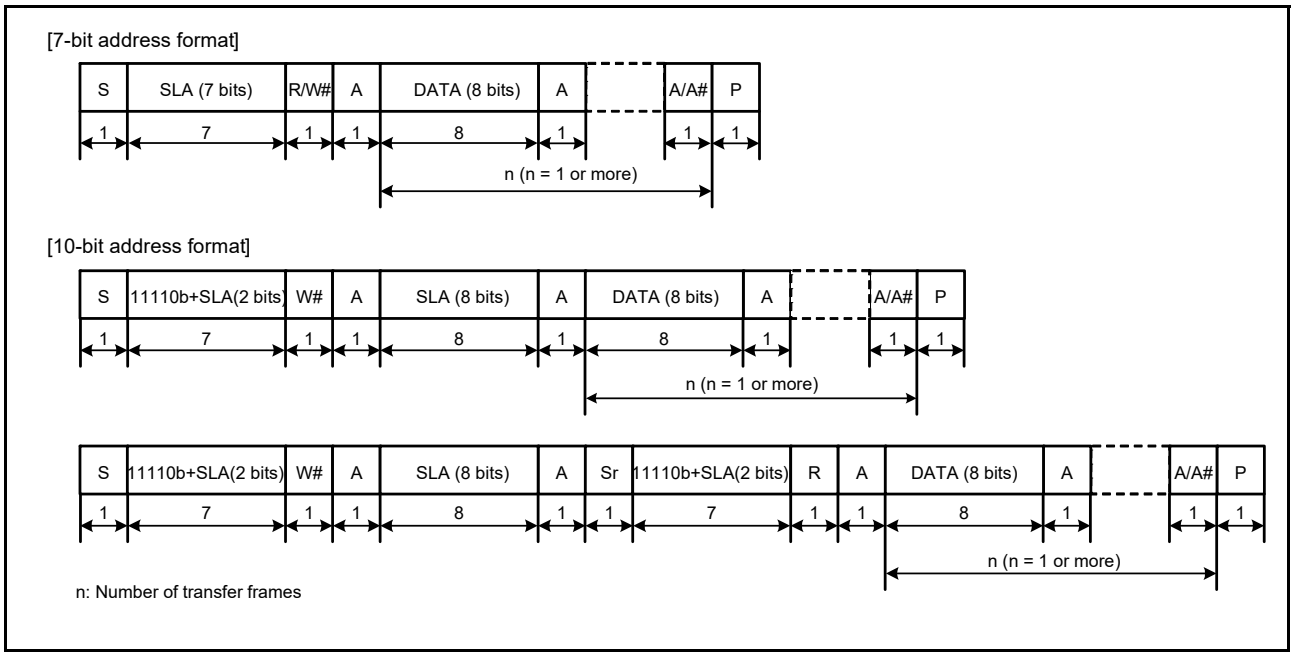


Figure 31.3 I2C bus format

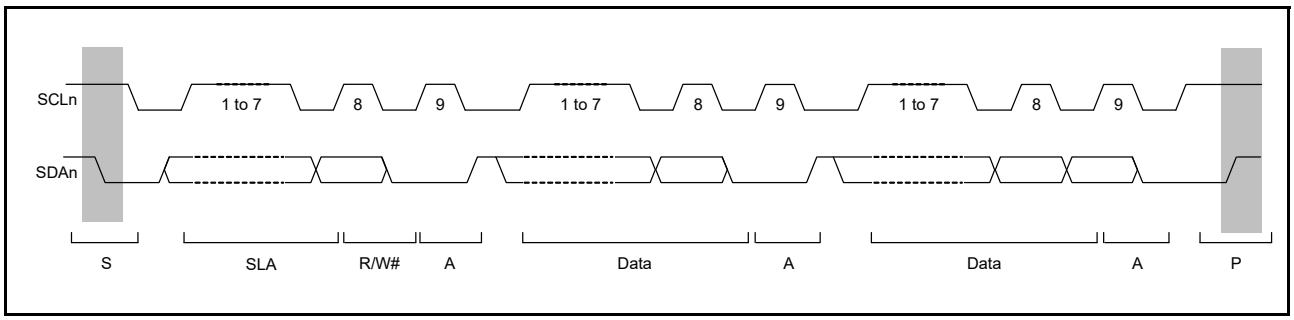


Figure 31.4 I2C bus timing (SLA = 7 bits)

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 high after the setup time elapses 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.

31.3.2 Initial Settings

Before starting data transmission or reception, initialize the IIC using the procedure shown in Figure 31.5.

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 necessary. For initial settings of the IIC, see Figure 31.5.
5. When the necessary register settings are complete, set the ICCR1.IICRST bit to 0 to release the IIC reset.

Note: This procedure is not necessary if the IIC initialization is already complete.

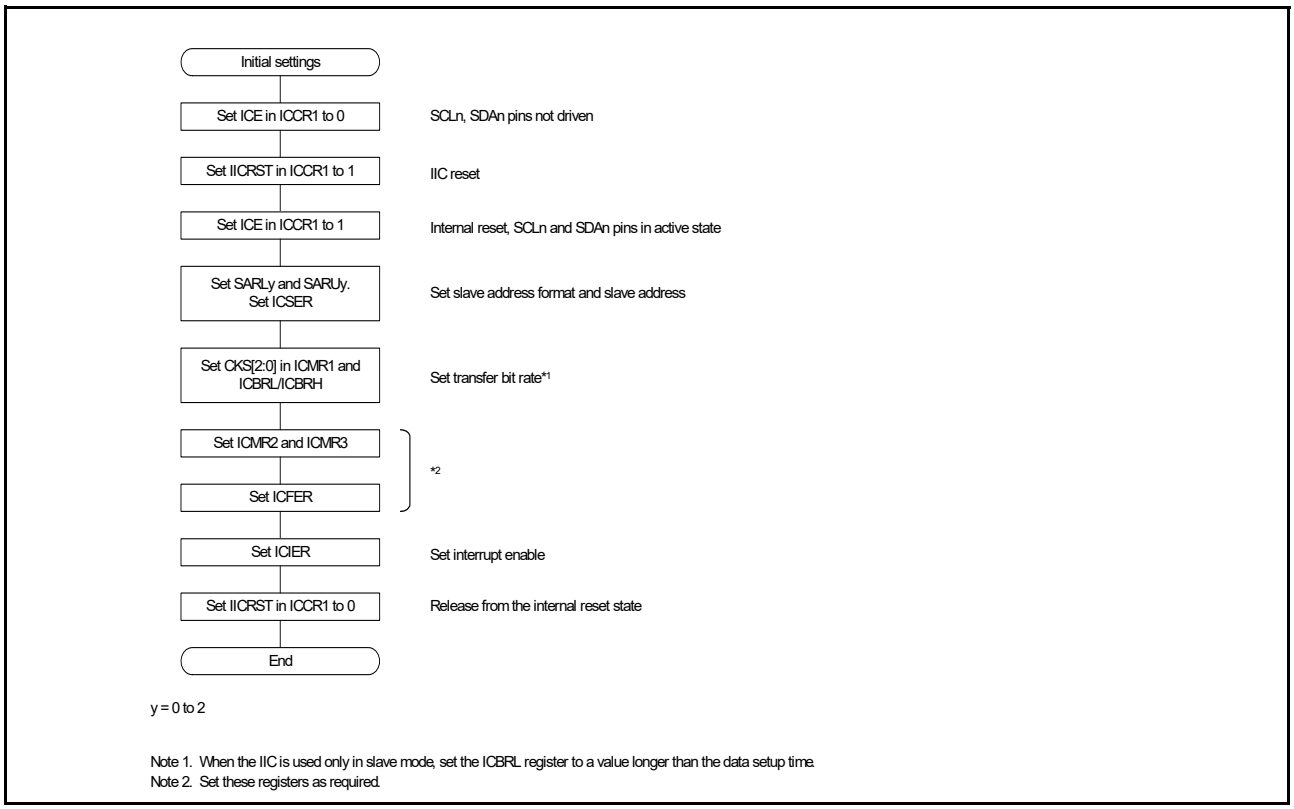


Figure 31.5 Example of IIC initialization flow

31.3.3 Master Transmit Operation

In master transmit operation, the IIC outputs the SCL clock and transmit data signals as the master device, and the slave device returns acknowledgments. Figure 31.6 shows an example of master transmission. Figure 31.7 to Figure 31.9 show the timing of operations in master transmission.

To set up and perform master transmission:

1. Initialize the IIC using the procedure described in section 31.3.2, Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is open, 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 flag and the START flag in ICSR2 automatically set to 1, and the ST bit is automatically 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 the start condition requested by the ST bit has successfully completed, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 in response to 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 is automatically set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again sets 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 according to 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.
   If the ICSR2.NACKF flag is 1, indicating 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.
   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 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 of 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. For details regarding issuing a stop condition, see section 31.11.3, 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. In addition, the IIC automatically sets the TDRE and TEND flags to 0, and sets the STOP flag in ICSR2 to 1.
7. After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

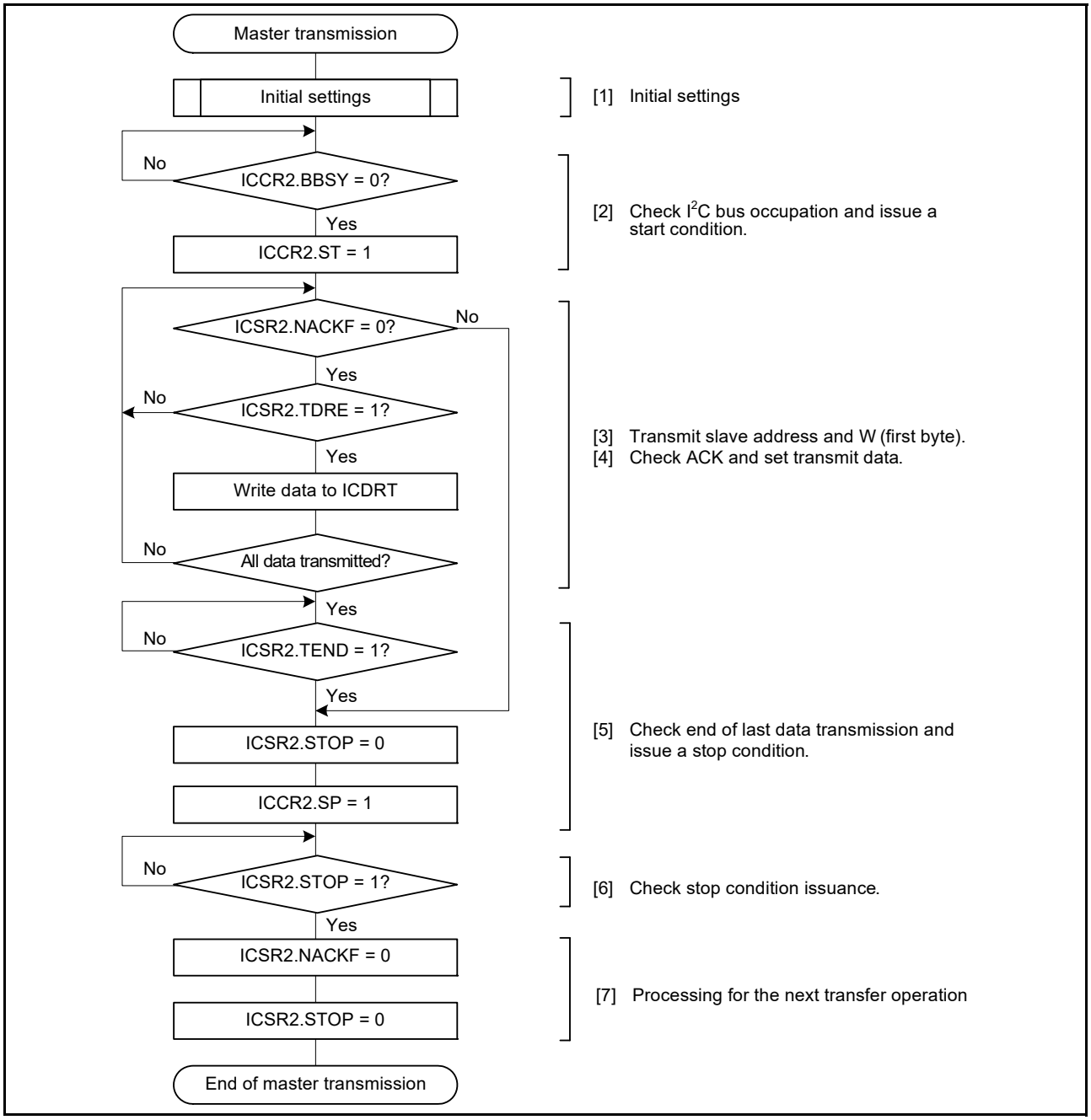


Figure 31.6 Example of master transmission flow

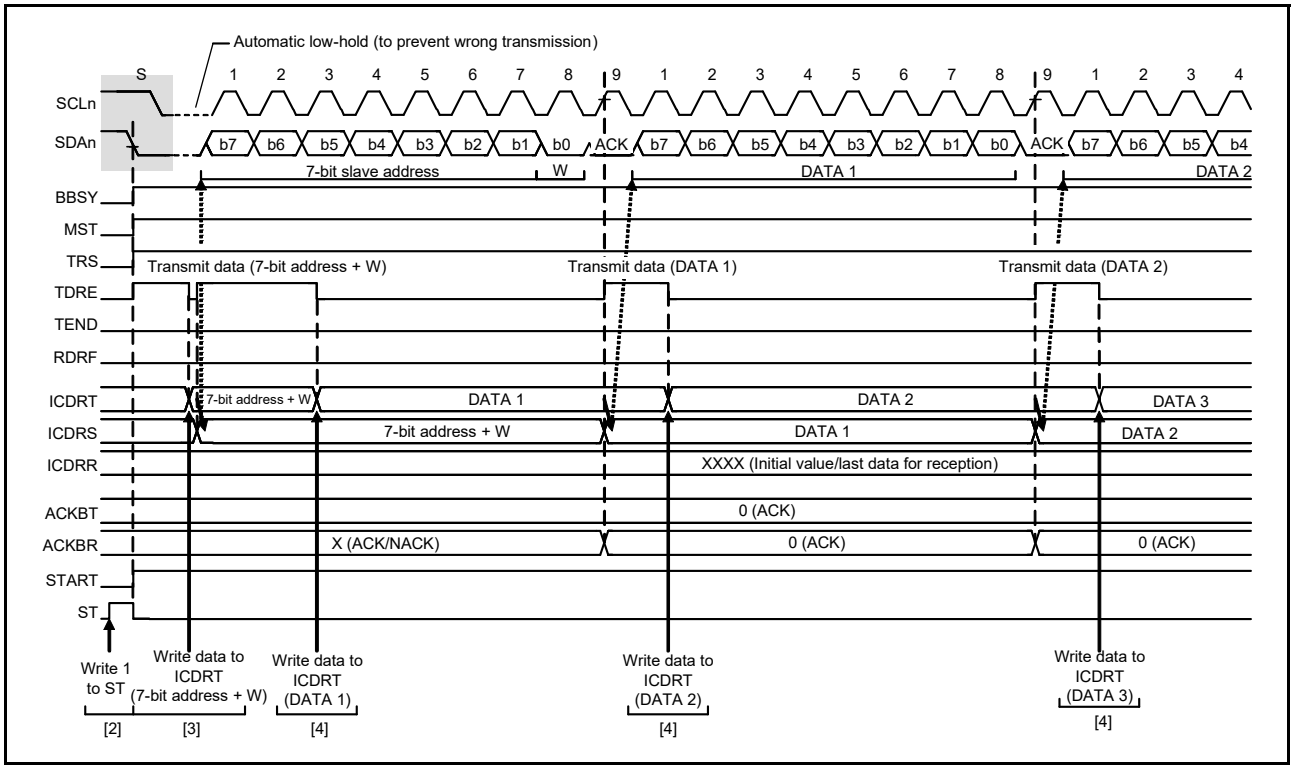


Figure 31.7 Master transmit operation timing (1) (7-bit address format)

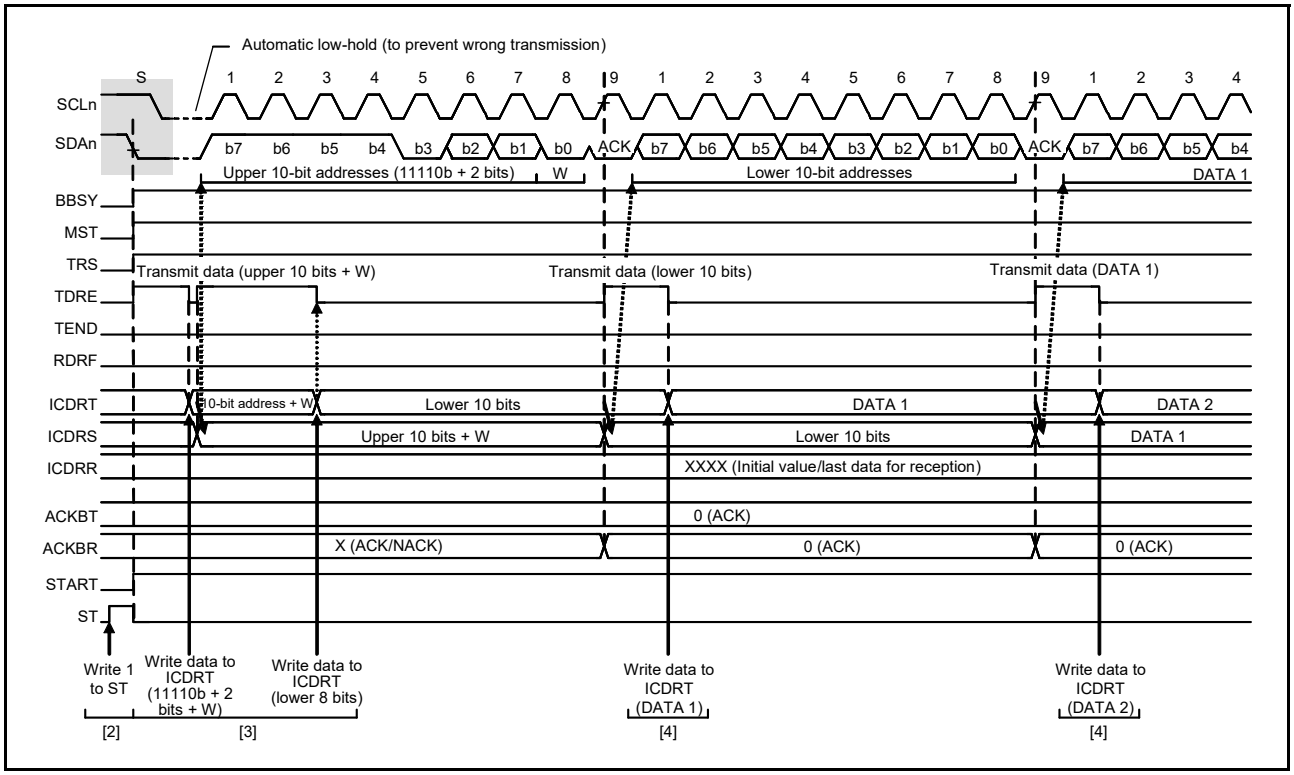


Figure 31.8 Master transmit operation timing (2) (10-bit address format)

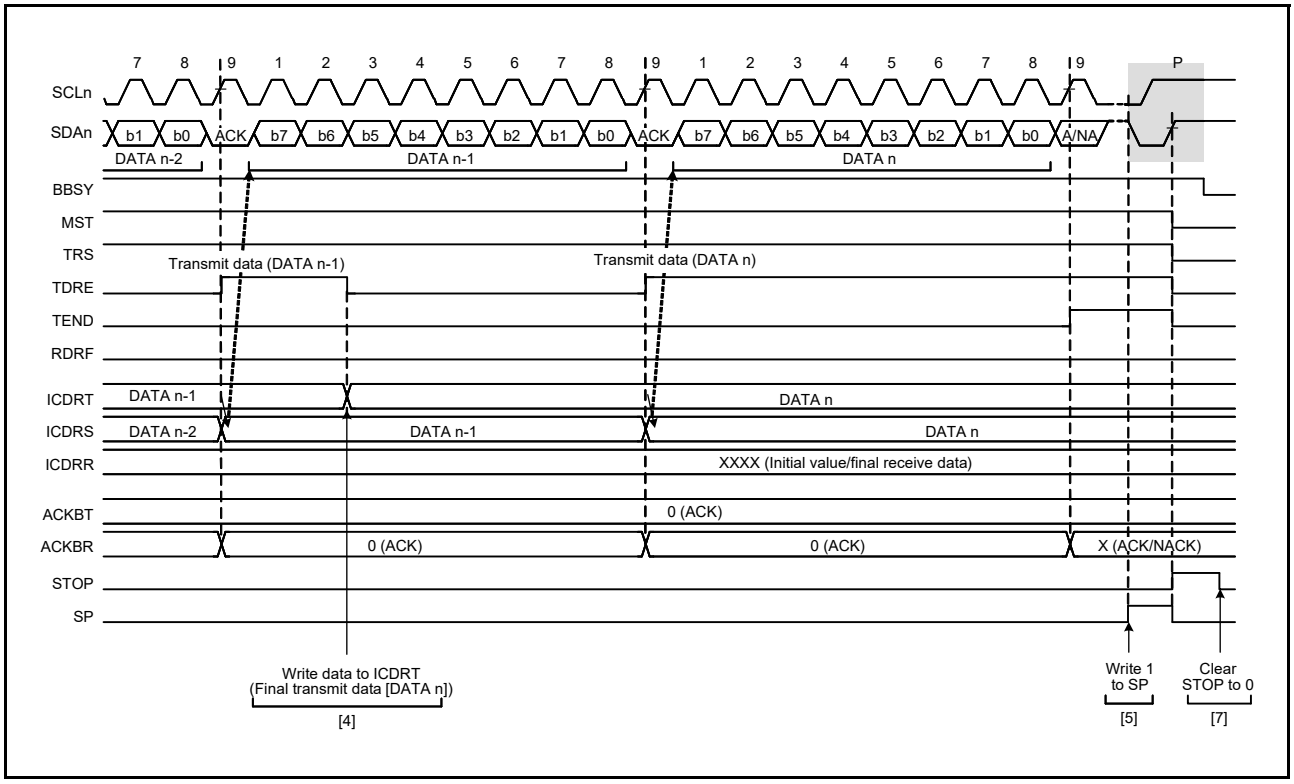


Figure 31.9 Master transmit operation timing (3)

31.3.4 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, the slave address phase of the procedure is performed in master transmit mode, and the subsequent steps are performed in master receive mode.

Figure 31.10 and Figure 31.11 show examples of master reception (7-bit address format). Figure 31.12 to Figure 31.14 show the timing of operations in master reception.

To set up and perform master reception:

1. If the IIC has not been initialized, initialize the IIC using the procedure in section 31.3.2, Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is open, and then set the ST bit in ICCR2 to 1 to request issue of a start condition. On receiving the request, the IIC issues a start condition. When the IIC detects the start condition, the BBSY flag and the START flag in ICSR2 automatically set to 1 and the ST bit is automatically 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 the issuance of the start condition as requested by the ST bit successfully completed, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 in response to 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 is automatically set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag is again 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 according to 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 9th cycle of SCL clock, placing the IIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag is automatically set to 1.
   If the ICSR2.NACKF flag is 1, indicating that no slave device recognized the address or that there is 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, transmit 1111 0b, the two upper bits of the slave address, and the R bit to place the IIC in master receive mode.

4. Dummy read ICDRR after confirming that the RDRF flag in ICSR2 is 1. Doing so causes the IIC to 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 SCL clock, as selected by the RDRFS bit in ICMR3. Reading the ICDRR register produces the received data, and automatically sets the RDRF flag to 0. The value of the acknowledgment field received during the 9th 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 next-to-last byte, set the ICMR3.WAIT bit to 1 for wait insertion before reading the ICDRR register, containing the second byte from the last. 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 low on the rising edge of the 9th clock cycle in reception of the last byte, which enables the issue of a stop condition.
6. When the ICMR3.RDRFS bit is 0 and the slave device must be notified that it should 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 (to request stop condition), and then read the last byte from the ICDRR register. When the ICDRR register is read, the IIC is released from the wait state and issues the stop condition after low-level output in the 9th 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 ICCR2.MST and ICCR2.TRS bits 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, and then set the ICSR2.NACKF and ICSR2.STOP flags to 0 for the next transfer operation.

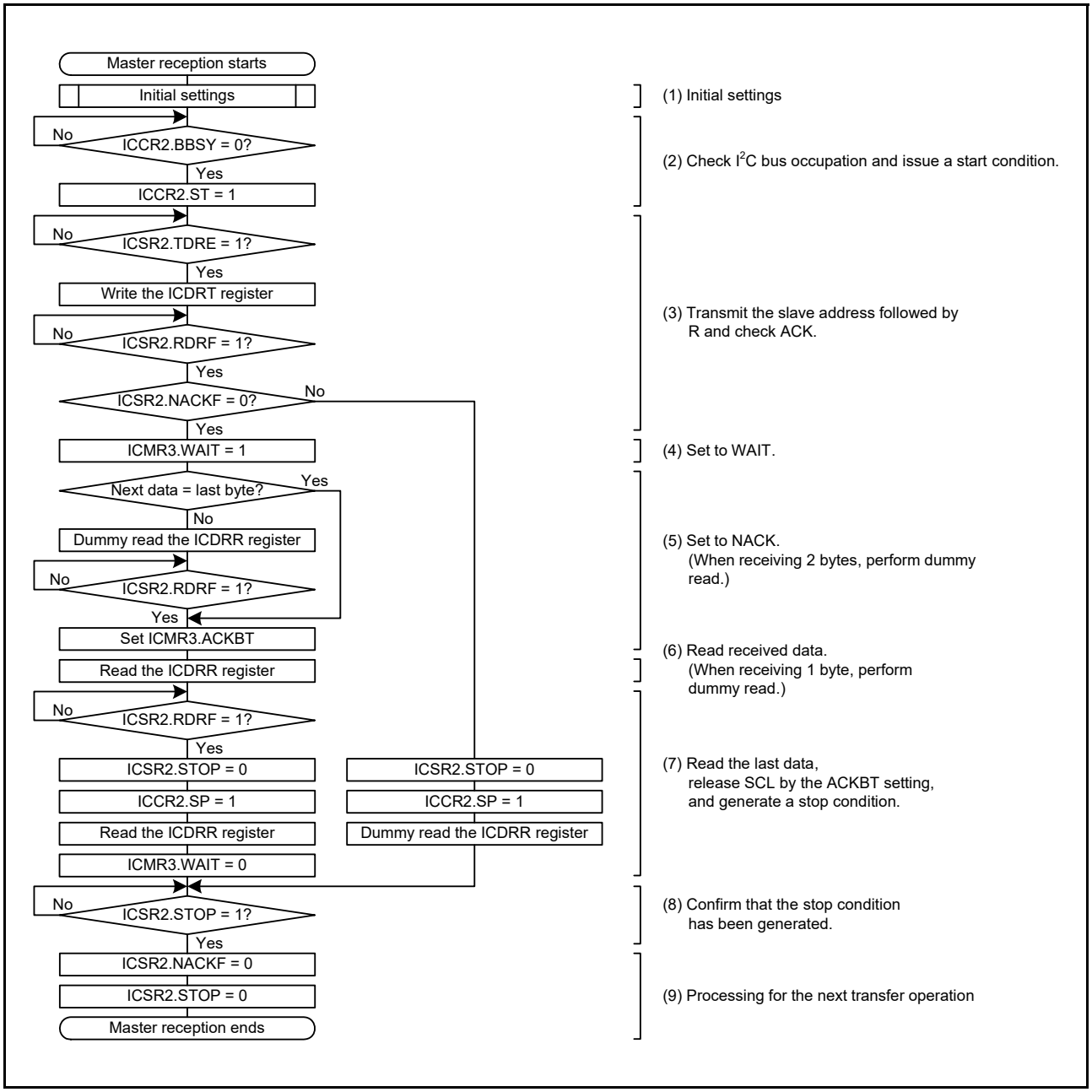


Figure 31.10 Example of master reception with 7-bit address format, with 1 or 2 bytes

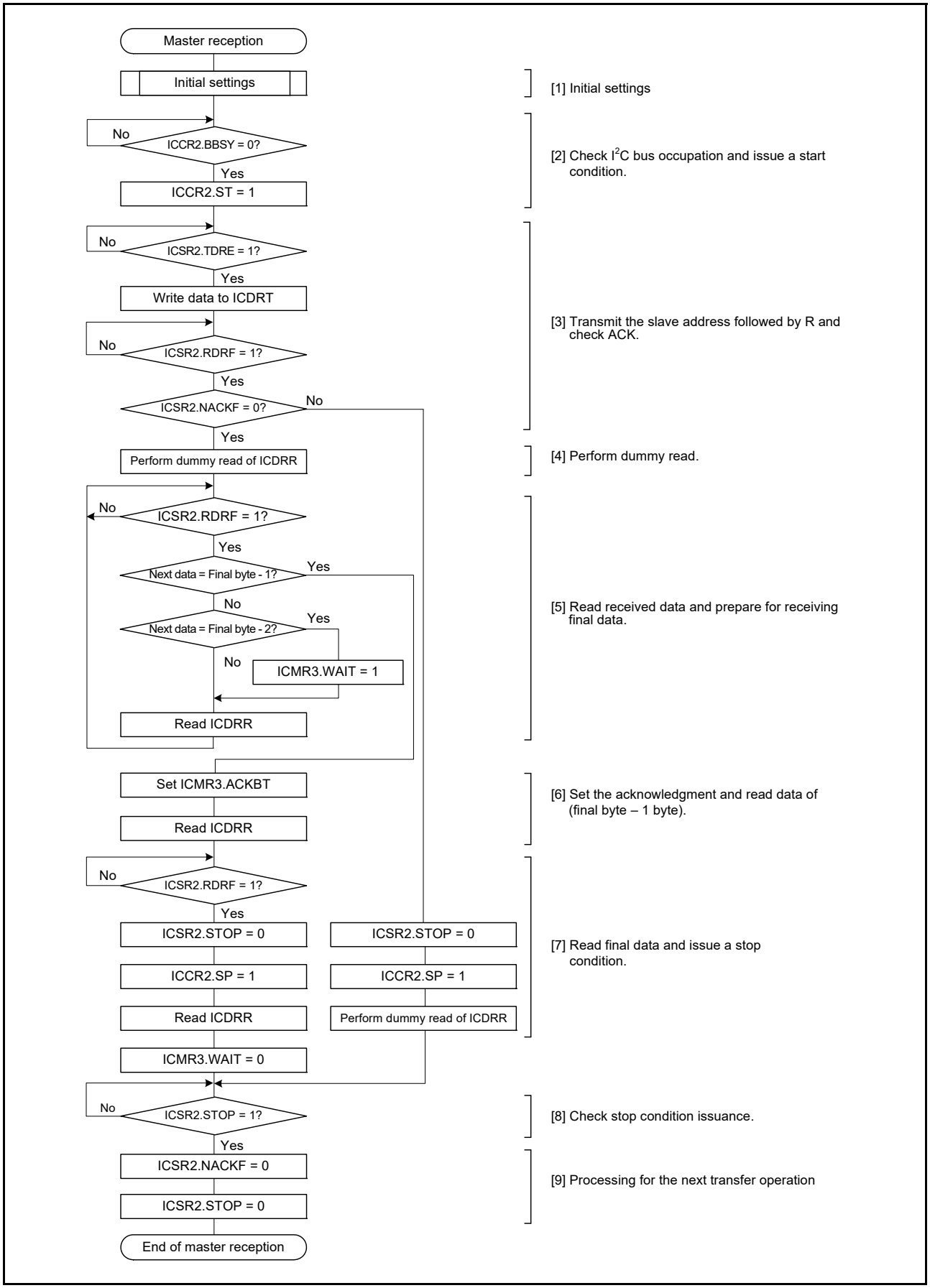


Figure 31.11 Example of master reception with 7-bit address format, with 3 bytes or more

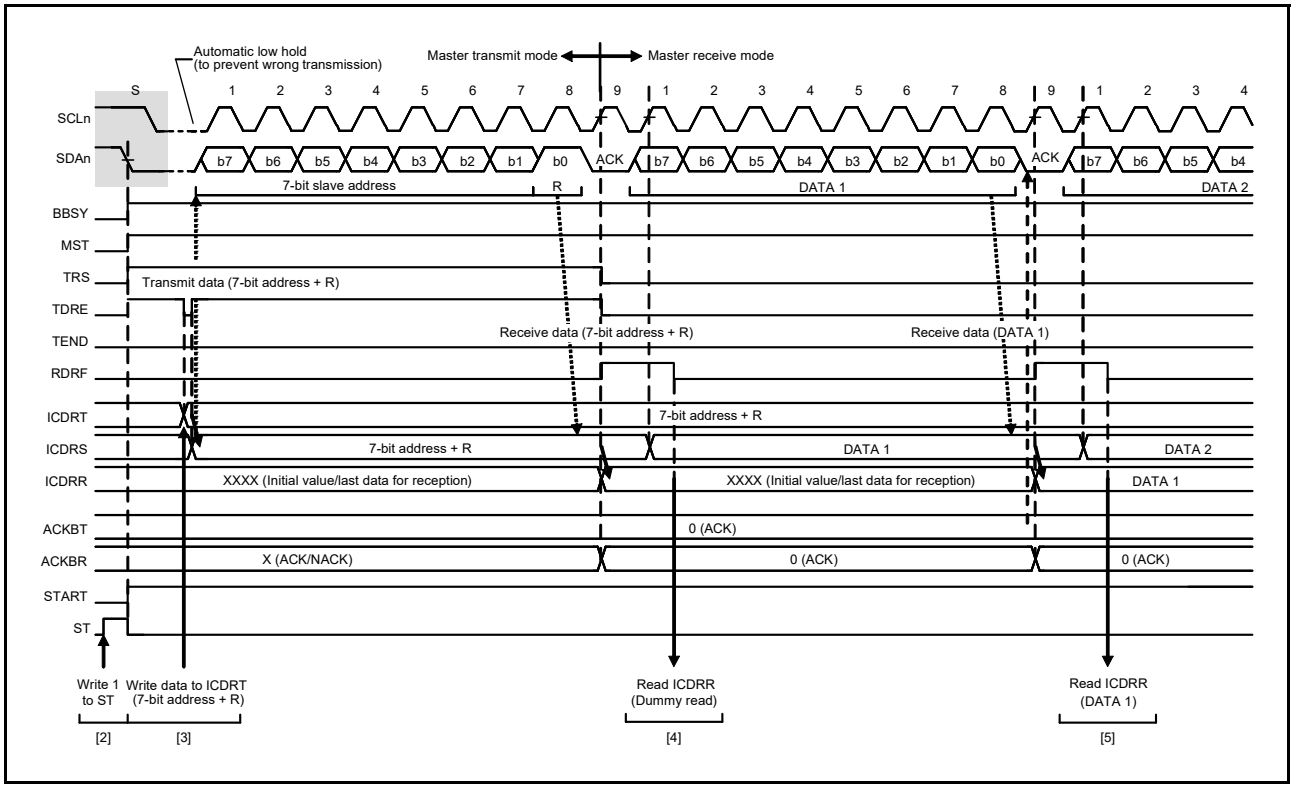


Figure 31.12 Master receive operation timing (1) with 7-bit address format, when RDRFS = 0

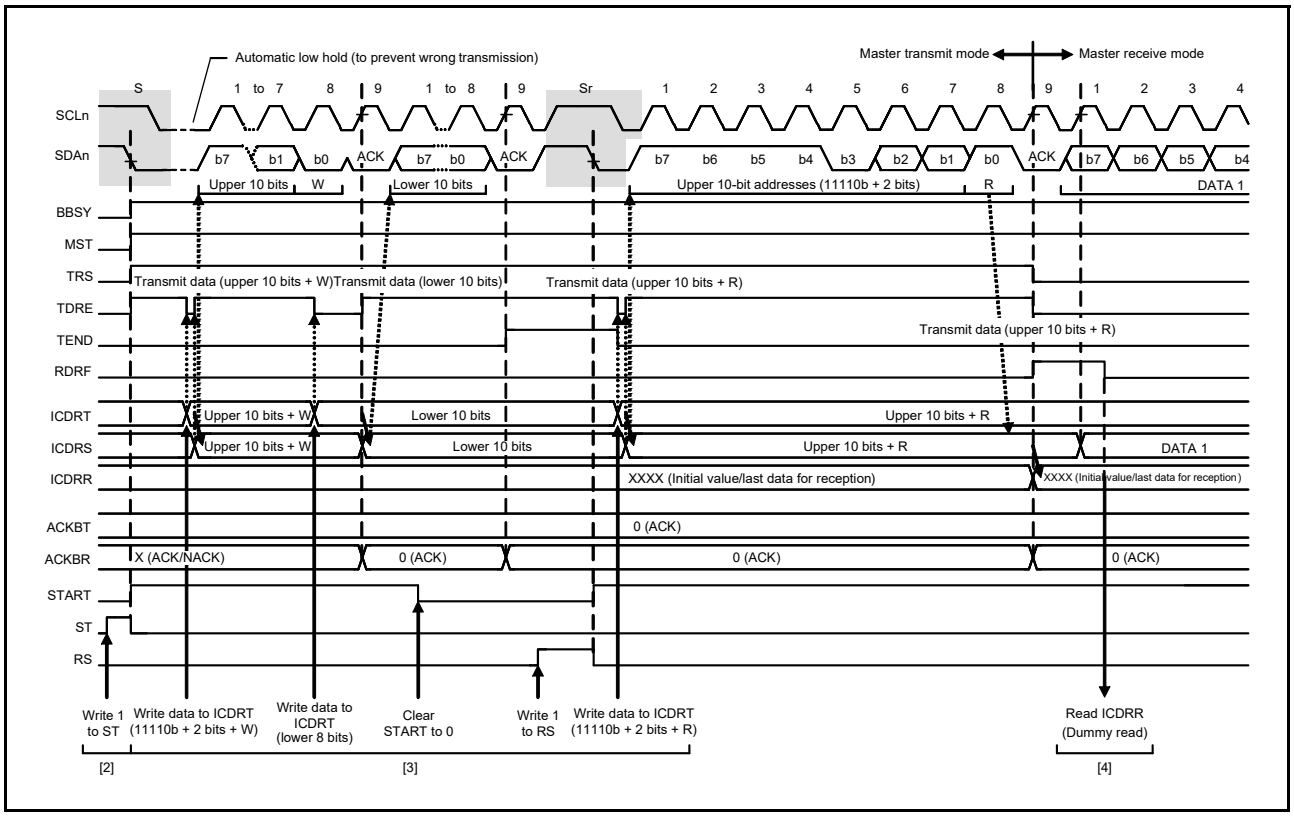


Figure 31.13 Master receive operation timing (2) with 10-bit address format, when RDRFS = 0

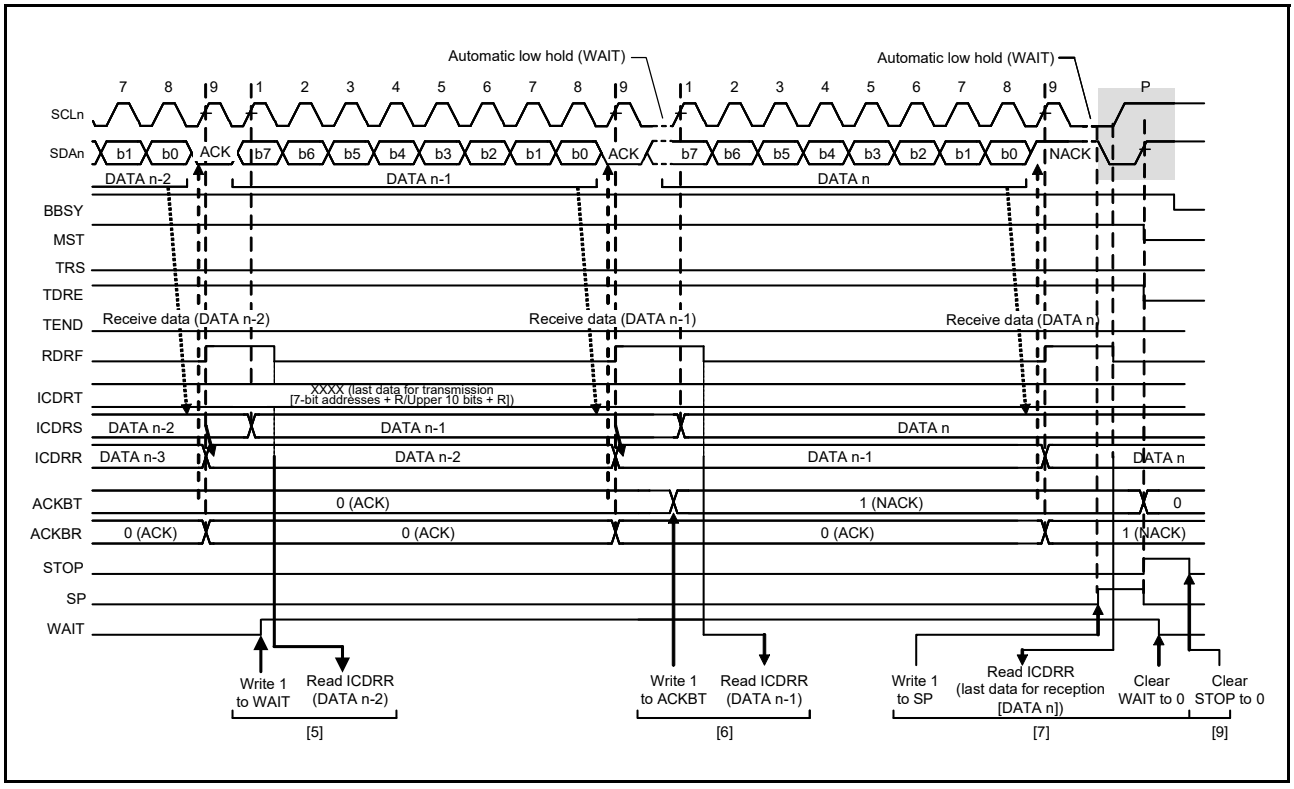


Figure 31.14 Master receive operation timing (3) when RDRFS = 0

31.3.5 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 31.15 shows an example of slave transmission. Figure 31.16 and Figure 31.17 show the operation timing in slave transmission.

To set up and perform slave transmission:

1. If the IIC has not been initialized, initialize the IIC using the procedure in section 31.3.2, Initial Settings. After initialization, the IIC stays in the standby state until it receives a slave address that it matches.
2. After receiving a matching slave address, the IIC sets one of the associated bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the 9th cycle of SCL clock and outputs the value set in the ICMR3.ACKBT bit as the acknowledge bit on the 9th cycle of SCL clock. If the value of the received 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.TEND flag is 1, and write the transmit data to the ICDRT register. At this time, 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 unit 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 9th falling edge of SCL clock.
5. When the ICSR2.NACKF flag or the 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 (y = 0 to 2) bits, 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, and set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

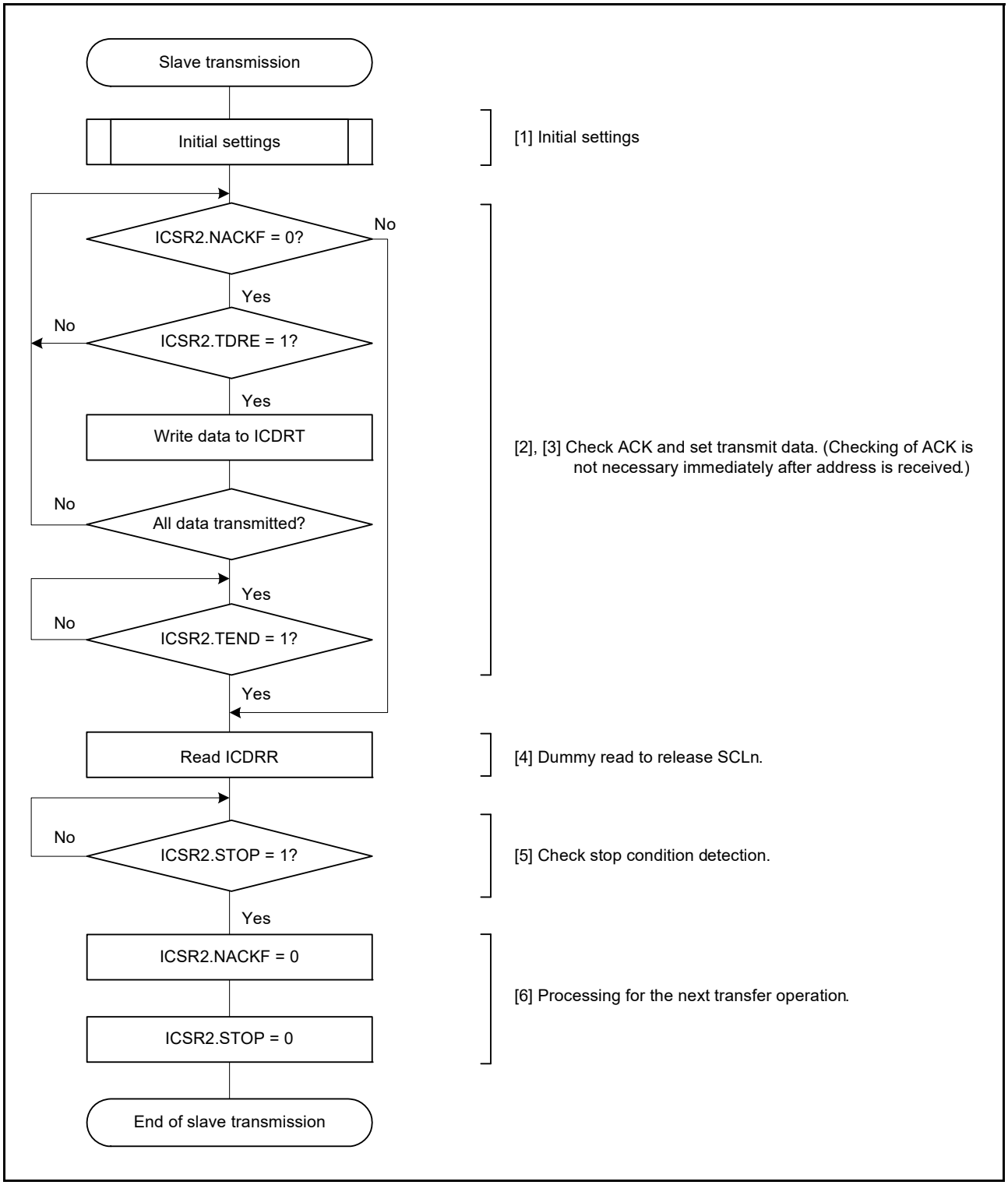


Figure 31.15 Example slave transmission flow

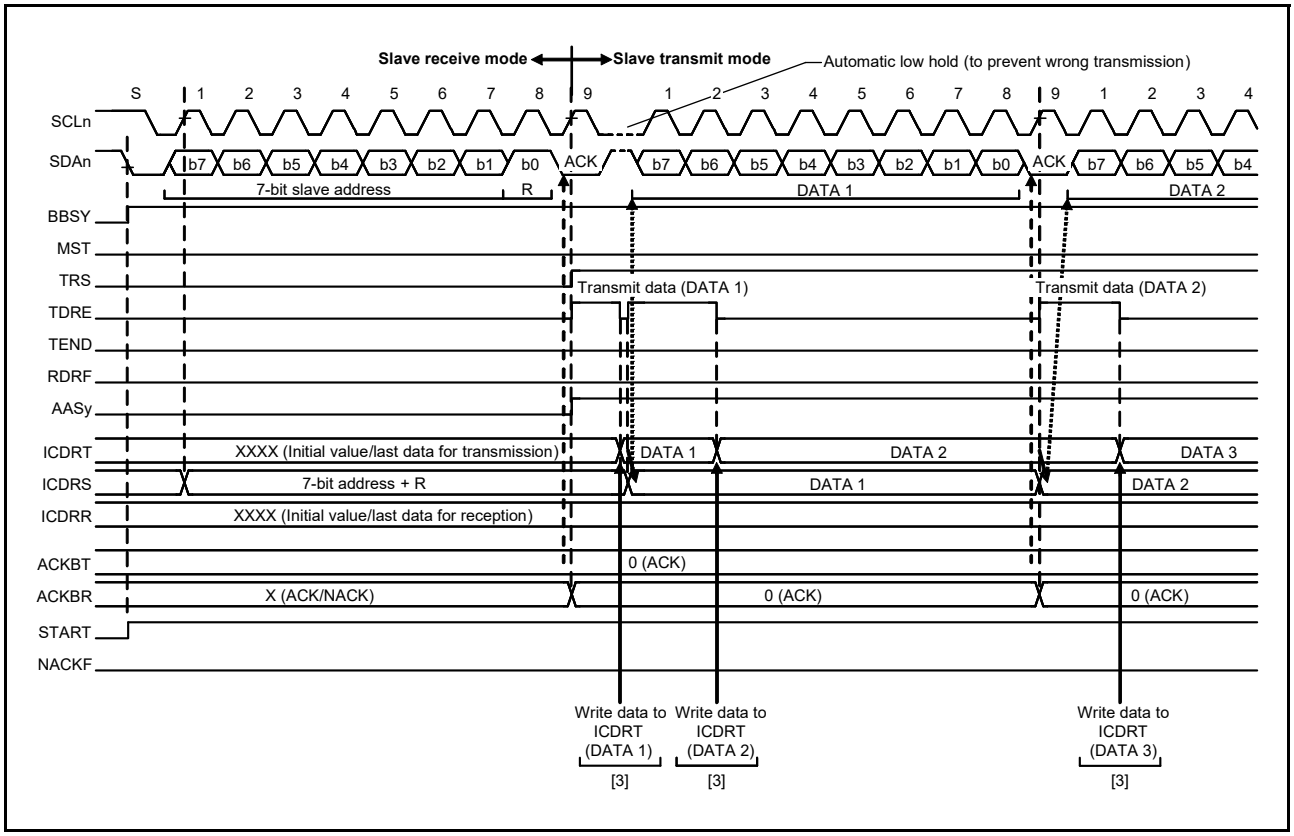


Figure 31.16 Slave transmit operation timing (1) with 7-bit address format

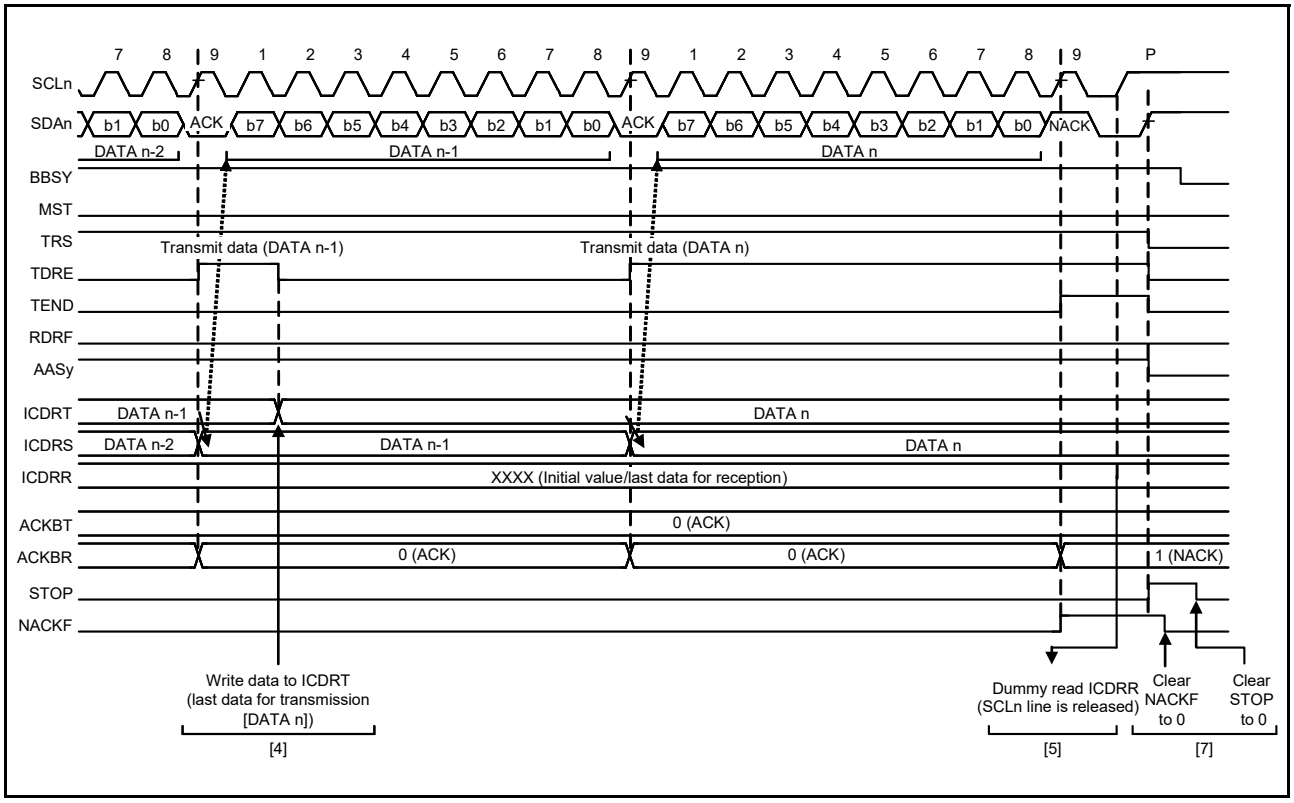


Figure 31.17 Slave transmit operation timing (2)

31.3.6 Slave Receive Operation

In a slave receive operation, the master device outputs the SCL clock and transmit data, and the IIC returns acknowledgments as a slave device.

Figure 31.18 shows an example of slave reception. Figure 31.19 and Figure 31.20 show the timing of operations in slave reception.

To set up and perform slave reception:

1. If the IIC has not been initialized, initialize the IIC using the procedure in section 31.3.2, Initial Settings. After initialization, the IIC stays in the standby state until it receives a slave address that matches an address enabled in the ICSER register.
2. After receiving a matching slave address, the IIC sets one of the associated ICSR1.HOA, GCA, and AASy (y = 0 to 2) bits to 1 on the rising edge of the 9th cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the 9th cycle of the SCL clock. If the value of the received 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, and then dummy read the ICDRR register. 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 the next byte is received while the RDRF flag is still set to 1, the IIC holds the SCLn line low until one SCL cycle before the point where RDRF must be set. In this case, reading the ICDRR register releases the SCLn line from being held low.
   When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1, read the ICDRR register until all the data is completely received.
5. On detecting the stop condition, the IIC automatically clears the ICSR1.HOA, GCA, and AASy (y = 0 to 2) bits to 0.
6. Check that the ICSR2.STOP flag is 1, and then set the ICSR2.STOP flag to 0 for the next transfer operation.

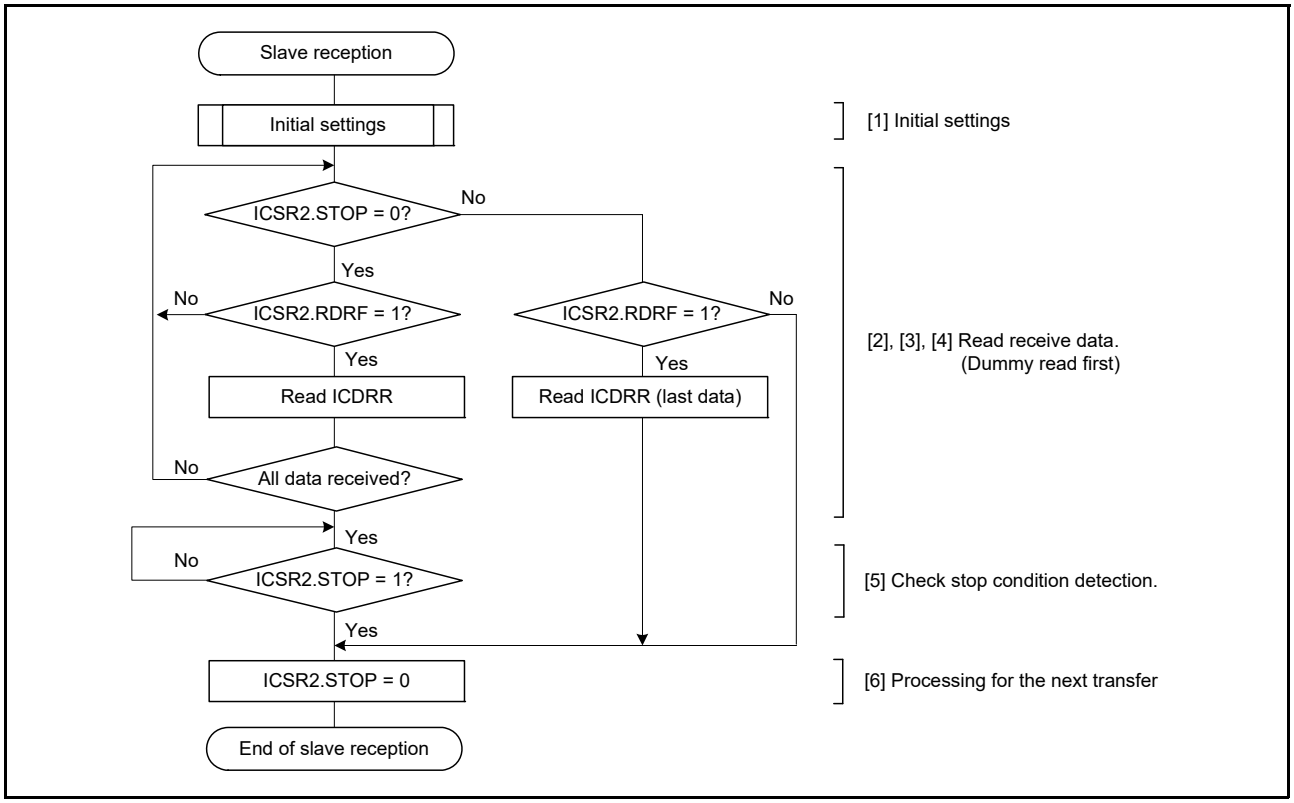


Figure 31.18 Example slave reception flow

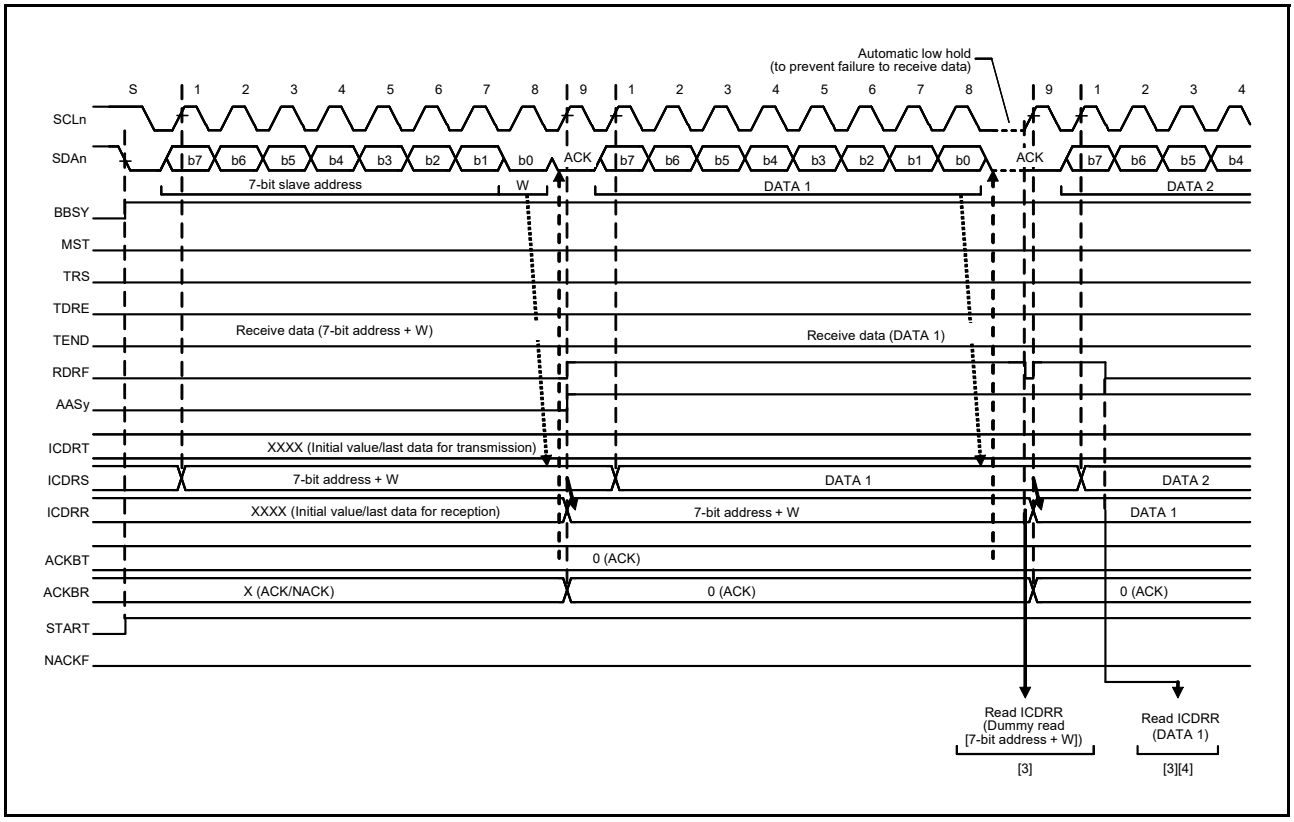


Figure 31.19 Slave receive operation timing (1) with 7-bit address format, when RDRFS = 0

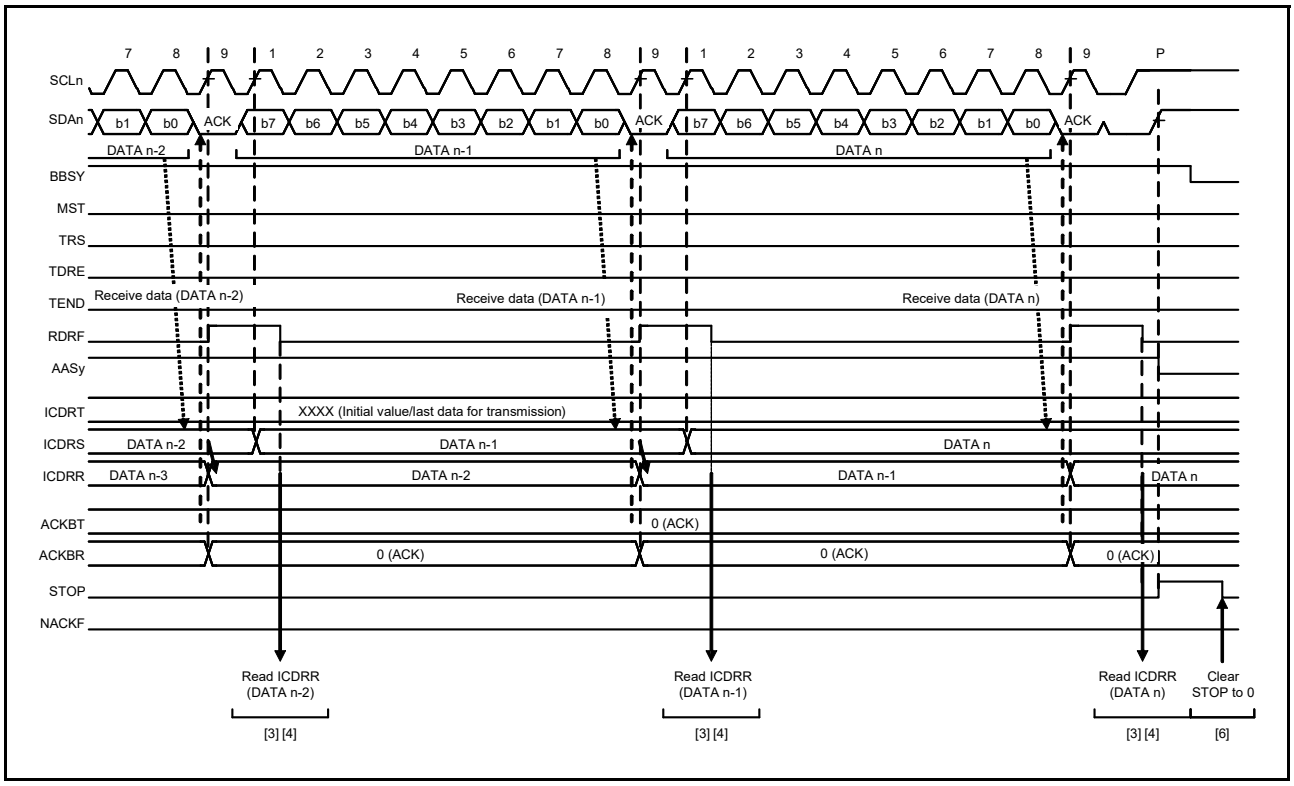


Figure 31.20 Slave receive operation timing (2) when RDRFS = 0

31.4 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 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 (releases the line) when it completes counting. The IIC repeats this process to generate the SCL clock.

If multiple master devices are connected to the I2C 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, and the level on the SCLn line falls because an SCL signal is being generated by another master device, the IIC stops counting when it detects the falling edge, drives the level on the SCLn line low, and starts counting the low-level period specified in ICBRL. When the IIC finishes counting the low-level period, it stops driving the SCLn line to 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 was 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.

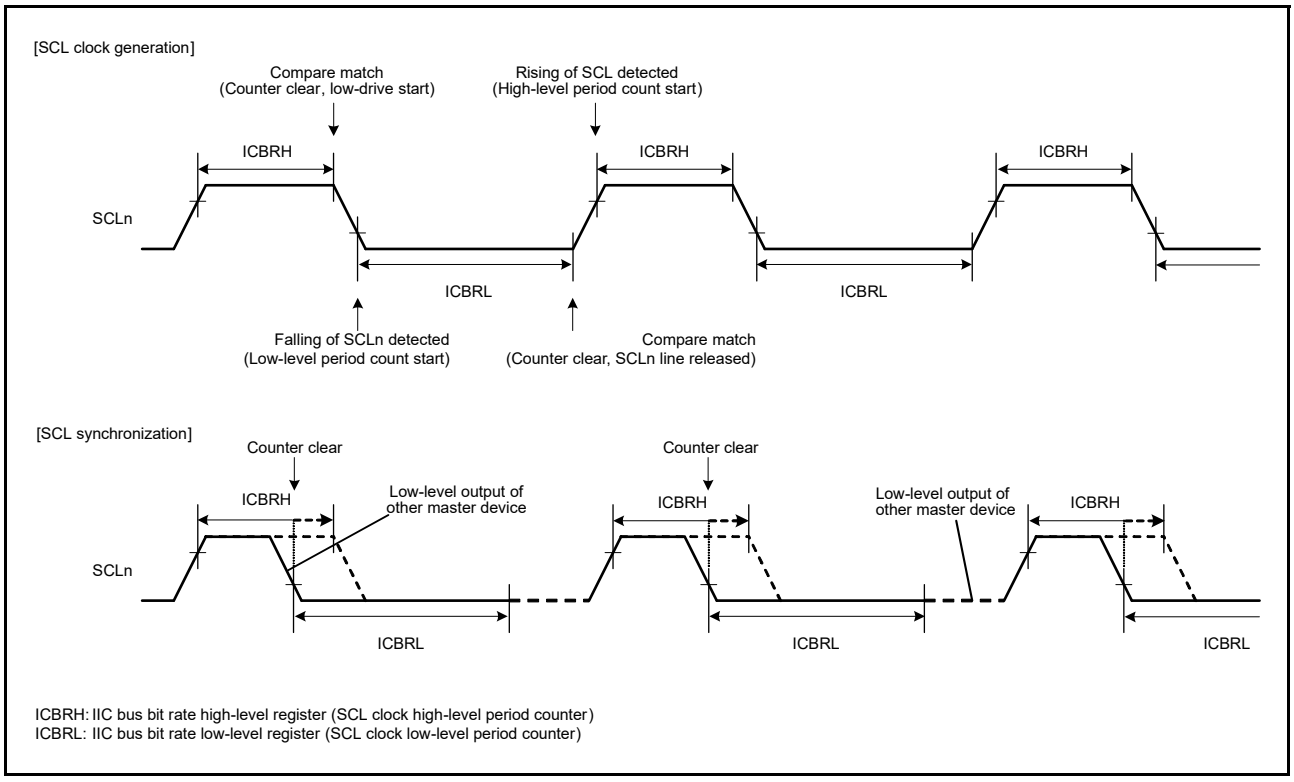


Figure 31.21 Generation and synchronization of SCL signal from IIC

31.5 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 to 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 any value other than 000b, and disabled by setting the same bits to 000b.

When the SDA output delay function is enabled, for example, while the SDDL[2:0] bits in ICMR2 are set to any value other than 000b, the DLCS bit in ICMR2 selects the clock source for counting by the SDA output delay counter, either as the internal base clock (IICϕ) for the IIC module or as the internal base clock divided by two (IICϕ/2). The counter counts the number of cycles set in the SDDL[2:0] bits in ICMR2. When the delay 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.

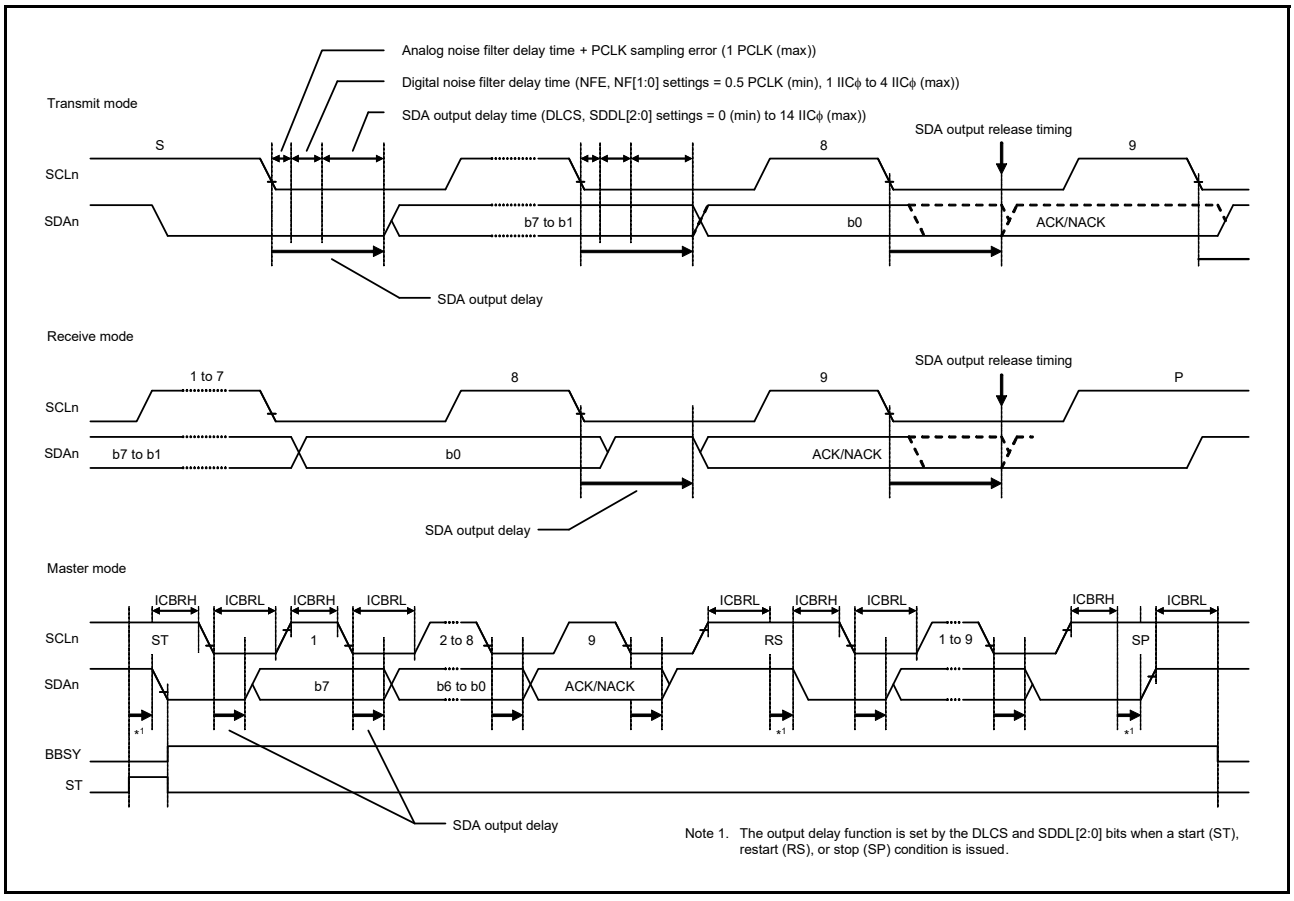


Figure 31.22 SDA output delay function

31.6 Digital Noise Filter Circuits

The states of the SCLn and SDAn pins are conveyed to the internal circuitry through analog noise-filter and digital noise-filter circuits. Figure 31.23 is 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 effective stages in the digital noise filter is selected in the NF[1:0] bits in ICMR3. The selected number of effective stages determines the noise-filtering capability as a period from one to four 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 effective flip-flop circuit stages as selected in the NF[1:0] bits in ICMR3, the signal level is conveyed to 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 instance, if data transfer at 400 kbps with PCLKB = 4 MHz, the characteristics of the digital noise filter might lead to the elimination of the required signals as noise. In such a case, 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.

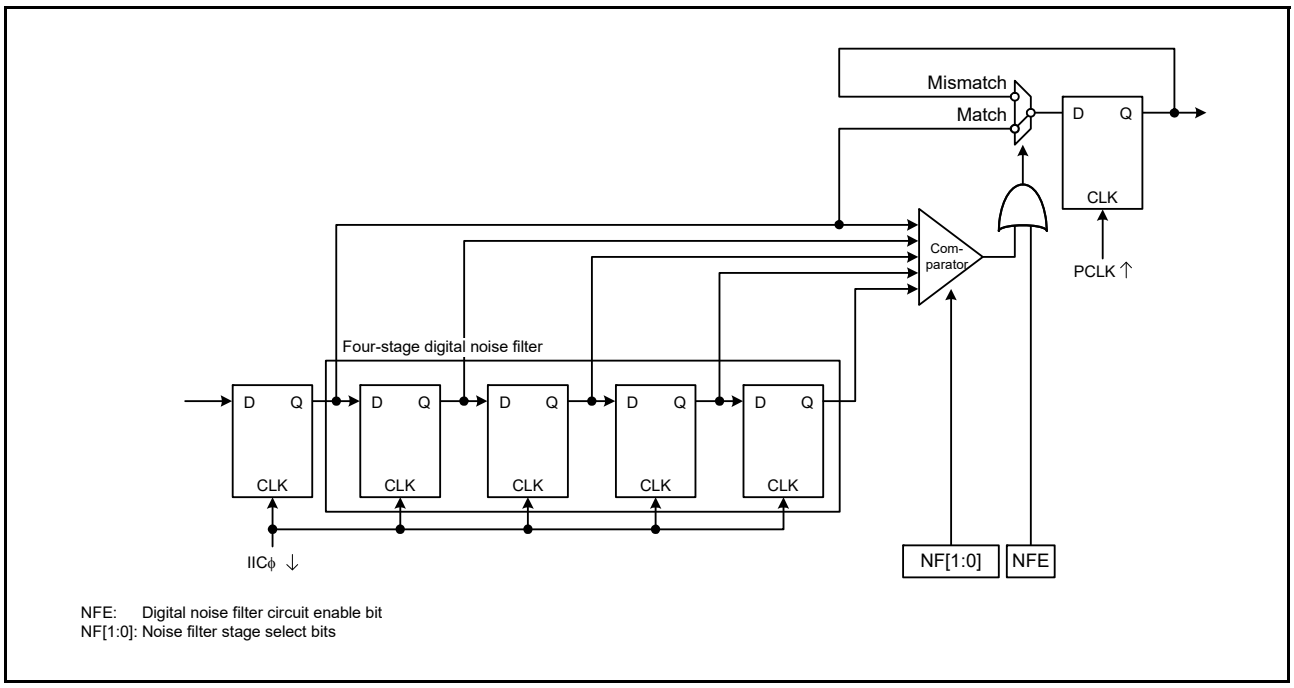


Figure 31.23 Digital noise filter circuit block diagram

31.7 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.

31.7.1 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 9th 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 31.24 to Figure 31.26 show the AASy flag set timing in three cases.

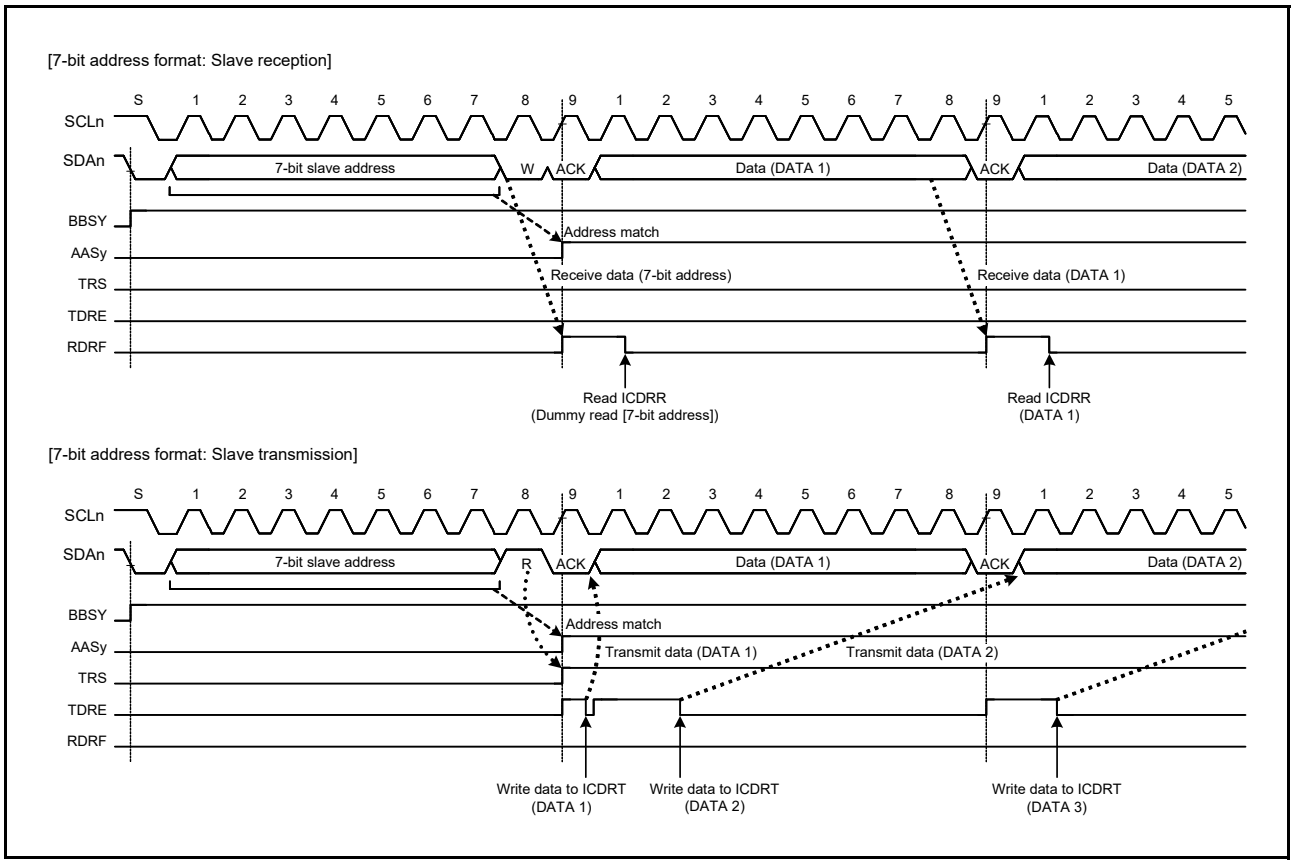


Figure 31.24 AASy flag set timing with 7-bit address format selected

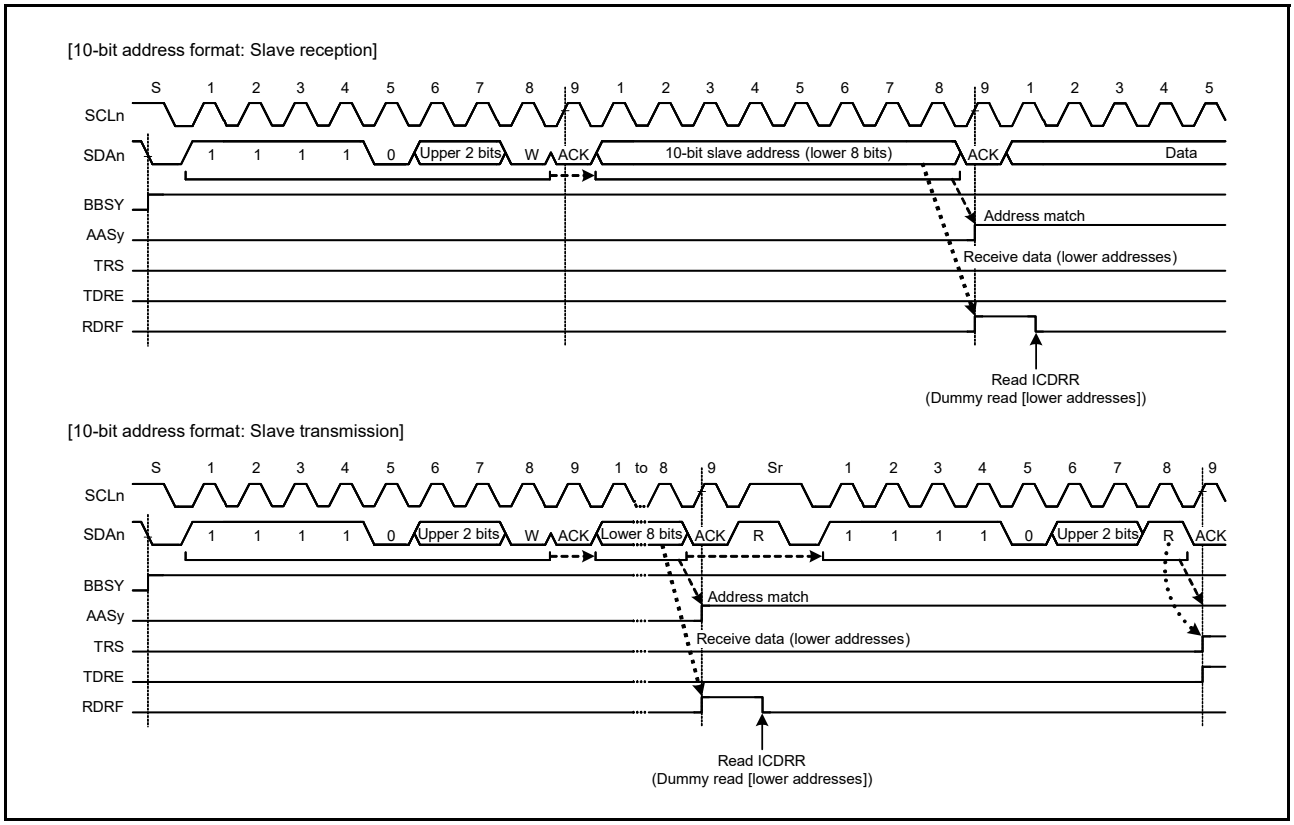


Figure 31.25 AASy flag set timing with 10-bit address format selected

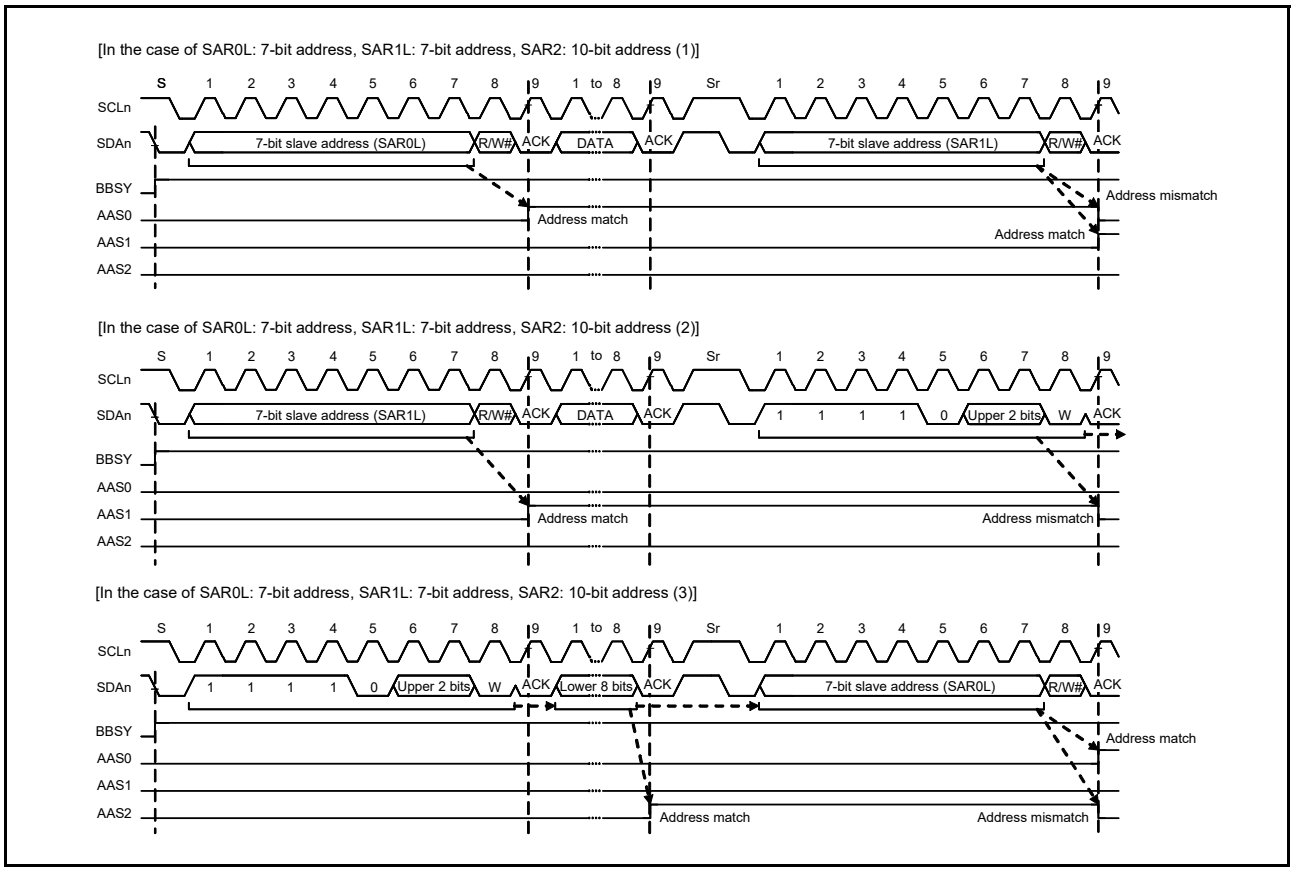


Figure 31.26 AASy flag set and clear timing with 7-bit and 10-bit address formats mixed

31.7.2 Detection of General Call Address

The IIC provides detection of the general call address (0000 000b + 0 [W]). General call address detection 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 are set to 1 on the rising edge of the 9th cycle of 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 whether the general call address is transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

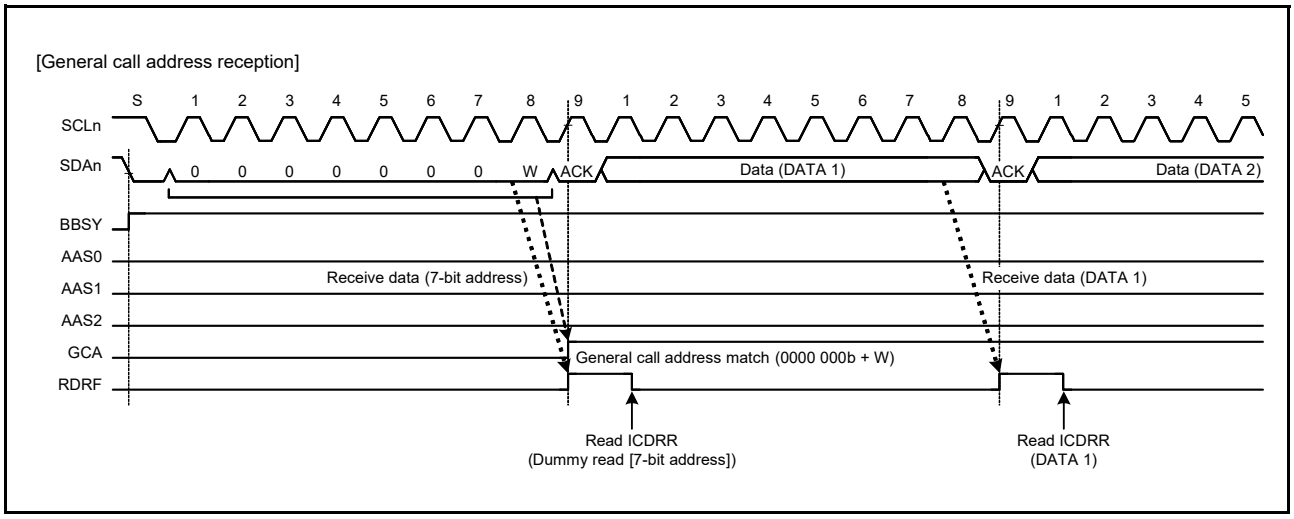


Figure 31.27 Timing of GCA flag setting during reception of general call address

31.7.3 Device ID Address Detection

The IIC module provides detection of device ID address in compliance with the I2C bus specification (Rev. 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, the IIC recognizes the address as a device ID, sets the DID flag in ICSR1 to 1 on the rising edge of the 8th 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 the detection of a restart condition. 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 of the information that must be included in device ID fields, contact NXP Semiconductors.

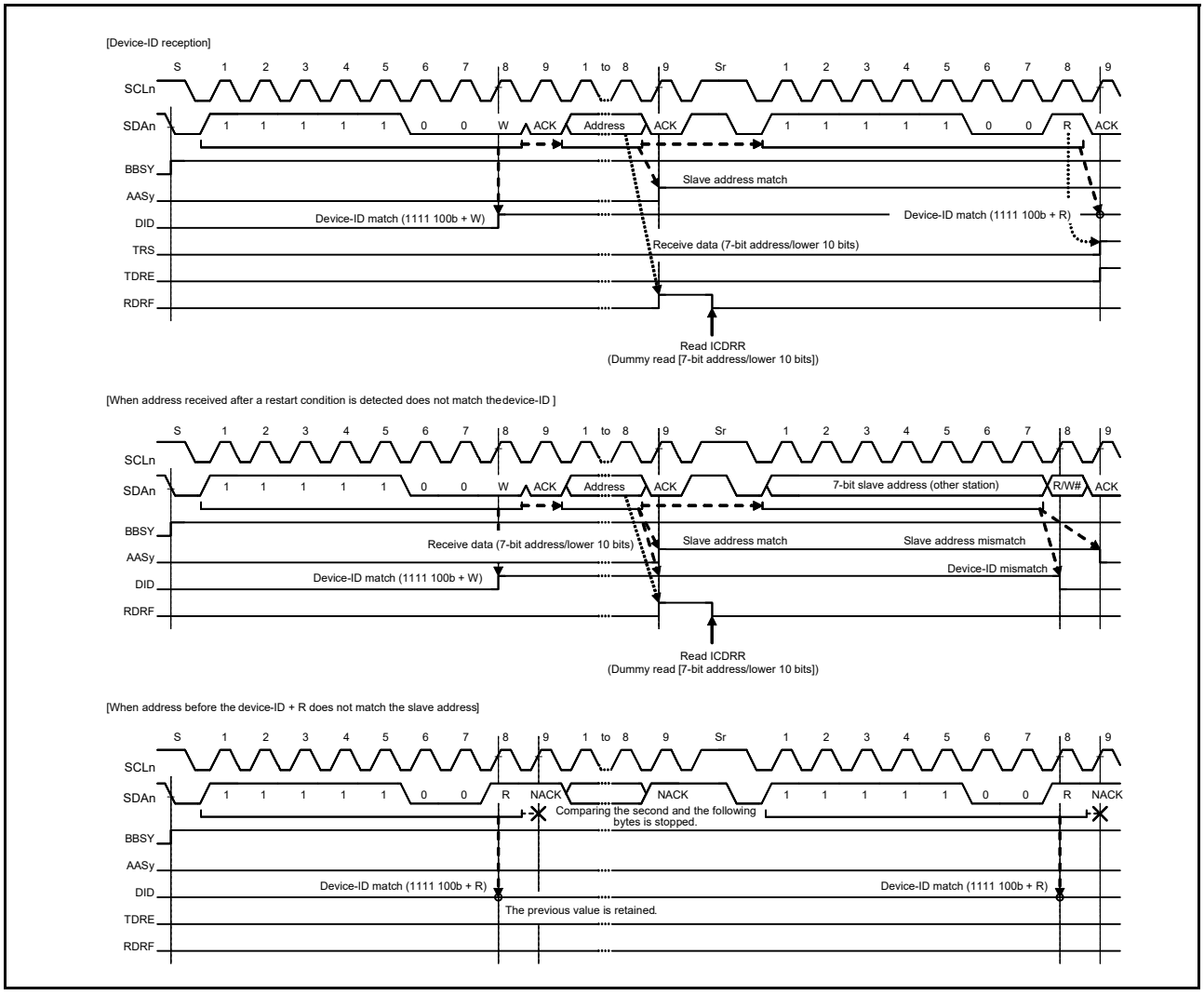


Figure 31.28 AASy/DID flag set/clear timing during reception of device ID

31.7.4 Host Address Detection

The IIC provides host address detection when operating in SMBus mode. When the HOAE bit in the ICSER register is set to 1 while the SMBS bit in the ICMR3 register is 1, the IIC can detect the host address (0001 000b) in slave receive mode (ICCR2.MST and ICCR2.TRS bits = 00b).

When the IIC detects the host address, the HOA flag in the ICSR1 register is set to 1 on the rising edge of the 9th SCL clock cycle. At the same time, the RDRF flag in the ICSR2 register is set to 1 if the R/W# bit is 0. This causes a receive data full interrupt (IICn_RXI) to be generated. The HOA flag indicates that the host address was detected.

If the bit following the host address (0001 000b) is a read 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.

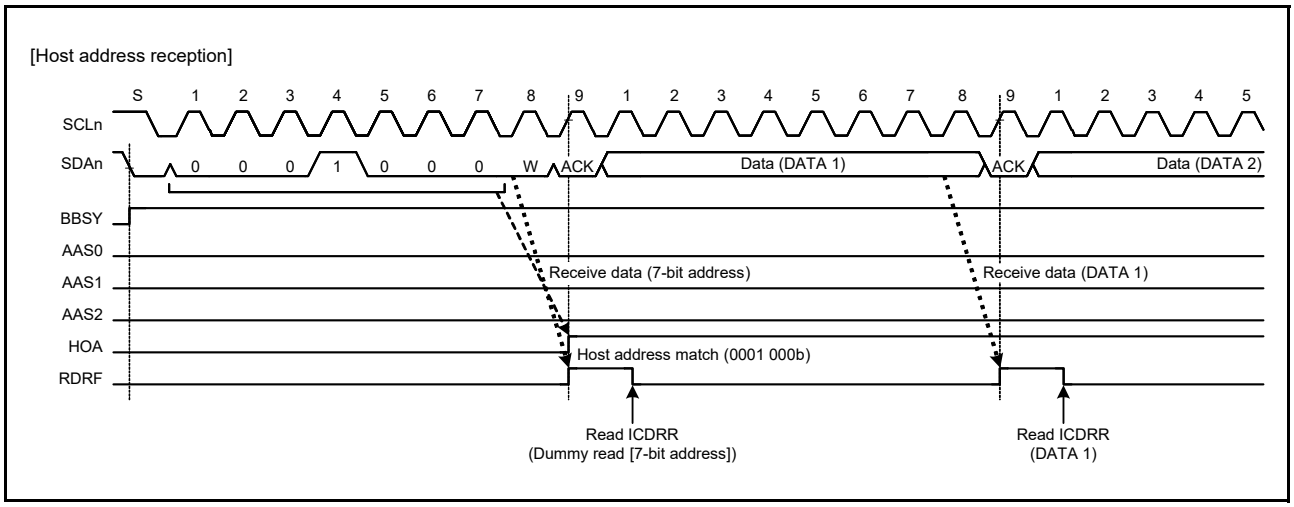


Figure 31.29 HOA flag set timing during reception of host address

31.8 Wakeup Function

The IIC provides a wakeup function that causes the MCU to transition from Software Standby mode to normal operation. The wakeup function enables the reception of data when the system clock is stopped, and it generates a wakeup interrupt signal on the match of the slave address of the received data. This wakeup interrupt signal triggers the return to normal operation.

The wakeup function has four operation modes:

- Normal wakeup mode 1
- Normal wakeup mode 2
- Command recovery mode
- EEP response mode.

Table 31.9 describes the behavior in these modes.

Table 31.9 Wakeup operation modes

Operation mode	ACK response timing	ACK response before wakeup	SCL state during wakeup
Normal wakeup mode 1	Before wakeup	ACK	Fixed low
Normal wakeup mode 2	After wakeup	Before wakeup: no response After wakeup: ACK response	Fixed low
Command recovery mode	Before wakeup	ACK	Open
EEP response mode	Before wakeup	NACK	Open

Precautions on the use of the wakeup function

- Disable the wakeup function (WUE = 0) after a wakeup interrupt triggers the transition from the Software Standby mode to normal operation.
- Do not change the content of the IIC registers while WUF = 0, even if the wakeup interrupt recovers the system clock. Specify the register settings after confirming that WUF = 1.
- Set WUE = WUIE = 1 and MST = TRS = 0 (slave reception mode) before entering the Software Standby mode.
- Do not invoke the Software Standby mode while BBSY = 1.
- The wakeup function supports the 7-bit slave address of slave address register SARL0, the general call address, and the host address. 10-bit slave addresses, SARL1 and SARL2, are not supported.
- When the wakeup function is enabled, disable the interrupts selectable in the TIE, TEIE, RIE, NAKIE, SPIE, STIE, ALIE, and TMOIE bits in the ICIER register.
- When the wakeup function is enabled, do not use the timeout function.
- If the transition from Software Standby mode is triggered by an interrupt other than a wakeup interrupt, for example IRQn, the WUF flag is not set to 1.

31.8.1 Normal Wakeup Mode 1

This section describes the behavior, timing, and an example operation in 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

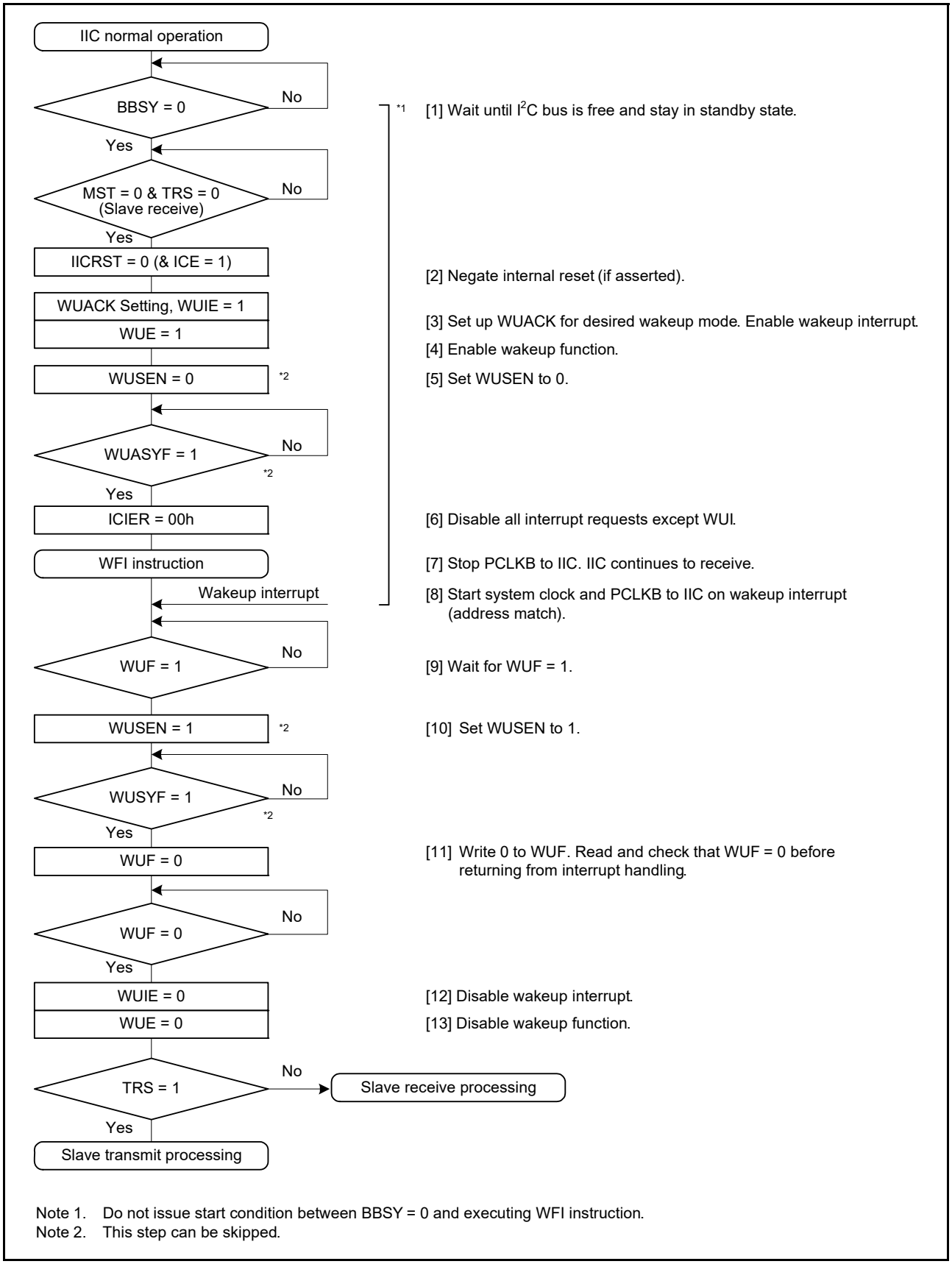
After wakeup: Normal operation continues.

Figure 31.30 shows an operation example and Figure 31.32 shows the detailed timing.

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.

Note 1. Between the 9th clock cycle and 1st clock cycle during wakeup, WAIT = 1 does not work.

If the transition from Software Standby mode is triggered by an interrupt other than a wakeup interrupt, for example IRQn, the WUF flag is not set to 1. Figure 31.31 shows an operation example.



Note 1. Do not issue start condition between BBSY = 0 and executing WFI instruction.
Note 2. This step can be skipped.

Figure 31.30 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.

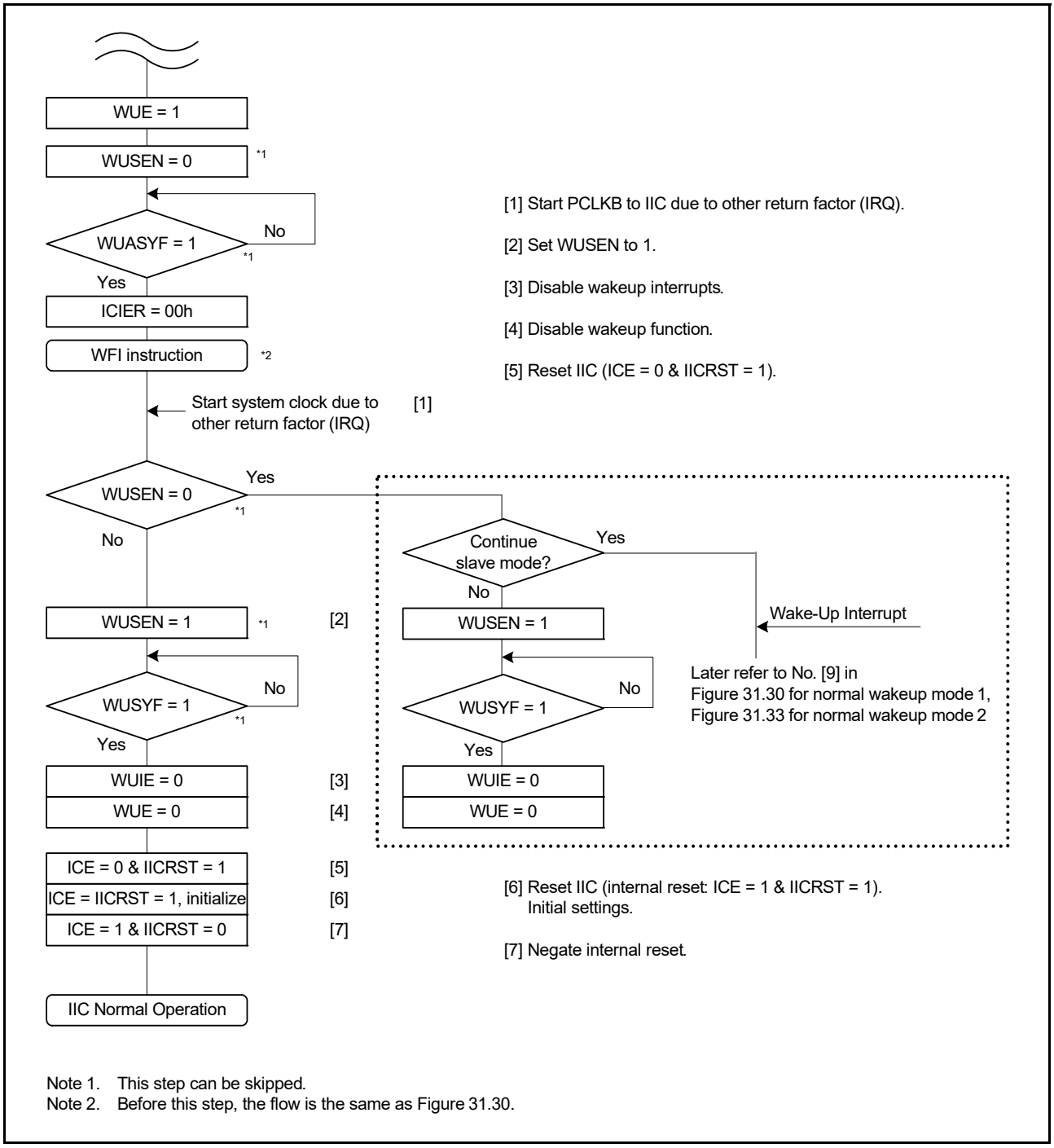


Figure 31.31 Example operation of normal wakeup modes 1 and 2 when wakeup is triggered by an interrupt other than IIC wakeup interrupt, for example, IRQn

Note: For details of the IIC initial settings, see section 31.3.2, Initial Settings.

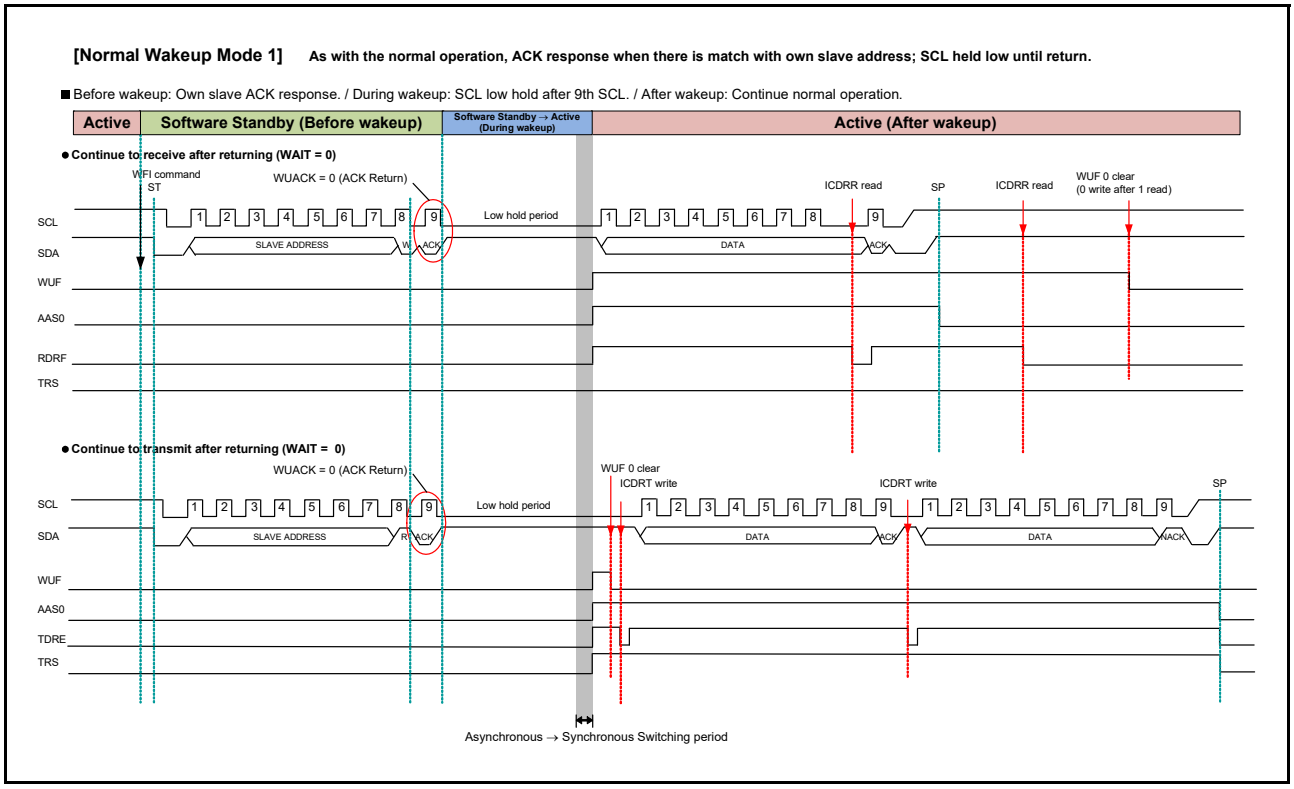


Figure 31.32 Timing of normal wakeup mode 1

31.8.2 Normal Wakeup Mode 2

This section describes the behavior, timing, and an example operation in normal wakeup mode 2.

In normal wakeup mode 2, a wakeup interrupt triggered by the match of the slave address initiates the transition to normal operation as follows:

Before wakeup: No response to the 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.

For an example operation in normal wakeup mode 2, see Figure 31.33. Figure 31.34 shows the detailed timing.

If the slave address does not match, the SCL line is not held low after the 8th SCL clock cycle, and the slave operation continues.

If the transition from Software Standby mode is triggered by an interrupt other than a wakeup interrupt, for example IRQn, the WUF flag is not set to 1. Follow the operation shown in Figure 31.31.

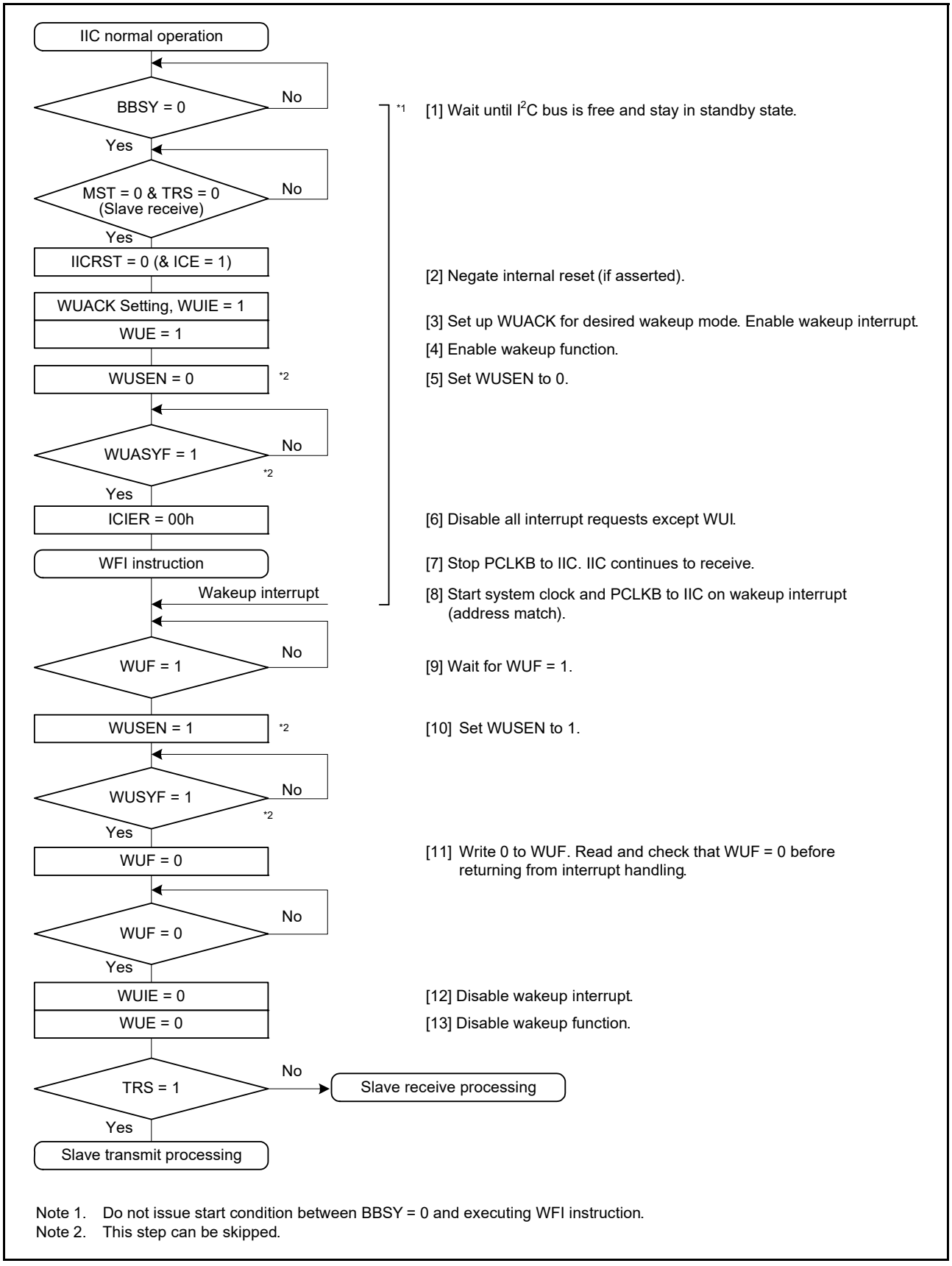


Figure 31.33 Example operation of normal wakeup mode 2 when wakeup is triggered by a wakeup interrupt on match of the slave address

Note: See Precautions on the use of the wakeup function.

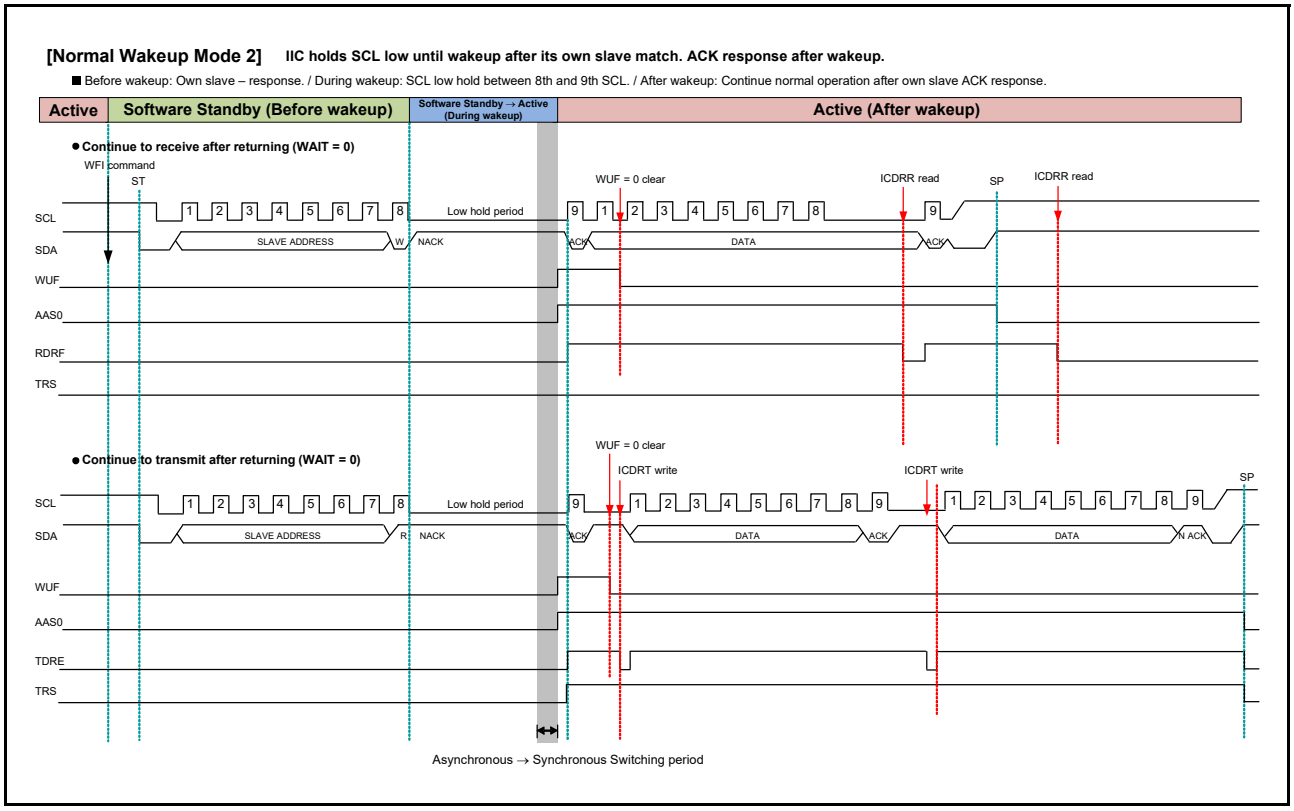


Figure 31.34 Timing of normal wakeup mode 2

31.8.3 Command Return Mode/ EEP Response Mode (Special Wakeup Mode)

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, the other IIC devices can use the I2C bus during this period.

A wakeup interrupt triggered by the 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 the IIC initialization.

For an example operation in command recovery mode and EEP response mode, see Figure 31.35. Figure 31.37 provides the detailed timing.

If the slave address does not match, the slave operation continues.

Note 1. 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 2. 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, AAS0, AAS1, and AAS2 in the ICSR1 register.

If the transition from Software Standby 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. Follow the operation shown in Figure 31.36.

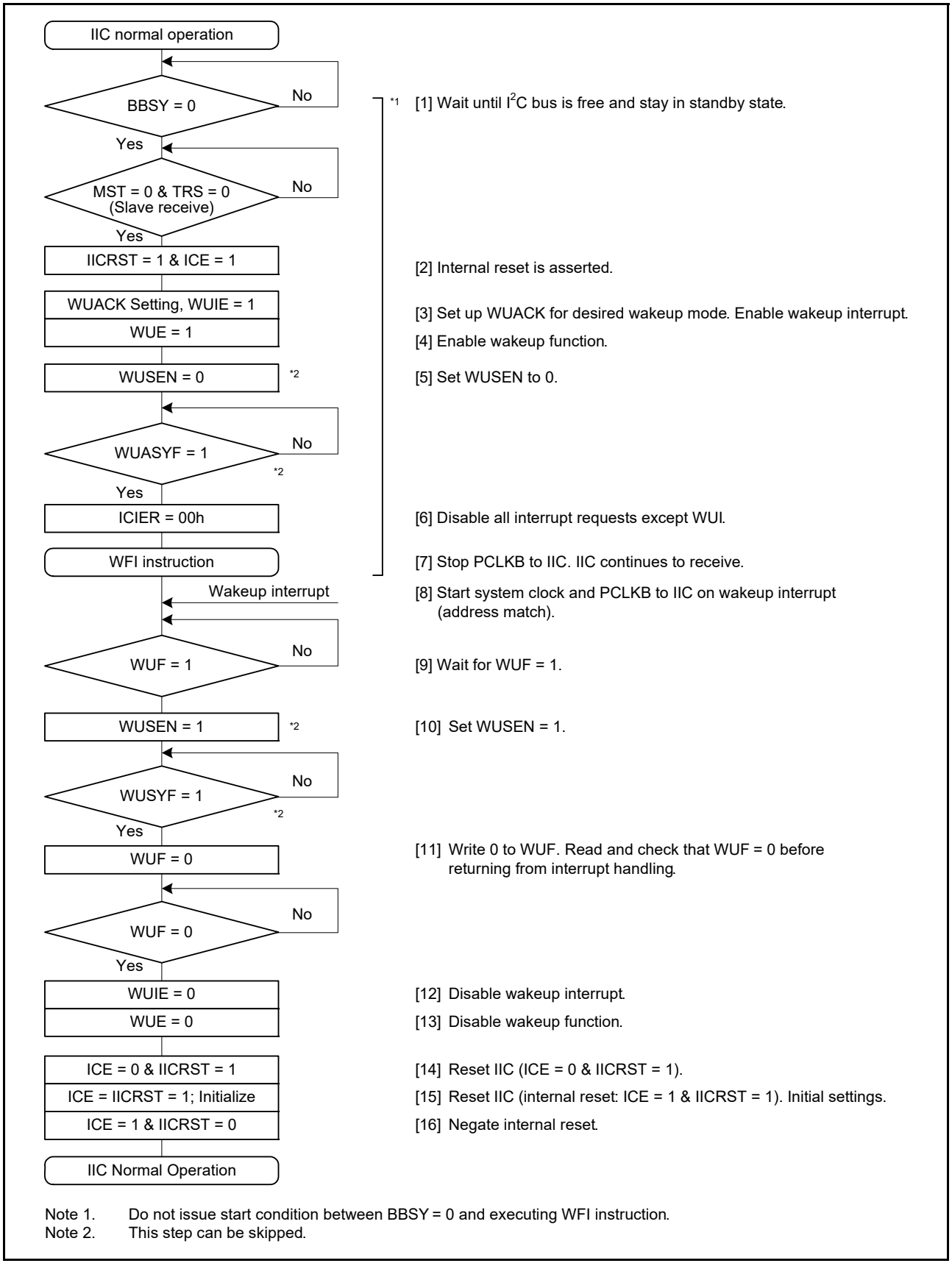


Figure 31.35 Example operation of command recovery and EEP response modes when wakeup is triggered by a wakeup interrupt on a match of the slave address

Note: See Precautions on the use of the wakeup function.

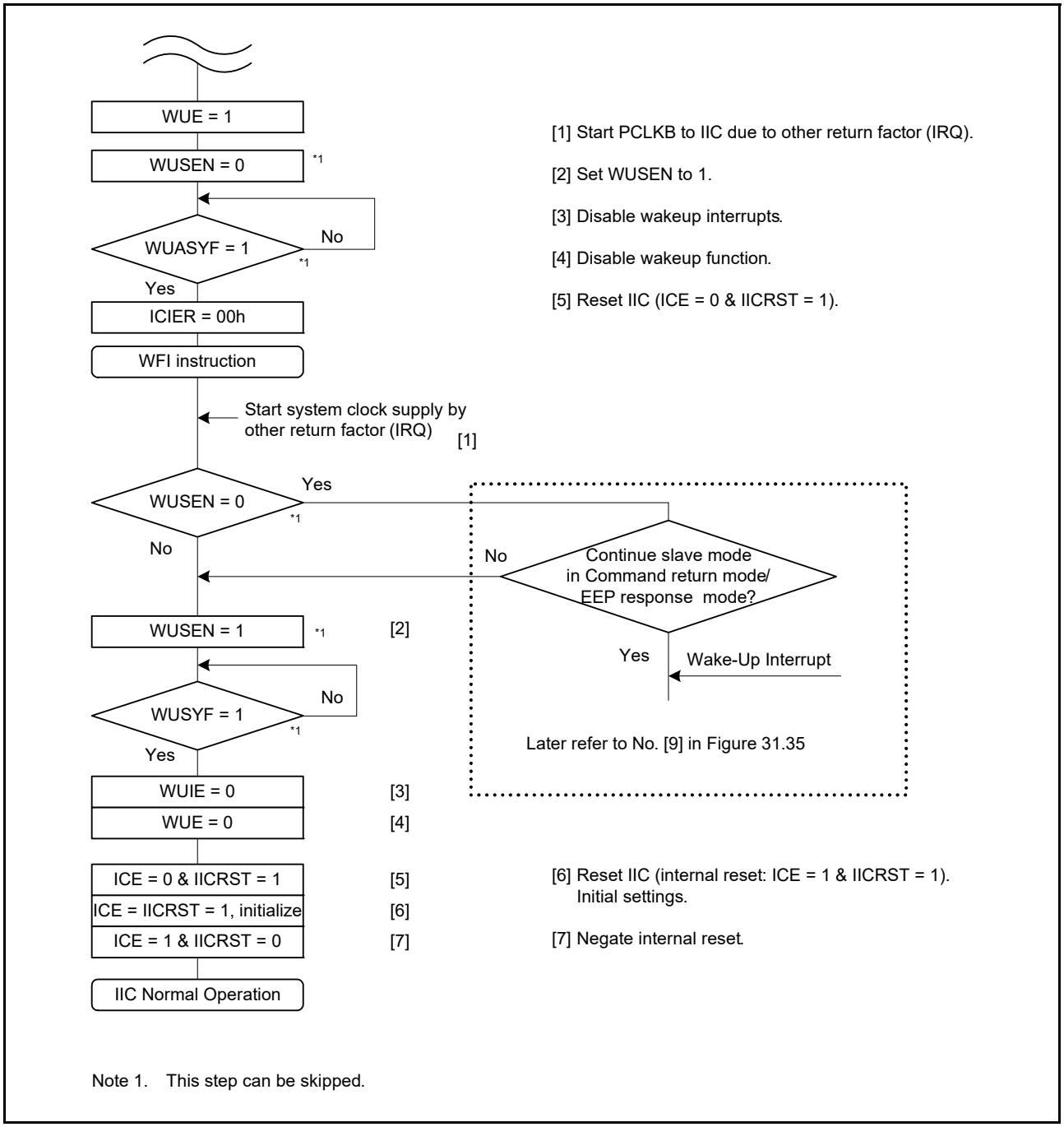


Figure 31.36 Example operation of command recovery mode and EEP response mode when wakeup is triggered by an interrupt other than IIC wakeup interrupt, for example, the IRQn

Note: For details of the IIC initial settings, see section 31.3.2, Initial Settings.

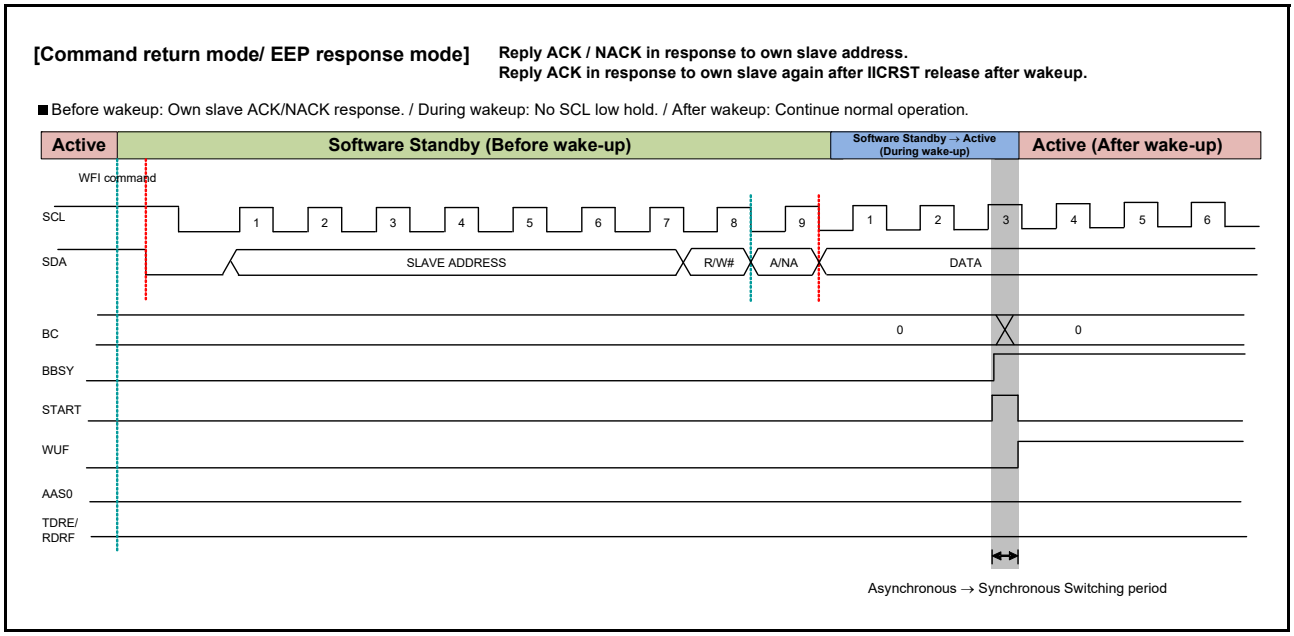


Figure 31.37 Timing of command recovery mode and EEP response mode

31.8.4 Precautions for WFI Instruction Execution

In the wakeup function examples shown in Figure 31.30, Figure 31.33, and Figure 31.35, make sure that the start condition is not issued during the period from the setting of BBSY = 0 to the execution of the WFI instruction.

When a start condition is issued during this period, NACK is returned after the reception of the first byte of the first data block. Detection of the start or restart condition then enables the wakeup function.

31.9 Automatic Low-Hold Function for SCL

31.9.1 Function to Prevent Wrong Transmission of Transmit Data

If the I2C Bus Shift Register (ICDRS) is empty when data has not been written to the I2C Bus Transmit Data Register (ICDRT) with the IIC in transmission mode (ICCR2.TRS = 1), the SCLn line is automatically held at the low level 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.

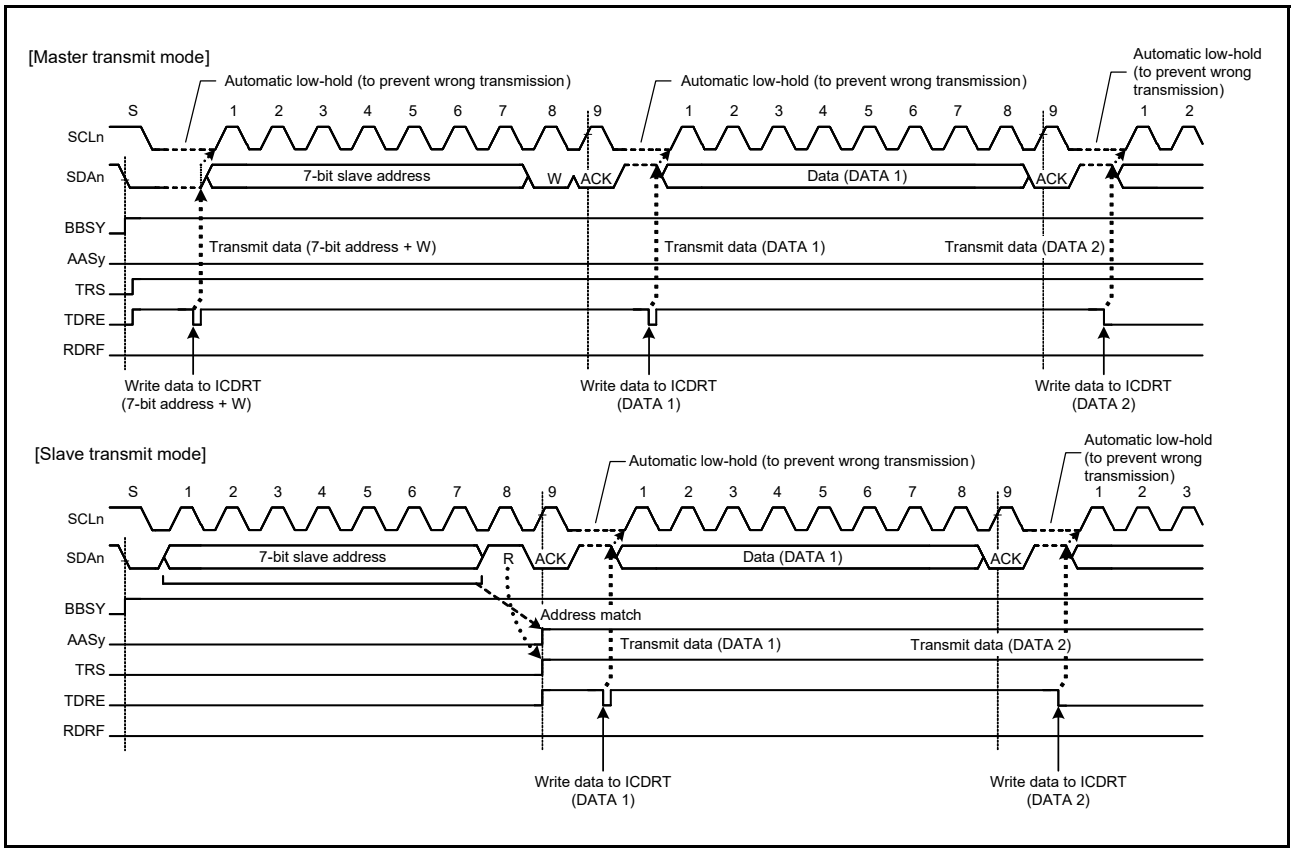


Figure 31.38 Automatic low-hold operation in transmit mode

31.9.2 NACK Reception Transfer Suspension Function

This function suspends transfer operation when NACK is received in transmit mode (ICCR2.TRS = 1). This function is enabled when the NACKE bit in the ICFER register is set to 1. If the next transmit data is already written (ICSR2.TDRE = 0) 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 (ICSR2.NACKF = 1), transmit and receive operations are discontinued. To restore transmit and receive operations, set the NACKF flag to 0. In master transmit mode, after a restart or stop condition is issued, set the NACKF flag to 0, and then issue a start condition again.

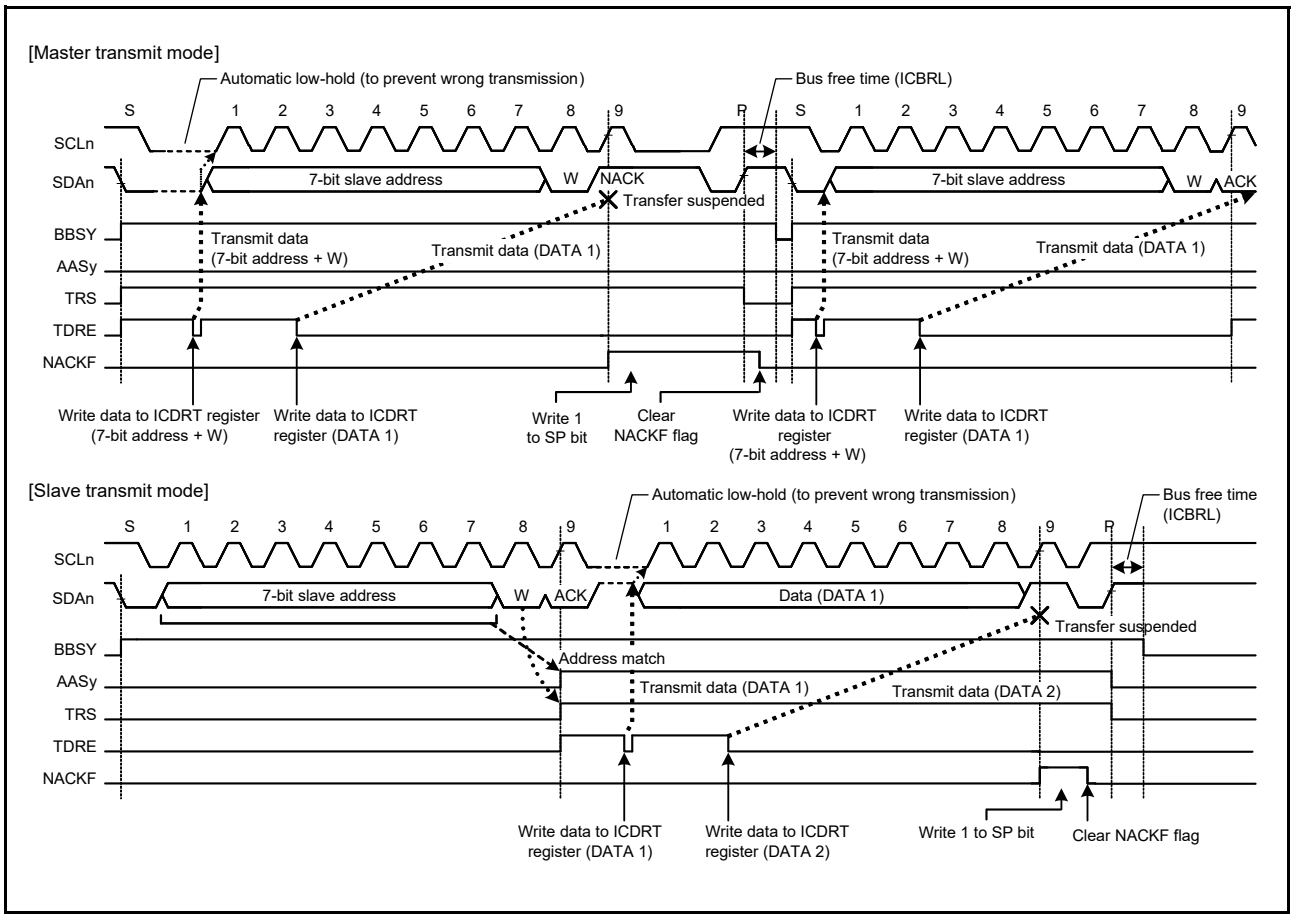


Figure 31.39 Suspension of data transfer when NACK is received (NACKE = 1)

31.9.3 Function to Prevent Failure to Receive Data

If response processing when receive data (ICDRR) read is delayed for a period of one transfer frame or more with receive data full (ICSR2.RDRF = 1) in receive mode (ICCR2.TRS = 0), the IIC holds the SCLn line low automatically immediately before the next data is received to prevent failure to receive data.

This function is also 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 disturb 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 the ICMR3 register.

(1) 1-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the WAIT bit in the ICMR3 register 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 ICMR3.ACKBT bit value 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 the ICDRR register, 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 receive mode 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 the ICMR3 register 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 (receive data full) in ICSR2 is set to 1 on the rising edge of the 8th SCL clock cycle, and the SCLn line is automatically held low on the falling edge of the 8th SCL clock cycle. This low-hold is released by writing to the ACKBT bit in the ICMR3 register, 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 receive mode or slave receive mode.

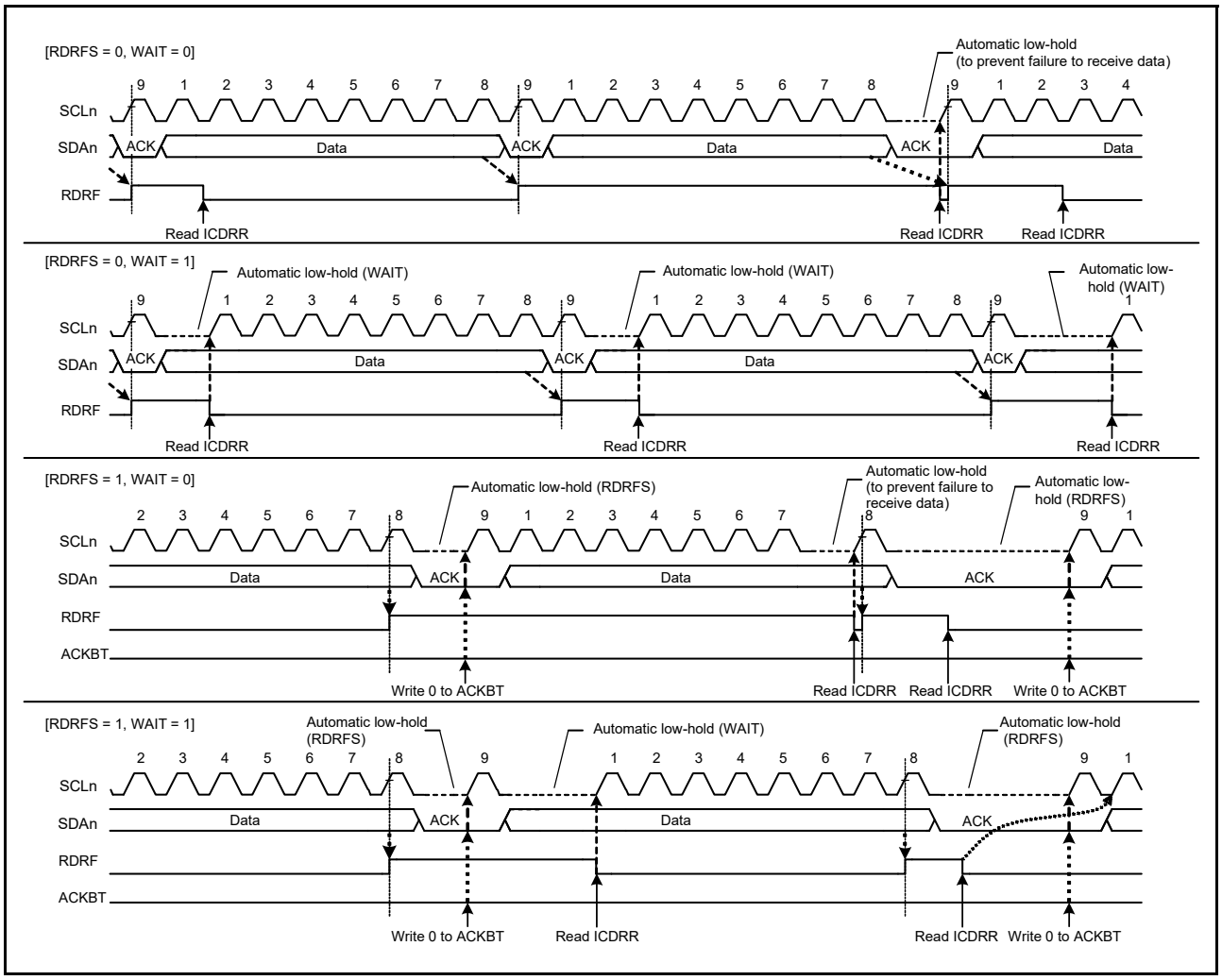


Figure 31.40 Automatic low-hold operation in receive mode using RDRFS and WAIT bits

31.10 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I2C bus standard, the IIC has functions to prevent double-issue of a start condition, detect arbitration-lost during transmission of NACK, and detect arbitration-lost in slave transmit mode.

31.10.1 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 issue 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 the arbitration. This prevents a failure of transfer resulting from a start condition 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, the IIC loses the 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 register is 1 (master arbitration-lost detection enabled).

[Master arbitration-lost conditions]

- Non-matching 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 ICCR2.BBSY flag is set to 0 (erroneous issuing of a start condition)
- Setting of the ICCR2.ST bit 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).

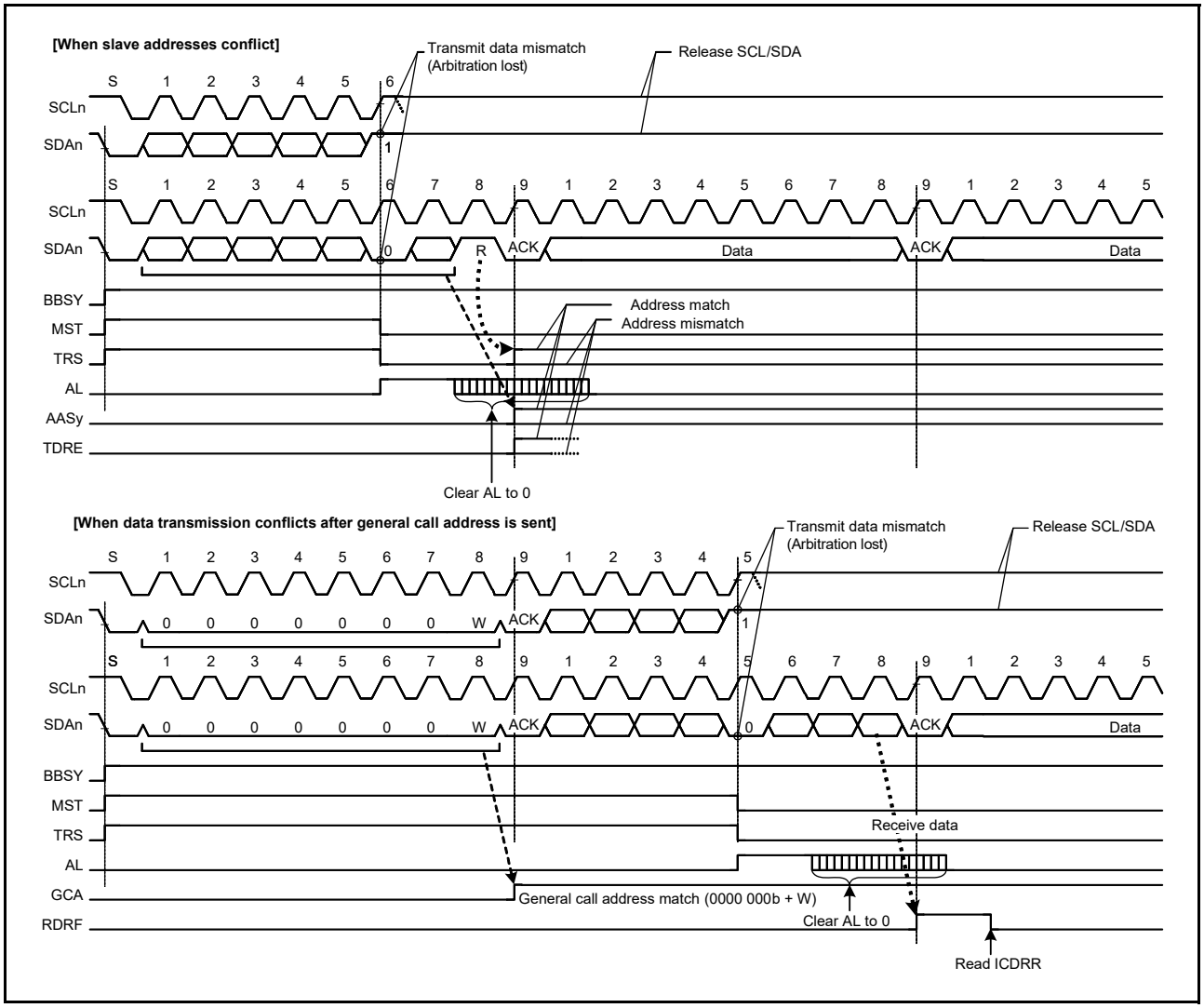


Figure 31.41 Examples of master arbitration-lost detection (MALE = 1)

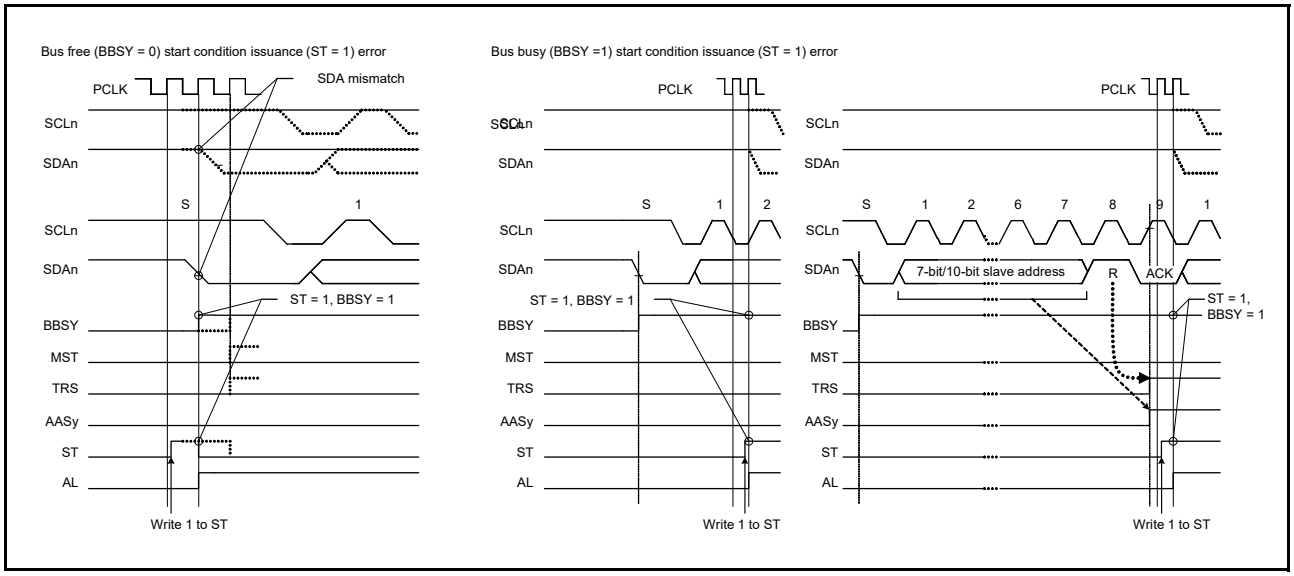


Figure 31.42 Arbitration-lost when start condition is issued (MALE = 1)

31.10.2 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 during transmission of NACK in receive mode. Arbitration is lost because of a conflict between NACK transmission and ACK transmission 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 31.43 shows an example of arbitration-lost detection during transmission of NACK.

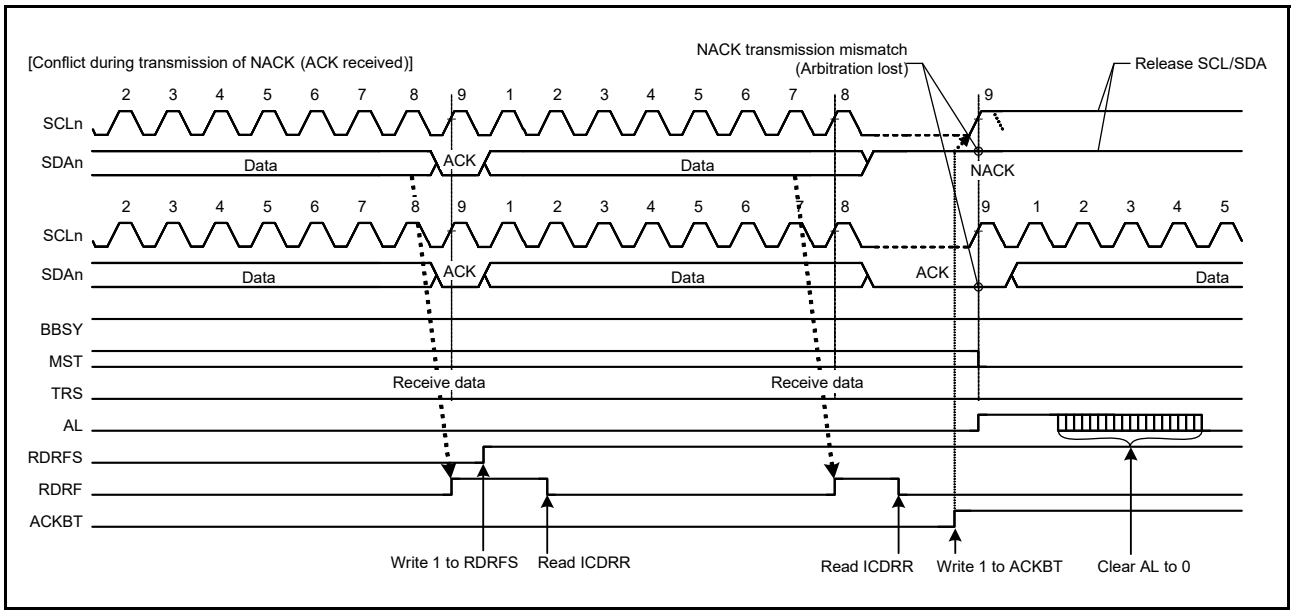


Figure 31.43 Example of arbitration-lost detection during transmission of NACK (NALE = 1)

The following description 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 both master A and 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 necessary 4 bytes of data. 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. Therefore, 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 FFh transmission processing, which is necessary 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 the ICFER register 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 (ICMR3.ACKBT = 1).

31.10.3 Slave Arbitration-Lost Detection (SALE Bit)

This function causes arbitration to be lost if the transmit data and the level on the SDAn line do not match 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, or processing for the transmission of FFh.

The IIC detects slave arbitration-lost when the following condition is met with the SALE bit in the ICFER register 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).

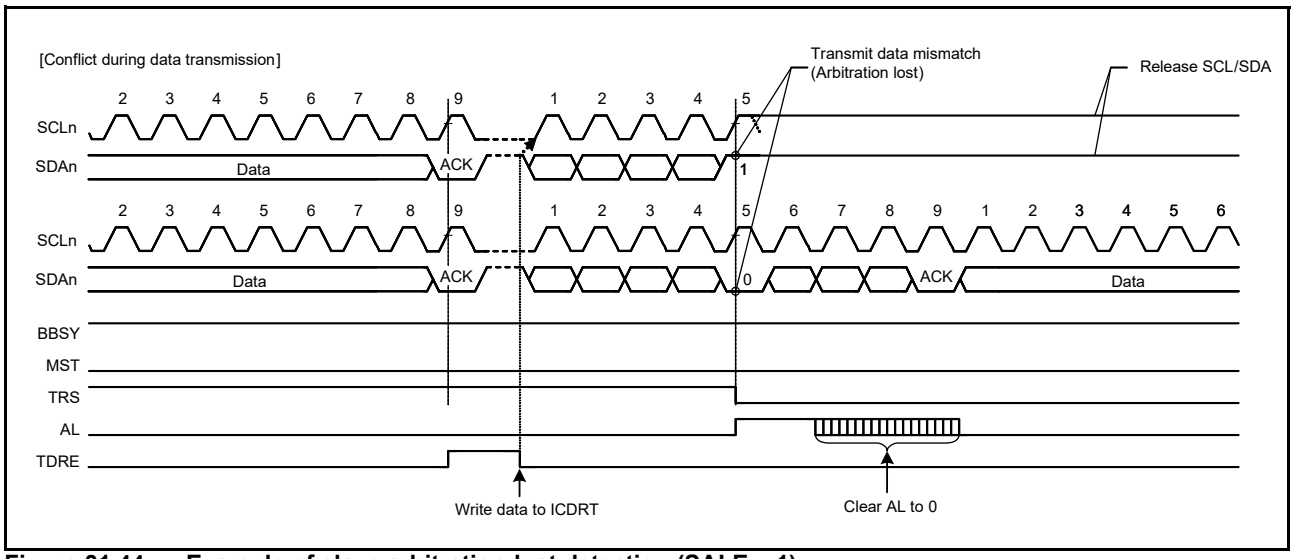


Figure 31.44 Example of slave arbitration-lost detection (SALE = 1)

31.11 Start, Restart, and Stop Condition Issuing Function

31.11.1 Issuing a Start Condition

The IIC issues a start condition when the ST bit in the ICCR2 register is set to 1.

When the ST bit is set to 1, a start condition request is made. The IIC issues a start condition when the BBSY flag in the ICCR2 register 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 the ICBRH register and the start condition hold time elapse.
3. Drive the SCLn line low (high level to low level).
4. Detect low level of the SCLn line and ensure that the low-level period of the SCLn line set in the ICBRL register elapses.

31.11.2 Issuing a Restart Condition

The IIC issues a restart condition when the RS bit in the ICCR2 register is set to 1.

When the RS bit is set to 1, a restart condition request is made. 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 that the low-level period of SCLn line set in the ICBRL register elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level of the SCLn line and ensure that the time set in the ICBRL register and the restart condition setup time elapse.
5. Drive the SDAn line low (high level to low level).
6. Ensure that the time set in the ICBRH register and the restart condition hold time elapse.
7. Drive the SCLn line low (high level to low level).
8. Detect a low level of the SCLn line and ensure the low-level period of SCLn line set in the ICBRL register elapses.

Note: When issuing restart condition requests, write the slave address to the ICDRT register after confirming that ICCR2.RS is 0. Data written while ICCR2.RS is 1 is not forwarded because of the retransmission condition before the occurrence.

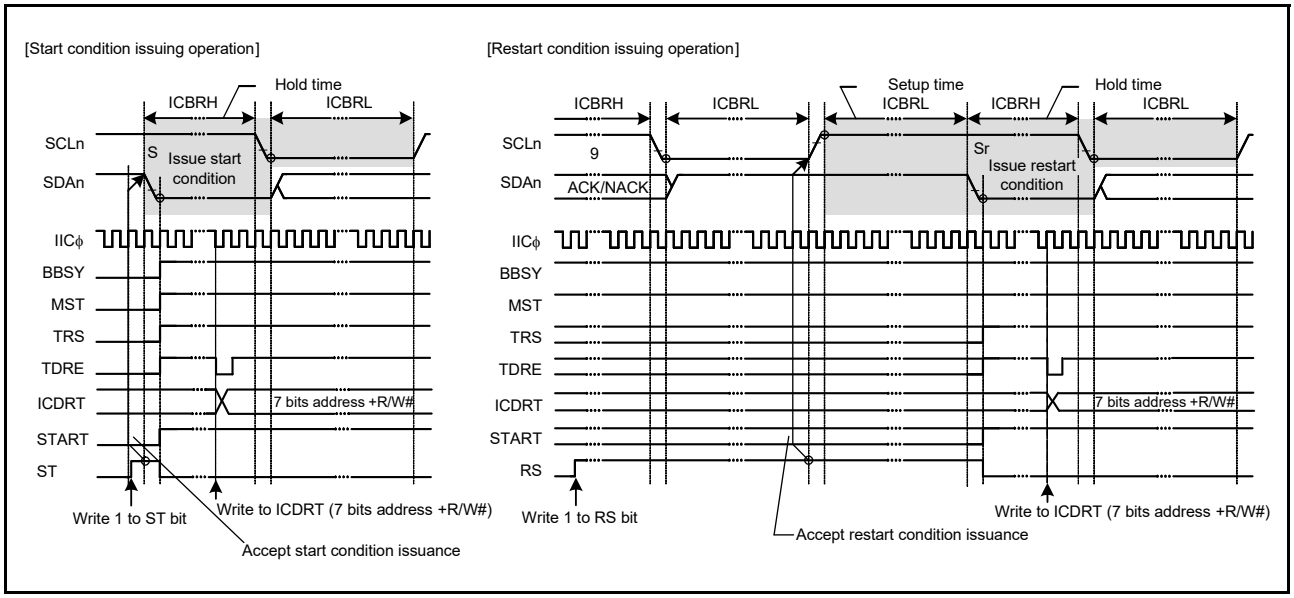


Figure 31.45 Start or restart condition issue timing (ST and RS bits)

Figure 31.46 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 details provided in section 31.3.2, Initial Settings.
2. Read the IICR2.BBSY flag to check that the bus is free, and then set the ICCR2.ST bit to 1 (start condition request). On receiving the request, the IIC issues a start condition. At the same time, the ICSR2.BBSY flag and ICSR2.START flag 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. 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 sets to 1 when the TRS bit is set to 1.
3. Check that the ICSR2.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to the ICDRT register. After the transmit data is written to the ICDRT register, the TDRE flag is automatically set to 0, the data is transferred from the ICDRT register to the ICDRS register, and the TDRE flag again sets 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 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 the ICDRT register as the first address transmission. Then, as the second address transmission, write the 8 lower bits of the slave address to the ICDRT register.
4. After confirming that the TDRE flag in ICSR2 is 1, write the data for transmission to the ICDRT register. The IIC automatically holds the SCLn line low until the data for transmission is ready, and a restart condition 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 ICSR2.TEND flag returns to 1. Then, after checking that the ICSR2.START flag is 1, set the ICSR2.START flag to 0.
6. Set the ICCR2.RS bit to 1 (restart condition issue request). On receiving the request, the IIC issues a restart condition.
7. After checking that the ICSR2.START flag is 1, write the value for transmission (the slave address and the R/W# bit) to the ICDRT register.

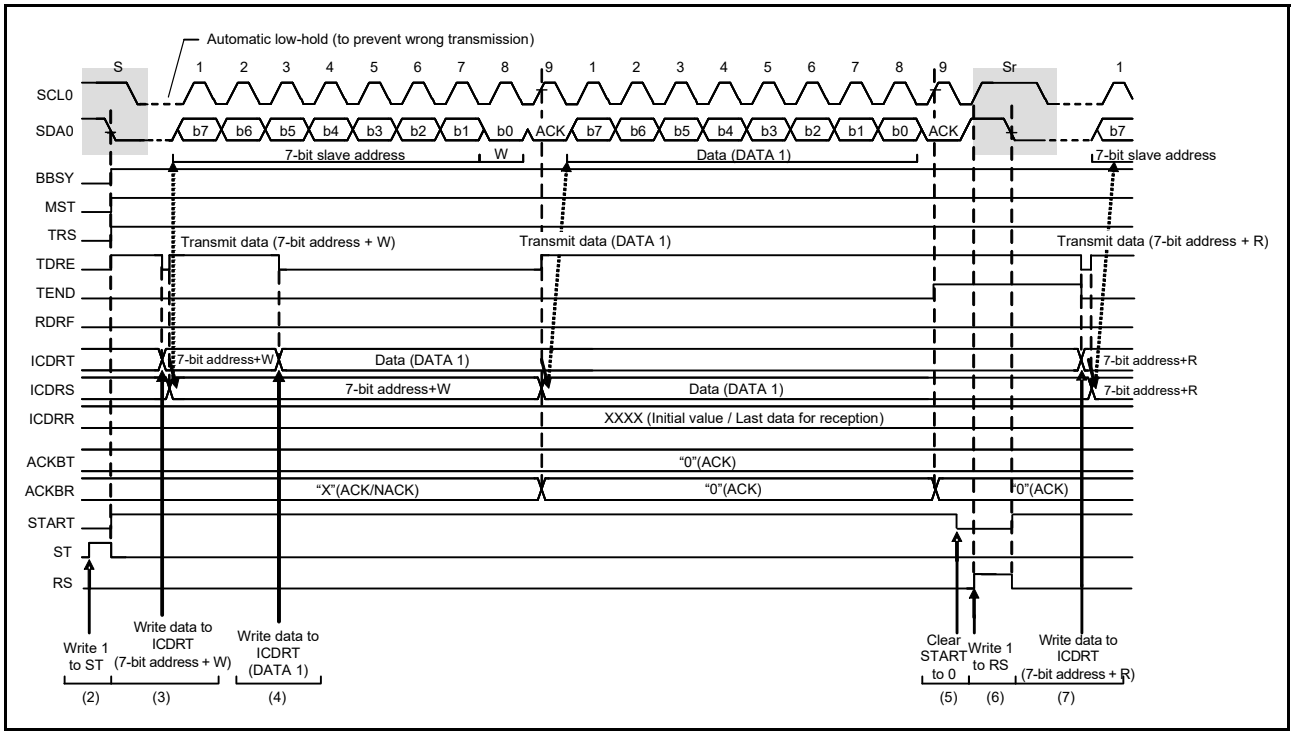


Figure 31.46 Restart condition issue timing after master transmission

31.11.3 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 ICCR2.BBSY flag is 1 (bus busy state) and the ICCR2.MST bit is 1 (master mode).

To issue a stop condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure that the low-level period of SCLn line set in the ICBRL register elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level of the SCLn line and ensure that the time set in the ICBRH register and the stop condition setup time elapse.
5. Release the SDAn line (low level to high level).
6. Ensure that the time set in the ICBRL register and the bus free time elapse.
7. Clear the BBSY flag to 0 to release the bus mastership.

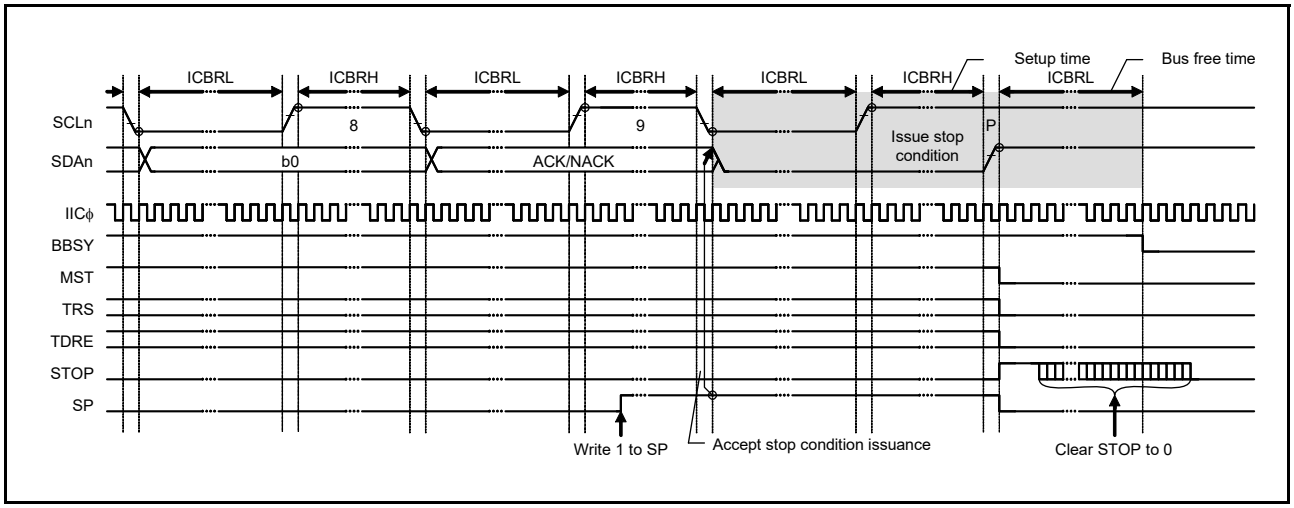


Figure 31.47 Stop condition issue timing (SP bit)

31.12 Bus Hanging

If the clock signals from the master and slave devices are out of synchronization because of noise or other factors, the I2C 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
- A function for the output of an extra SCL clock cycle to release the bus from a hung state because of clock signals being out of synchronization
- The IIC reset function
- An internal reset function.

By checking the SCLO, SDAO, SCLI, and SDAI bits in the ICCR1 register, it is possible to determine whether the IIC or its communicating partner is placing the low level on the SCLn or SDAn lines.

31.12.1 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-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCLn line changes (rising or falling), 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 00h) and the bus is busy (ICCR2.BBSY flag is 1) in slave mode (ICCR2.MST bit is 0).
- The bus is free (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 the ICMR1 register as a count source. It functions as a 16-bit counter when long mode is selected (ICMR2.TMOS = 0) or a 14-bit counter when short mode is selected (ICMR2.TMOS = 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 the ICMR2 register. If both TMOL and TMOH bits are set to 0, the internal counter is disabled.

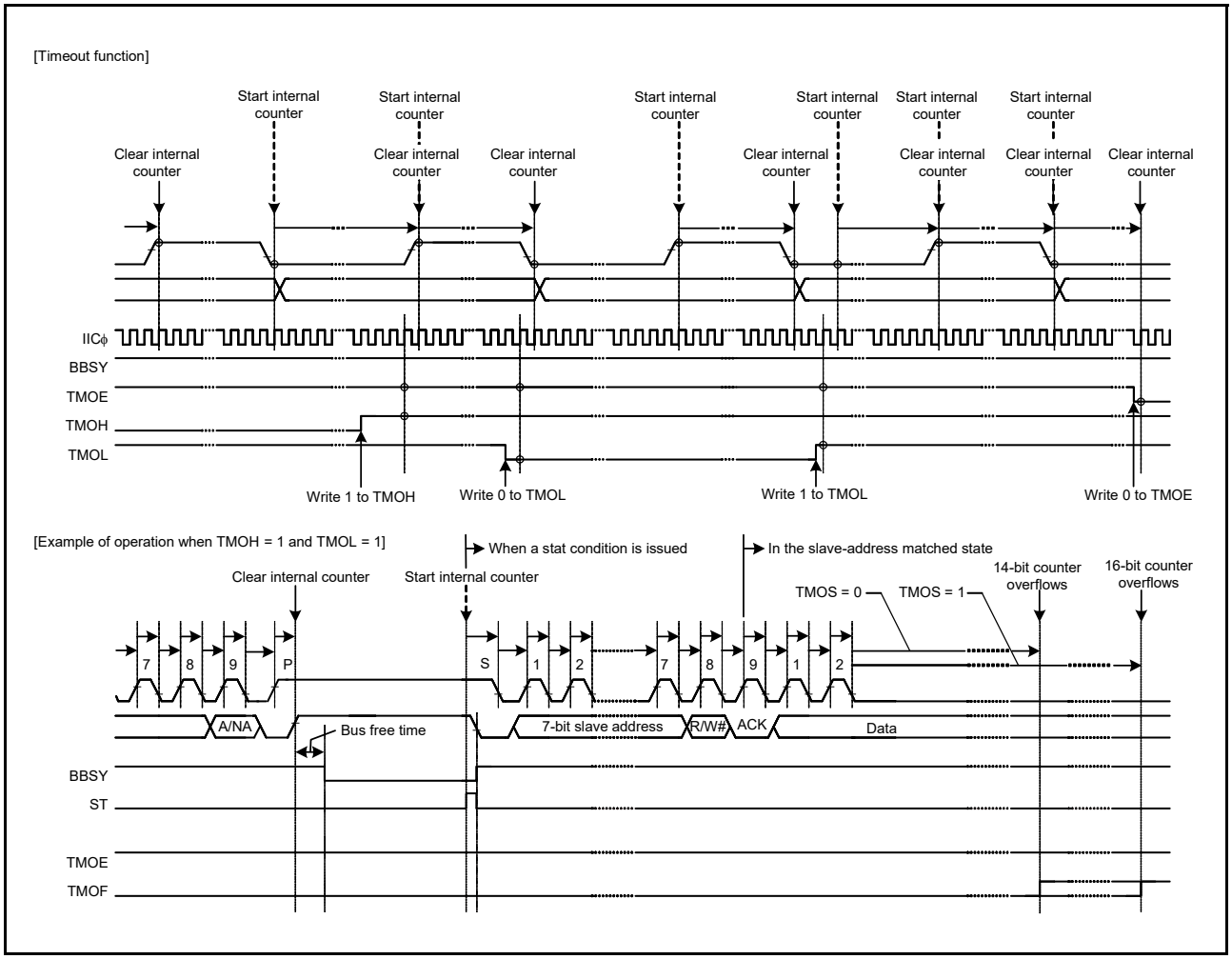


Figure 31.48 Timeout function (TMOE, TMOS, TMOH, and TMOL bits)

31.12.2 Extra SCL Clock Cycle Output Function

In master mode, this function outputs extra 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 uses single cycles of the SCL clock for a bus error when the IIC cannot issue a stop condition because the slave device is holding the SDAn line low. Do not use this function in normal situations. Using it when communications are proceeding correctly leads to malfunctions.

When the CLO bit in the ICCR1 register is set to 1 in master mode, a single cycle of the SCL clock at the transfer rate specified in the ICMR1.CKS[2:0] bits, 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. Therefore, additional extra clock cycles can be output consecutively by writing 1 to the CLO bit after having read CLO = 0.

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 the effects of 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 recover the bus from an unusable state. Release of the SDAn line by the slave device can be monitored by reading the ICCR1.SDAI bit. After confirming the release of the SDAn line by the slave device, complete communications by reissuing the stop condition.

Use this function with the MALE bit in the ICFER register set to 0 (master arbitration-lost detection disabled). If the MALE bit is set to 1 (enabled), arbitration is lost when the value of the ICCR1.SDAO bit does not match the state of the SDAn line.

[Output conditions for using the CLO bit in ICCR1]:

- When the bus is free (ICCR2.BBSY = 0) or in master mode (ICCR2.MST = 1 and ICCR2.BBSY = 1)
- When the communication device does not hold the SCLn line low.

Figure 31.49 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

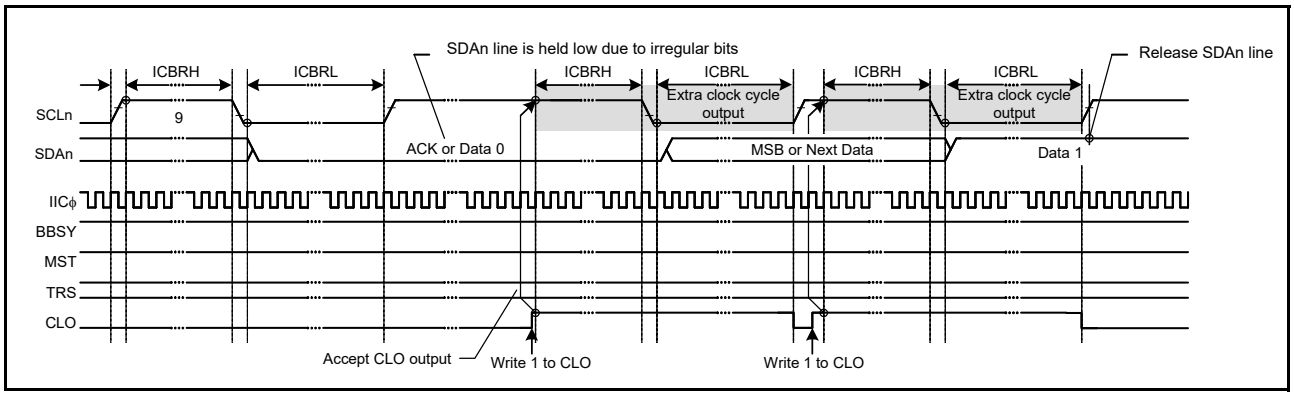


Figure 31.49 Extra SCL clock cycle output function (CLO bit)

31.12.3 IIC Reset and Internal Reset

The IIC has two types of resets:

- IIC reset, which initializes all registers, including the BBSY flag in the ICCR2 register
- 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, be sure to set the ICCR1.IICRST bit to 0. Both types of reset are effective 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 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 31.15, Register States when Issuing each Condition.

31.13 SMBus Operation

The IIC supports data communication conforming to the SMBus Specification (version 2.0). To perform SMBus communication, set the SMBS bit in the ICMR3 register to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus standard, set the ICMR1.CKS[2:0] bits, and the ICBRH and ICBRL registers. In addition, determine the values of the ICMR2.DLCS bit and the ICMR2.SDDL[2:0] bits 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 the ICBRL register 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-bit/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 ICFER.SALE bit to 1 to enable the slave arbitration-lost detection function.

31.13.1 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 (max.) 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 ICCR1.IICRST bit to issue an internal reset of the IIC. When an internal reset is issued, the IIC stops driving the bus for the SCLn pin and SDAn pin and makes the SCLn/SDAn pin 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), and 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 $T_{LOW:\ MEXT}$ 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 ICSR2.TEND flag in master transmit mode (master transmitter) and the ICSR2.RDRF flag in master receive mode (master receiver). Perform byte-wise transmit operations in master transmit mode, and hold the ICMR3.RDRFS bit 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 the ICDRT register).

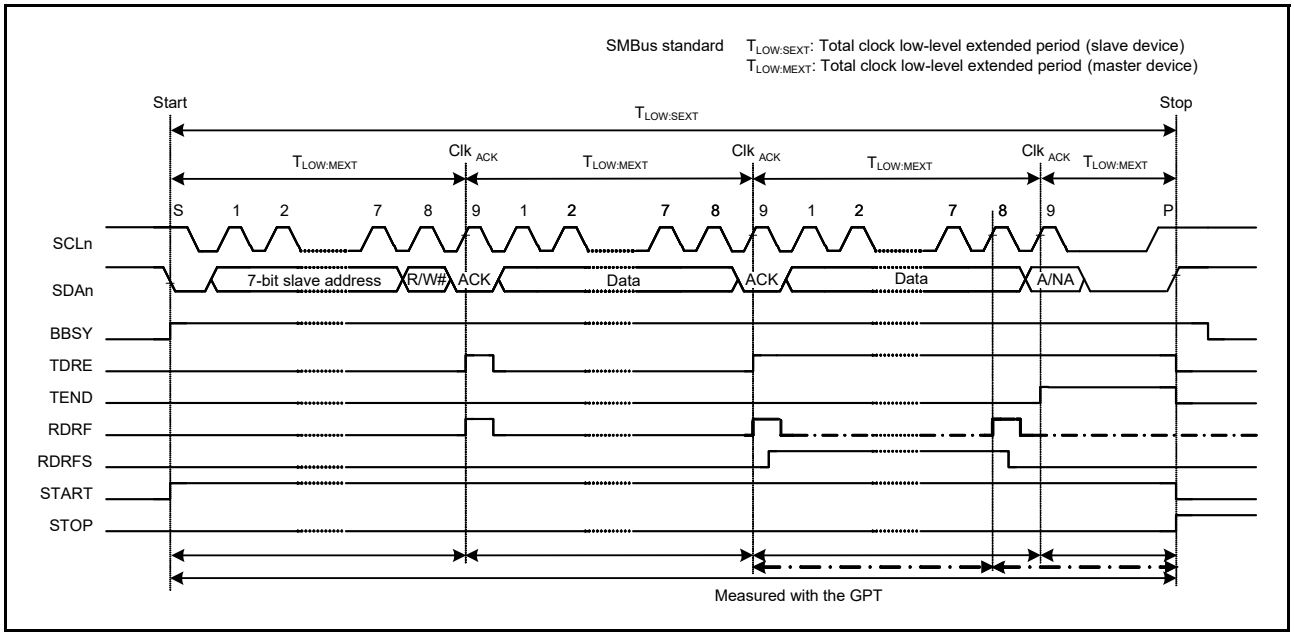


Figure 31.50 SMBus timeout measurement

31.13.2 Packet Error Code (PEC)

The MCU provides a CRC calculator that enables transmission of a packet error code (PEC) or allows checking the received data in SMBus data communication. For the CRC generating polynomials of the CRC calculator, see section 35, Cyclic Redundancy Check (CRC) Calculator.

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC Data Input Register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode can be checked by writing all receive data to the CRCDIR register 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 according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the ICMR3.RDRFS bit 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.

31.13.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communications 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.

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 has a function for detecting the host address. To detect the host address as a slave address, set the ICMR3.SMBS bit and the ICSER.HOAE bit to 1. Operation after the host address is detected is the same as normal slave operation.

31.14 Interrupt Sources

The IIC issues four 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.

Table 31.10 lists details of the interrupt requests. The receive data full and transmit data empty conditions can activate data transfer by the DTC or DMAC.

Table 31.10 Interrupt sources

Symbol	Interrupt source	Interrupt flag	DTC activation	DMAC activation	Interrupt condition
IICn_EEI*5	Transfer error/event generation	AL	Not possible	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	Possible	RDRF = 1, RIE = 1
IICn_TXI*1, *5	Transmit data empty	TDRE	Possible	Possible	TDRE = 1, TIE = 1
IICn_TEI*3, *5	Transmit end	TEND	Not possible	Not possible	TEND = 1, TEIE = 1
IIC0_WUI*4	Slave address match during wakeup function	WUF	Not Possible	Not possible	Slave address match Slave receive complete RWAK operation ASY0 = 1 WUIE = 1

Note: There is a delay time between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. When an interrupt flag is cleared or masked, read the relevant flag again to check whether clearing or masking has completed, and then return from interrupt handling. Not doing so creates the possibility of repeated processing of the same interrupt.

Note 1. Because IICn_TXI is an edge-detected interrupt, it does not require clearing. Additionally, the ICSR2.TDRE flag (a condition for IICn_TXI) is automatically set to 0 when transmit data is written to the ICDRT register or a stop condition is detected (ICSR2.STOP = 1).
Note 2. Because IICn_RXI is an edge-detected interrupt, it does not require clearing. Additionally, the ICSR2.RDRF flag (a condition for IICn_RXI) is automatically set to 0 when data is read from the ICDRR register.
Note 3. When using the IICn_TEI interrupt, clear the ICSR2.TEND flag in the IICn_TEI interrupt handling. The ICSR2.TEND flag is automatically set to 0 when transmit data is written to the ICDRT register or a stop condition is detected (ICSR2.STOP = 1).
Note 4. Only channel 0 has a wakeup function, therefore IIC0_WUI is for channel 0 only.
Note 5. Channel number (n = 0 to 2).

Clear or mask each flag during interrupt handling.

31.14.1 Buffer Operation for IICn_TXI and IICn_RXI Interrupts

If the conditions for generating an IICn_TXI and IICn_RXI interrupt are satisfied while the associated IR flag is 1, the interrupt request is not output for the ICU but saved internally. One request per source can be saved internally.

An interrupt request that is saved within the ICU is output when the value of the ICU.IELSRn.IR flag becomes 0. Internally saved interrupt requests are automatically cleared under normal usage conditions. Internally saved interrupt requests can also be cleared by writing 0 to the interrupt enable bit within the associated peripheral module.

31.15 Register States when Issuing each Condition

The IIC has 2 dedicated resets, IIC reset, and internal reset. Table 31.11 lists the register states when issuing each condition.

Table 31.11 Register states when issuing each condition

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	Reset	Saved	Saved	Saved	Saved
	SCLO, SDAO		Reset	Reset		
	Others			Saved		
ICCR2	BBSY	Reset	Reset	Saved	Set	Saved
	ST			Reset	Saved	Saved
	TRS, MST				Set or saved	Reset
	Others				Reset	Reset or saved
ICMR1	BC[2:0]	Reset	Reset	Reset	Reset	Saved
	Others			Saved	Saved	
ICMR2		Reset	Reset	Saved	Saved	Saved
ICMR3		Reset	Reset	Saved	Saved	Saved
ICFER		Reset	Reset	Saved	Saved	Saved
ICSER		Reset	Reset	Saved	Saved	Saved
ICIER		Reset	Reset	Saved	Saved	Saved
ICSR1		Reset	Reset	Reset	Saved	Reset
ICSR2	TDRE, TEND	Reset	Reset	Reset	Saved	Reset
	START				Set	
	STOP				Saved	Set
	Others					Saved
ICWUR		Reset	Reset	Saved	Saved	Saved
SARL0, SARL1, SARL2 SARU0, SARU1, SARU2		Reset	Reset	Saved	Saved	Saved
ICBRH, ICBRL		Reset	Reset	Saved	Saved	Saved
ICDRT		Reset	Reset	Saved	Saved	Saved
ICDRR		Reset	Reset	Saved	Saved	Saved
ICDRS		Reset	Reset	Reset	Saved	Saved
Timeout function		Reset	Reset	Operation	Operation	Operation
Bus free time measurement		Reset	Reset	Operation	Operation	Operation

31.16 Event Link Output

IIC0 to IIC2 handle 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 transfer, the associated event signal can be output to another module by the ELC.

31.16.1 Interrupt Handling and Event Linking

Each of the IIC interrupt types (see Table 31.10) has an enable bit to control enabling and disabling of the associated interrupt signal. An interrupt request signal is output for 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 settings of the interrupt enable bits. For details on interrupt sources, see Table 31.10.

31.17 Usage Notes

31.17.1 Settings for the Module-Stop State

The Module Stop Control Register B (MSTPCRB) can enable or disable IIC operation. The module is initially stopped after a reset. The registers become accessible on release from the module-stop state.

For details on Module Stop Control Register B, see section 11, Low Power Modes.

31.17.2 Notes on Starting Transfer

If the IR flag associated with the IIC interrupt is 1 when transfer is started (ICCR1.ICE = 1), use the following procedure to clear 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 unanticipated behavior of the IR flag.

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.

32. Controller Area Network (CAN) Module

32.1 Overview

The CAN module uses a message-based protocol to receive and transmit data between multiple slaves and masters in electromagnetically noisy applications. The module complies with the ISO 11898-1 (CAN 2.0A/CAN 2.0B) standard and supports up to 32 mailboxes, which can be configured for transmission or reception in normal mailbox and FIFO modes. Both standard (11-bit) and extended (29-bit) messaging formats are supported. The CAN module requires an additional external CAN transceiver.

Table 32.1 lists the features of the CAN module and Figure 32.1 shows the block diagram.

Table 32.1 CAN module specifications (1 of 2)

Parameter	Description
Data transfer	ISO11898-1 compliant for standard and extended frames
Bit rate	Programmable up to 1 Mbps (fCAN ≥ 8 MHz). fCAN: CAN clock source
Message box	32 mailboxes, with two selectable mailbox modes: • Normal mode: 32 mailboxes independently configurable for either transmission or reception • FIFO mode: 24 mailboxes independently configurable for either transmission or reception, with remaining mailboxes used for receive and transmit 4-stage FIFOs.
Reception	• Support for data frame and remote frame reception • Reception ID format selectable to only standard ID, only extended ID, or mixed IDs • Programmable one-shot reception function • Selectable between overwrite mode (unread message overwritten) and overrun mode (unread message saved) • Reception complete interrupt independently enabled or disabled for each mailbox.
Acceptance filter	• Eight acceptance masks (one for every four mailboxes) • Masks independently enabled or disabled for each mailbox.
Transmission	• Support for data frame and remote frame transmission • Transmission ID format selectable to only standard ID, only extended ID, or mixed IDs • Programmable one-shot transmission function • Broadcast messaging function • Priority mode selectable based on message ID or mailbox number • Support for transmission request abort, with abort completion confirmable in status flag • Transmission complete interrupt independently enabled or disabled for each mailbox.
Mode transition for bus-off recovery	Mode transition for the recovery from the bus-off state selectable to: • ISO11898-1 specification-compliant • Automatic invoking of CAN halt mode on bus-off entry • Automatic invoking of CAN halt mode on bus-off end • Invoking of CAN halt mode through software • Transition to error-active state through software.
Error status monitoring	• Monitoring of CAN bus errors, including stuff error, form error, ACK error, 15-bit CRC error, bit error, and ACK delimiter error • Detection of transition to error states, including error-warning, error-passive, bus-off entry, and bus-off recovery • Support reading of error counters.
Time stamping	• Time stamp function using a 16-bit counter • Reference clock selectable from 1-, 2-, 4- and 8-bit time periods.
Interrupt function	Support five interrupt sources: • Reception complete • Transmission complete • Receive FIFO • Transmit FIFO • Error interrupts.
CAN sleep mode	CAN clock stopped to reduce power consumption
Software support unit	Three software support units: • Acceptance filter support • Mailbox search support, including receive mailbox search, transmit mailbox search, and message lost search • Channel search support.

Table 32.1 CAN module specifications (2 of 2)

Parameter	Description
CAN clock source	PCLKB or CANMCLK
Test mode	Three test modes available for evaluation purposes: • Listen-only mode • Self-test mode 0 (external loopback) • Self-test mode 1 (internal loopback).
Module stop function	Module-stop state can be set to reduce power consumption

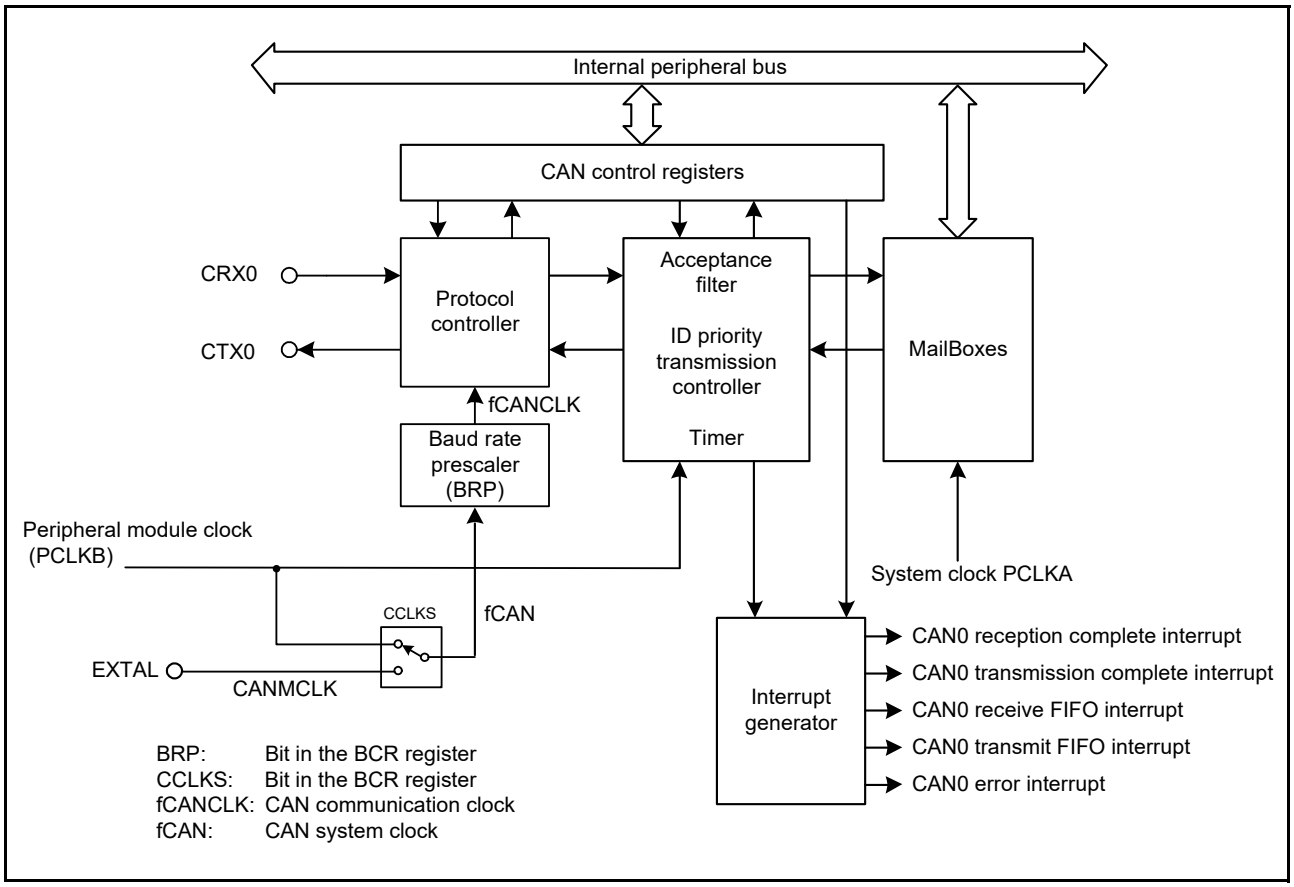


Figure 32.1 CAN module block diagram

The CAN module includes the following blocks:

- CRX0 and CTX0
  CAN input and output pins.
- Protocol controller
  Handles CAN protocol processing such as bus arbitration, bit timing during transmission and reception, stuffing, and error handling.
- Mailboxes
  32 mailboxes, which can be configured as either transmit or receive. Each mailbox has an individual ID, data length code (DLC), a data field (8 bytes), and a time stamp.
- Acceptance filter
  Filters received messages using MKR0 to MKR7 register settings.
- Timer
  Used for the time stamp function. The timer value when a message is stored in the mailbox is written as the time stamp.

- Interrupt generator
  Generates the following types of interrupts:
  - CAN0 reception complete interrupt
  - CAN0 transmission complete interrupt
  - CAN0 receive FIFO interrupt
  - CAN0 transmit FIFO interrupt
  - CAN0 error interrupt.

The CAN module communicates on the pins listed in Table 32.2. These pins are multiplexed with other signals on the MCU. For details, see section 20, I/O Ports.

Table 32.2 Pin configuration

Pin name	I/O	Function
CRX0	Input	Data receive pin
CTX0	Output	Data transmit pin

32.2 Register Descriptions

32.2.1 Control Register (CTLR)

Address(es): CAN0.CTLR 4005 0840h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RBOC	BOM[1:0]		SLPM	CANM[1:0]		TSPS[1:0]		TSRC	TPM	MLM	IDFM[1:0]		MBM
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MBM	CAN Mailbox Mode Select*1	0: Normal mailbox mode 1: FIFO mailbox mode.	R/W
b2, b1	IDFM[1:0]	ID Format Mode Select *1	b2 b1 0 0: Standard ID mode: All mailboxes, including FIFO mailboxes, handle only standard IDs 0 1: Extended ID mode: All mailboxes, including FIFO mailboxes, handle only extended IDs 1 0: Mixed ID mode: All mailboxes, including FIFO mailboxes, handle both standard and extended IDs. In normal mailbox mode, use the associated IDE bit to differentiate standard and extended IDs. In FIFO mailbox mode, the associated IDE bits are used for mailboxes 0 to 23, the IDE bits in FIDCR0 and FIDCR1 are used for the receive FIFO, and the IDE bit associated with mailbox 24 is used for the transmit FIFO. 1 1: Setting prohibited.	R/W
b3	MLM	Message Lost Mode Select*1	0: Overwrite mode 1: Overrun mode.	R/W
b4	TPM	Transmission Priority Mode Select*1	0: ID priority transmit mode 1: Mailbox number priority transmit mode	R/W
b5	TSRC	Time Stamp Counter Reset Command*4	0: Do not reset time stamp counter 1: Reset time stamp counter.*3	R/W
b7, b6	TSPS[1:0]	Time Stamp Prescaler Select*1	b7 b6 0 0: Every 1-bit time 0 1: Every 2-bit time 1 0: Every 4-bit time 1 1: Every 8-bit time.	R/W

Bit	Symbol	Bit name	Description	R/W
b9, b8	CANM[1:0]	CAN Mode of Operation Select*5	b9 b8 0 0: CAN operation mode 0 1: CAN reset mode 1 0: CAN halt mode 1 1: CAN reset mode (forced transition).	R/W
b10	SLPM	CAN Sleep Mode*5, *6	0: Exit sleep mode 1: Enter sleep mode.	R/W
b12, b11	BOM[1:0]	Bus-Off Recovery Mode*1	b12 b11 0 0: Normal Mode (ISO11898-1 specification-compliant) 0 1: Entry to CAN halt mode automatic on entering bus-off state 1 0: Entry to CAN halt mode automatic on end of bus-off state 1 1: Entry to CAN halt mode during bus-off recovery period through a software request.	R/W
b13	RBOC	Forced Return from Bus-Off*2	0: No return occurred 1: Forced return from bus-off state.*3	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Write to the BOM[1:0], TSPS[1:0], TPM, MLM, IDFM[1:0], and MBM bits in CAN reset mode.
Note 2. Set the RBOC bit to 1 in the bus-off state.
Note 3. This bit automatically clears to 0 after being set to 1. It should be read as 0.
Note 4. Set the TSRC bit to 1 in CAN operation mode.
Note 5. When the CANM[1:0] and SLPM bits are changed, check STR to ensure that the mode is switched. Do not change the CANM[1:0] bits or SLPM bit until the mode is switched.
Note 6. Write to the SLPM bit in CAN reset mode or CAN halt mode. When changing the SLPM bit, write 0 or 1 to only the SLPM bit.

MBM bit (CAN Mailbox Mode Select)

When the MBM bit is 0 (normal mailbox mode), mailboxes 0 to 31 are configured as transmit or receive mailboxes.

When the MBM bit is 1 (FIFO mailbox mode)

- Mailboxes 0 to 23 are configured as transmit or receive mailboxes
- Mailboxes 24 to 27 are configured as a transmit FIFO
- Mailboxes 28 to 31 are configured as a receive FIFO
- Transmit data is written into mailbox 24, the window mailbox for the transmit FIFO
- Receive data is read from mailbox 28, the window mailbox for the receive FIFO.

Table 32.3 lists the mailbox configuration.

IDFM[1:0] bits (ID Format Mode Select)

The IDFM[1:0] bits specify the ID format.

MLM bit (Message Lost Mode Select)

The MLM bit specifies the operation when a new message is captured in an unread mailbox. Overwrite mode or overrun mode can be selected. In both cases, the mode applies to all mailboxes, including the receive FIFO.

When the MLM bit is 0, all mailboxes are set to overwrite mode. Any new message received overwrites the pre-existing message.

When the MLM bit is 1, all mailboxes are set to overrun mode. Any new message received does not overwrite the pre-existing message, and the new message is discarded.

TPM bit (Transmission Priority Mode Select)

The TPM bit specifies the priority when transmitting messages. ID priority transmit mode or mailbox number transmit mode can be selected. All mailboxes are set for either ID priority transmission or mailbox number priority transmission.

When the TPM bit is 0, ID priority transmit mode is selected and transmission priority is arbitrated as defined in the ISO11898-1 CAN specification. In ID priority transmit mode, mailboxes 0 to 31 (in normal mailbox mode), and

mailboxes 0 to 23 (in FIFO mailbox mode), and the transmit FIFO are compared for the IDs of mailboxes configured for transmission. If two or more mailbox IDs are the same, the mailbox with the smaller number has higher priority.

Only the next message to be transmitted from the transmit FIFO is included in the transmission arbitration. If a FIFO message is currently being transmitted, the next pending message within the transmit FIFO is included in the transmission arbitration.

When the TPM bit is 1, mailbox number transmit mode is selected and the transmit mailbox with the smallest mailbox number has the highest priority. In FIFO mailbox mode, the transmit FIFO has lower priority than normal mailboxes (0 to 23).

TSRC bit (Time Stamp Counter Reset Command)

The TSRC bit resets the time stamp counter. When the TSRC bit is set to 1, the TSR register is set to 0000h. The TSRC bit is automatically set to 0.

TSPS[1:0] bits (Time Stamp Prescaler Select)

The TSPS[1:0] bits select the prescaler for the time stamp. The reference clock for the time stamp can be selected to either 1-, 2-, 4- or 8-bit time periods.

CANM[1:0] bits (CAN Mode of Operation Select)

The CANM[1:0] bits select one of the following modes for the CAN module:

- CAN operation mode
- CAN reset mode
- CAN halt mode.

The CAN sleep mode is set in the SLPM bit. For details, see section 32.3, Modes of Operation.

When the CAN module enters CAN halt mode based on the BOM[1:0] setting, the CANM[1:0] bits are automatically set to 10b.

SLPM bit (CAN Sleep Mode)

When the SLPM bit is set to 1, the CAN module enters CAN sleep mode. When the SLPM bit is set to 0, the CAN module exits CAN sleep mode. For details, see section 32.3, Modes of Operation.

BOM[1:0] bits (Bus-Off Recovery Mode)

The BOM[1:0] bits select bus-off recovery mode for the CAN module.

When the BOM[1:0] bits are 00b, the recovery from bus-off is compliant with the ISO11898-1 specification. The CAN module recovers CAN communication (error-active state) after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request is generated when recovering from bus-off.

When the BOM[1:0] bits are 01b and the CAN module reaches the bus-off state, the CANM[1:0] bits in CTLR are set to 10b to enter CAN halt mode. No bus-off recovery interrupt request is generated when recovering from bus-off, and the TECR and RECR registers are set to 00h.

When the BOM[1:0] bits are 10b, the CANM[1:0] bits are set to 10b as soon as the CAN module reaches the bus-off state. The CAN module enters CAN halt mode after a recovery from the bus-off state, that is, after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request is generated when recovering from bus-off, and the TECR and RECR registers are set to 00h.

When the BOM[1:0] bits are 11b, the CAN module enters CAN halt mode by setting the CANM[1:0] bits to 10b while the CAN module is still in the bus-off state. No bus-off recovery interrupt request is generated when recovering from bus-off and the TECR and RECR registers are set to 00h. However, a bus-off recovery interrupt request is generated if the CAN module recovers from bus-off after detecting 11 consecutive recessive bits 128 times before the CANM[1:0] bits are set to 10b.

If the CPU requests an entry to the CAN reset mode at the same time as the CAN module attempts to enter CAN halt mode (at bus-off entry when the BOM[1:0] bits are 01b, or at bus-off end when the BOM[1:0] bits are 10b), then the CPU request has higher priority.

RBOC bit (Forced Return from Bus-Off)

When the RBOC bit is set to 1 in the bus-off state, the CAN module forcibly exits the bus-off state. The RBOC bit is automatically set to 0, and the error state changes from bus-off to error-active. When the RBOC bit is set to 1, the RECR and TECR registers are set to 00h and the BOST bit in STR is set to 0, indicating no bus-off state. The other registers remain unchanged when the RBOC bit is set to 1. No bus-off recovery interrupt request is generated by this recovery from the bus-off state. Use the RBOC bit only when the BOM[1:0] bits are 00b (normal mode).

Table 32.3 Mailbox configuration

Mailbox	MBM bit = 0 (normal mailbox mode)	MBM bit = 1*1 to *5 (FIFO mailbox mode)
Mailboxes 0 to 23	Normal mailbox	Normal mailbox
Mailboxes 24 to 27		Transmit FIFO
Mailboxes 28 to 31		Receive FIFO

Note 1. The transmit FIFO is controlled by the TFCR register. The MCTL_TXj registers associated with mailboxes 24 to 27 is disabled. MCTL_TX24 to MCTL_TX27 cannot be used by the transmit FIFO.
Note 2. The receive FIFO is controlled by the RFCR register. The MCTL_RXj registers associated with mailboxes 28 to 31 are disabled.
MCTL_RX28 to MCTL_RX31 cannot be used by the receive FIFO.
Note 3. See the MIER_FIFO register description for information on FIFO interrupts.
Note 4. The MKIVLR register bits associated with mailboxes 24 to 31 are disabled. Set these bits to 0.
Note 5. The transmit and receive FIFOs can be used for both data and remote frames.

32.2.2 Bit Configuration Register (BCR)

Address(es): CAN0.BCR 4005 0844h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSEG1[3:0]				—	—	BRP[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	SJW[1:0]		—	TSEG2[2:0]			—	—	—	—	—	—	—	CCLKS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CCLKS	CAN Clock Source Selection	0: PCLKB (generated by the PLL clock) 1: CANMCLK (generated by the main clock).	R/W
b7 to b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	TSEG2[2:0]	Time Segment 2 Control	b10 b8 0 0 0: Setting prohibited 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13, b12	SJW[1:0]	Synchronization Jump Width Control	b13 b12 0 0: 1 Tq 0 1: 2 Tq 1 0: 3 Tq 1 1: 4 Tq.	R/W

Bit	Symbol	Bit name	Description	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25 to b16	BRP[9:0]	Baud Rate Prescaler select*1	These bits set the frequency of the CAN communication clock (fCANCLK)	R/W
b27, b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b28	TSEG1[3:0]	Time Segment 1 Control	b31 b28 0 0 0 0: Setting prohibited 0 0 0 1: Setting prohibited 0 0 1 0: Setting prohibited 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq.	R/W

Tq: Time Quantum

Note 1. Do not select a value less than 1 when the SCKSCR.CKSEL[2:0] bits are 011b (selecting the main clock oscillator).

For details about setting the bit timing, see section 32.4, Data Transfer Rate Configuration. Set the BCR register before entering CAN halt mode or CAN operation mode from CAN reset mode. After the setting is made once, this register can be written to in CAN reset mode or CAN halt mode. A 32-bit read/write access must be performed carefully so as not to change bits 0 to 7.

CCLKS bit (CAN Clock Source Selection)

When the CCLKS bit is 0, the peripheral module clock (PCLKB) produced by the PLL frequency synthesizer is used as the CAN clock source (fCAN). When the CCLKS bit is 1, CANMCLK produced externally by the EXTAL pins is used as the CAN clock source (fCAN).

TSEG2[2:0] bits (Time Segment 2 Control)

The TSEG2[2:0] bits specify the length of the phase buffer segment 2 (PHASE_SEG2) with a Tq value. A value from 2 to 8 Tq can be set. Set a value smaller than that of the TSEG1[3:0] bits.

SJW[1:0] bits (Synchronization Jump Width Control)

The SJW[1:0] bits specify the synchronization jump width with a Tq value. A value from 1 to 4 Tq can be set. Set a value smaller than or equal to that of the TSEG2[2:0] bits.

BRP[9:0] bits (Baud Rate Prescaler select)

The BRP[9:0] bits set the frequency of the CAN communication clock (fCANCLK). The fCANCLK cycle is 1 Tq. If the setting is P (0 to 1023), the baud rate prescaler divides fCAN by P + 1.

TSEG1[3:0] bits (Time Segment 1 Control)

The TSEG1[3:0] bits specify the total length of the propagation time segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1) with a time quantum (Tq) value. A value from 4 to 16 Tq can be set.

32.2.3 Mask Register k (MKRk) (k = 0 to 7)

Address(es): CAN0.MKR[0] 4005 0400h to CAN0.MKR[7] 4005 041Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	SID[10:0]											EID[17:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EID[17:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b17 to b0	EID[17:0]	Extended ID	0: Do not compare associated EID[17:0] bit 1: Compare associated EID[17:0] bit.	R/W
b28 to b18	SID[10:0]	Standard ID	0: Do not compare associated SID[10:0] bit 1: Compare associated SID[10:0] bit.	R/W
b31 to b29	—	Reserved	The read value is undefined. The write value should be 0.	R/W

For the mask function in FIFO mailbox mode, see section 32.6, Acceptance Filtering and Masking Functions.

Write to MKRk registers in CAN reset mode or CAN halt mode.

EID[17:0] bits (Extended ID)

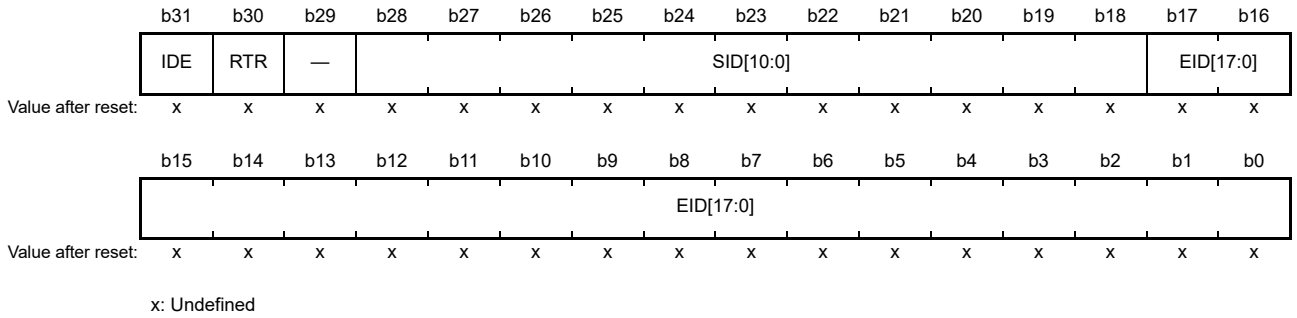
The EID[17:0] bits are the filter mask bits associated with the CAN extended ID bits. They are used to receive extended ID messages. When an EID[17:0] bit is set to 0, the respective received ID bit is not compared with the associated mailbox ID bit. When an EID[17:0] bit is set to 1, the respective received ID bit is compared with the associated mailbox ID bit.

SID[10:0] bits (Standard ID)

The SID[10:0] bits are the filter mask bits associated with the CAN standard ID bits. They are used to receive both standard ID and extended ID messages. When an SID[10:0] bit is set to 0, the respective received ID is not compared with the associated mailbox ID bit. When an SID[10:0] bit is set to 1, the respective received ID is compared with the associated mailbox ID for that bit.

32.2.4 FIFO Received ID Compare Registers 0 and 1 (FIDCR0 and FIDCR1)

Address(es): CAN0.FIDCR0 4005 0420h, CAN0.FIDCR1 4005 0424h



x: Undefined

Bit	Symbol	Bit name	Description	R/W
b17 to b0	EID[17:0]	Extended ID	Extended ID of the data and remote frames	R/W
b28 to b18	SID[10:0]	Standard ID	Standard ID of the data and remote frames	R/W
b29	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame.	R/W
b31	IDE	ID Extension*1	0: Standard ID 1: Extended ID.	R/W

Note 1. When the CTLR.IDFM[1:0] bits are any value other than 10b, the IDE bit should be written with 0 and read as 0.

The FIDCR0 and FIDCR1 registers are enabled when the MBM bit in the CTLR register is set to 1 (FIFO mailbox mode). In FIFO mailbox mode, the EID[17:0], SID[10:0], RTR, and IDE in mailbox 28 to mailbox 31 registers are disabled. Write to the FIDCR0 and FIDCR1 registers in CAN reset mode or CAN halt mode. For information on using the FIDCR0 and FIDCR1 registers, see section 32.6, Acceptance Filtering and Masking Functions.

EID[17:0] bits (Extended ID)

The EID[17:0] bits set the extended ID of data and remote frames. They are used to receive extended ID messages.

SID[10:0] bits (Standard ID)

The SID[10:0] bits set the standard ID of data and remote frames. They are used to receive both standard ID and extended ID messages.

RTR bit (Remote Transmission Request)

The RTR bit sets the frame format to data frames or remote frames:

- When the RTR bits in both the FIDCR0 and FIDCR1 registers are set to 0, only data frames are received
- When the RTR bits in both the FIDCR0 and FIDCR1 registers are set to 1, only remote frames are received
- When the RTR bits in the FIDCR0 and FIDCR1 registers are set to different values, both data frames and remote frames are received.

IDE bit (ID Extension)

The IDE bit sets the ID format to standard ID or extended ID. The IDE bit is enabled when the IDFM[1:0] bits in the CTLR register are 10b (mixed ID mode):

- When the IDE bits in both the FIDCR0 and FIDCR1 registers are set to 0, only standard ID frames are received
- When the IDE bits in both the FIDCR0 and FIDCR1 registers are set to 1, only extended ID frames are received
- When the IDE bits in the FIDCR0 and FIDCR1 registers are set to different values, both standard ID and extended ID frames are received.

32.2.5 Mask Invalid Register (MKIVLR)

Address(es): CAN0.MKIVLR 4005 0428h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MB31	MB30	MB29	MB28	MB27	MB26	MB25	MB24	MB23	MB22	MB21	MB20	MB19	MB18	MB17	MB16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	MB31 to MB0	Mask Invalid	0: Mask valid 1: Mask invalid.	R/W

Each bit in the MKIVLR register is associated with a mailbox of the same number. Bit [0] in MKIVLR corresponds to mailbox 0 (MB0), and bit [31] corresponds to mailbox 31 (MB31).

Note: Set bits [31:24] to 0 in FIFO mailbox mode.

When an MBn bit is set to 1, the acceptance mask register becomes invalid for the associated mailbox. When an MBn bit is set to 1, a message is received by the associated mailbox only if the receive message ID matches the mailbox ID exactly. Write to the MKIVLR register in CAN reset mode or CAN halt mode.

32.2.6 Mailbox Register j (MBj_ID, MBj_DL, MBj_Dm, MBj_TS) (j = 0 to 31, m = 0 to 7)

Table 32.4 lists the CAN0 mailbox memory mapping and Table 32.5 lists the CAN data frame configuration. The value of the CAN0 mailbox is undefined after reset.

Write to MBj_ID, MBj_DL, MBj_Dm and MBj_TS only when the associated MCTL_TXj or MCTL_RXj (j = 0 to 31) is 00h and the associated mailbox is not processing an abort request. See Table 32.4 for specific register addresses.

Table 32.4 CAN0 mailbox memory mapping (1 of 2)

Address	Message content
CAN0	Memory mapping
4005 0200h + 16 × j + 0	IDE, RTR, SID10 to SID6
4005 0200h + 16 × j + 1	SID5 to SID0, EID17, EID16
4005 0200h + 16 × j + 2	EID15 to EID8
4005 0200h + 16 × j + 3	EID7 to EID0
4005 0200h + 16 × j + 4	—
4005 0200h + 16 × j + 5	Data length code (DLC[3:0])
4005 0200h + 16 × j + 6	Data byte 0
4005 0200h + 16 × j + 7	Data byte 1
4005 0200h + 16 × j + 8	Data byte 2
4005 0200h + 16 × j + 9	Data byte 3
4005 0200h + 16 × j + 10	Data byte 4
4005 0200h + 16 × j + 11	Data byte 5
4005 0200h + 16 × j + 12	Data byte 6

Table 32.4 CAN0 mailbox memory mapping (2 of 2)

Address	Message content
CAN0	**Memory mapping**
4005 0200h + 16 × j + 13	Data byte 7
4005 0200h + 16 × j + 14	Time stamp upper byte
4005 0200h + 16 × j + 15	Time stamp lower byte

Table 32.5 CAN data frame configuration

SID10 to SID6	SID5 to SID0	EID17 to EID16	EID15 to EID8	EID7 to EID0	DLC3 to DLC1	DATA0	DATA1	…	DATA7

The previous value of each mailbox is saved unless a new message is received.

Address(es): CAN0.MB0_ID 4005 0200h to CAN0.MB31_ID 4005 03F0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	—	SID[10:0]											EID[17:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EID[17:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b17 to b0	EID[17:0]	Extended ID*1	Extended ID of the data and remote frames	R/W
b28 to b18	SID[10:0]	Standard ID	Standard ID of the data and remote frames	R/W
b29	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame.	R/W
b31	IDE	ID Extension*2	0: Standard ID 1: Extended ID.	R/W

Note 1. If the mailbox receives a standard ID message, the EID bits in the mailbox are undefined.
Note 2. The IDE bit is enabled when the CTLR.IDFM[1:0] bits are 10b (mixed ID mode). When the CTLR.IDFM[1:0] bits are any value other than 10b, the IDE bit should be written with 0 and read as 0.

Address(es): CAN0.MB0_DL 4005 0204h to CAN0.MB31_DL 4005 03F4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	DLC[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DLC[3:0]	Data Length Code*1	b3 b0 0 0 0 0: Data length = 0 byte 0 0 0 1: Data length = 1 byte 0 0 1 0: Data length = 2 bytes 0 0 1 1: Data length = 3 bytes 0 1 0 0: Data length = 4 bytes 0 1 0 1: Data length = 5 bytes 0 1 1 0: Data length = 6 bytes 0 1 1 1: Data length = 7 bytes 1 x x x: Data length = 8 bytes.	R/W
b15 to b4	—	Reserved	The read value is undefined. The write value should be 0.	R/W

x: Don't care

Note 1. If the mailbox receives a message with data length (set in DLC[3:0]) of n bytes, where n is less than 8, the data in the DATAn to DATA7 registers in the mailbox is undefined. Here, DATA0 to DATA7 are data registers for this mailbox. For example, if data length is 6 bytes (DLC[3:0] = 6h), the data in DATA6 and DATA7 registers is undefined.

Address(es): CAN0.MB0_D0 4005 0206h to CAN0.MB31_D0 4005 03F6h

	b7	b6	b5	b4	b3	b2	b1	b0
	DATA0							
Value after reset:	x	x	x	x	x	x	x	x

Address(es): CAN0.MB0_D1 4005 0207h to CAN0.MB31_D1 4005 03F7h

	b7	b6	b5	b4	b3	b2	b1	b0
	DATA1							
Value after reset:	x	x	x	x	x	x	x	x

Address(es): CAN0.MB0_D2 4005 0208h to CAN0.MB31_D2 4005 03F8h

	b7	b6	b5	b4	b3	b2	b1	b0
	DATA2							
Value after reset:	x	x	x	x	x	x	x	x

Address(es): CAN0.MB0_D3 4005 0209h to CAN0.MB31_D3 4005 03F9h

	b7	b6	b5	b4	b3	b2	b1	b0
	DATA3							
Value after reset:	x	x	x	x	x	x	x	x

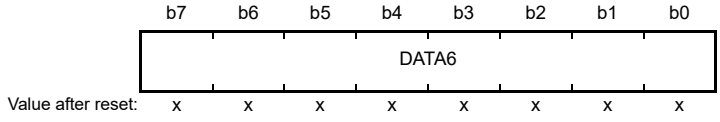
Address(es): CAN0.MB0_D4 4005 020Ah to CAN0.MB31_D4 4005 03FAh

	b7	b6	b5	b4	b3	b2	b1	b0
	DATA4							
Value after reset:	x	x	x	x	x	x	x	x

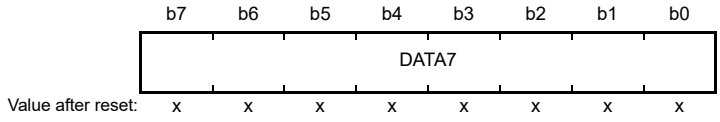
Address(es): CAN0.MB0_D5 4005 020Bh to CAN0.MB31_D5 4005 03FBh

	b7	b6	b5	b4	b3	b2	b1	b0
	DATA5							
Value after reset:	x	x	x	x	x	x	x	x

Address(es): CAN0.MB0_D6 4005 020Ch to CAN0.MB31_D6 4005 03FCh



Address(es): CAN0.MB0_D7 4005 020Dh to CAN0.MB31_D7 4005 03FDh



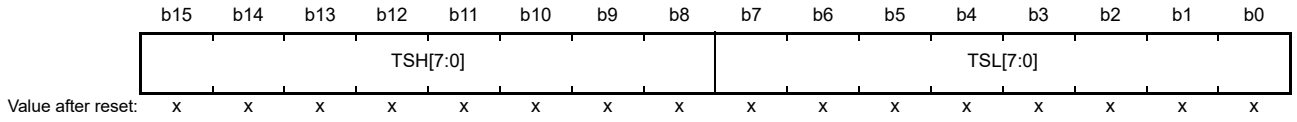
x: Undefined

Bit	Symbol	Bit name	Description	R/W
b7 to b0	DATA0 to DATA7	Data Bytes 0 to 7*1,*2	DATA0 to DATA7 store the transmitted or received CAN message data. Transmission or reception starts from DATA0. The bit order on the CAN bus is MSB-first, and transmission or reception starts from bit 7.	R/W

Note 1. If the mailbox receives a message with n bytes (where n is less than 8), the DATAn to DATA7 values in the mailbox are undefined. For example, if the received data length is 6 bytes, the values of DATA6 and DATA7 are undefined.

Note 2. If the mailbox receives a remote frame, the previous values of DATA0 to DATA7 in the mailbox are saved.

Address(es): CAN0.MB0_TS 4005 020Eh to CAN0.MB31_TS 4005 03FEh



x: Undefined

Bit	Symbol	Bit name	Description	R/W
b7 to b0	TSL[7:0]	Time Stamp Lower Byte	The TSH[7:0] and TSL[7:0] bits store the counter value of the time stamp when received messages are stored in the mailbox.	R/W
b15 to b8	TSH[7:0]	Time Stamp Higher Byte		

EID[17:0] bits (Extended ID)

The EID[17:0] bits set the extended ID of data and remote frames. They are used to transmit or receive extended ID messages.

SID[10:0] bits (Standard ID)

The SID[10:0] bits set the standard ID of data and remote frames. They are used to transmit or receive both standard ID and extended ID messages.

RTR bit (Remote Transmission Request)

The RTR bit sets the frame format to data frames or remote frames:

- The receive mailbox only receives frames with the format specified in the RTR bit
- The transmit mailbox transmits with the frame format specified in the RTR bit
- The receive FIFO mailbox receives the data frame, remote frame, or both frames specified in the RTR bit in the FIDCR0 and FIDCR1 registers
- The transmit FIFO mailbox transmits the data frame or remote frame, as specified in the RTR bit in the transmit message.

IDE bit (ID Extension)

The IDE bit sets the ID format to standard ID or extended ID. The IDE bit is enabled when the IDFM[1:0] bits in the CTLR register is 10b (mixed ID mode):

- The receive mailbox only receives the ID format specified in the IDE bit
- The transmit mailbox transmits with the ID format specified in the IDE bit
- The receive FIFO mailbox receives messages with the standard ID and extended ID settings specified in the IDE bits in the FIDCR0 and FIDCR1 registers
- The transmit FIFO mailbox transmits messages with the standard ID or extended ID setting specified in the IDE bit in the transmit message.

DLC[3:0] bits (Data Length Code)

The DLC[3:0] bits specify the data length to be transmitted in data frames. When a remote frame is used to request data, the DLC[3:0] bits specifies the requested data length.

When a data frame is received, the received data length is stored in DLC[3:0]. When a remote frame is received, the DLC[3:0] bits store the requested data length.

32.2.7 Mailbox Interrupt Enable Register (MIER)

Address(es): CAN0.MIER 4005 042Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MB31	MB30	MB29	MB28	MB27	MB26	MB25	MB24	MB23	MB22	MB21	MB20	MB19	MB18	MB17	MB16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b31 to b0	MB31 to MB0	Interrupt Enable	0: Disable interrupt 1: Enable interrupt. Bit [31] is associated with mailbox 31 (MB31) and bit [0] with mailbox 0 (MB0).	R/W

The MIER register allows independent enabling of interrupts for each mailbox. This register is available in normal mailbox mode. Do not access this register in FIFO mailbox mode.

Each bit is associated with the mailbox having the same number. These bits enable or disable transmission and reception complete interrupts for the associated mailboxes:

- Bit [0] in MIER is associated with mailbox 0 (MB0)
- Bit [31] in MIER is associated with mailbox 31 (MB31).

Write to MIER only when the related MCTL_TXj or MCTL_RXj (j = 0 to 31) is 00h and the associated mailbox is not processing a transmission or reception abort request.

32.2.8 Mailbox Interrupt Enable Register for FIFO Mailbox Mode (MIER_FIFO)

Address(es): CAN0.MIER_FIFO 4005 042Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	MB29	MB28	—	—	MB25	MB24	MB23	MB22	MB21	MB20	MB19	MB18	MB17	MB16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b23 to b0	MB23 to MB0	Interrupt Enable	0: Disable interrupt 1: Enable interrupt. Bit [23] is associated with mailbox 23 (MB23) and bit [0] with mailbox 0 (MB0).	R/W
b24	MB24	Transmit FIFO Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b25	MB25	Transmit FIFO Interrupt Generation Timing Control	0: Generated every time transmission completes 1: Generated when the transmit FIFO empties on transmission completion.	R/W
b27, b26	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b28	MB28	Receive FIFO Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b29	MB29	Receive FIFO Interrupt Generation Timing Control*1	0: Generated every time reception completes 1: Generated when the receive FIFO becomes buffer warning*2 by reception completion.	R/W
b31, b30	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Note 1. No interrupt request is generated when the receive FIFO becomes a buffer warning from full.
Note 2. "Buffer warning" indicates a state in which the third message is stored in the receive FIFO.

The MIER_FIFO register allows independent enabling of interrupts for each mailbox and FIFO. This register is available in FIFO mailbox mode. Do not access this register in normal mailbox mode.

The MB0 to MB23 bits are associated with the mailbox having the same number. These bits enable or disable transmission and reception complete interrupts for the associated mailboxes:

- Bit [0] is associated with mailbox 0 (MB0)
- Bit [23] is associated with mailbox 23 (MB23).

The MB24, MB25, MB28 and MB29 bits specify whether transmit and receive FIFO interrupts are enabled, and the timing of interrupt requests.

Write to the MIER_FIFO register only when the relevant MCTL_TXj or MCTL_RXj (j = 0 to 31) is 00h and the associated mailbox does not process a transmission or reception abort request. In addition, change the MIER_FIFO bits for the relevant FIFO only when all the following conditions are true:

- The TFCR.TFE bit is 0 and the TFCR.TFEST bit is 1
- The RFCR.RFE bit is 0 and the RFCR.RFEST bit is 1.

32.2.9 Message Control Register for Transmit (MCTL_TXj) (j = 0 to 31)

- Transmit mode (when the TRMREQ bit is 1 and the RECREQ bit is 0)

Address(es): CAN0.MCTL_TX[0] 4005 0820h to CAN0.MCTL_TX[31] 4005 083Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	TRMREQ	RECREQ	—	ONESHOT	—	TRMABT	TRMACTIVE	SENTDATA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SENTDATA	Transmission Complete Flag*1,*2	0: Transmission not complete 1: Transmission complete.	R/W
b1	TRMACTIVE	Transmission-in-Progress Status Flag	0: Transmission pending or not requested 1: Transmission in progress.	R
b2	TRMABT	Transmission Abort Complete Flag*1,*2	0: Transmission started, transmission abort failed because transmission completed, or transmission abort not requested 1: Transmission abort complete.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	ONESHOT	One-Shot Enable*2,*3	0: Disable one-shot transmission 1: Enable one-shot transmission.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	RECREQ	Receive Mailbox Request *2,*3,*4,*5	0: Do not configure for reception 1: Configure for reception.	R/W
b7	TRMREQ	Transmit Mailbox Request *2,*4	0: Do not configure for transmission 1: Configure for transmission.	R/W

Note 1. Write 0 only. Writing 1 has no effect.
Note 2. When writing to bits of this register, write 1 to the SENTDATA and TRMABT bits if they are not the write target.
Note 3. To enter one-shot transmit mode, write 1 to the ONESHOT bit at the same time as setting the TRMREQ bit to 1. To exit one-shot transmit mode, write 0 to the ONESHOT bit after the message is transmitted or aborted.
Note 4. Do not set both the RECREQ and TRMREQ bits to 1.
Note 5. When setting the RECREQ bit to 0, set bits SENDDATA, TRMACTIVE, and TRMABT to 0 simultaneously.

The MCTL_TXj register sets mailbox j to transmit or receive mode. In transmit mode, MCTL_TXj also controls and indicates the status of transmission. Do not access the MCTL_TXj register if mailbox j is in receive mode. Only write to MCTL_TXj in CAN operation mode or CAN halt mode. Do not use the MCTL_TX24 to MCTL_TX31 registers in FIFO mailbox mode.

SENTDATA flag (Transmission Complete Flag)

The SENTDATA flag is set to 1 when data transmission from the associated mailbox is complete. The SENTDATA flag is set to 0 through a software write.

To set the SENTDATA flag to 0, first set the TRMREQ bit to 0. The SENTDATA and TRMREQ bits cannot be set to 0 simultaneously. To transmit a new message from the associated mailbox, set the SENTDATA flag to 0.

TRMACTIVE flag (Transmission-in-Progress Status Flag)

The TRMACTIVE flag is set to 1 when the associated mailbox of the CAN module begins to transmit a message. The TRMACTIVE flag is set to 0 when the CAN module lose CAN bus arbitration, a CAN bus error occurs, or data transmission is complete.

TRMABT flag (Transmission Abort Complete Flag)

The TRMABT flag is set to 1 in the following cases:

- Following a transmission abort request, when the transmission abort completes before starting transmission
- Following a transmission abort request, when the CAN module detects CAN bus arbitration-lost or CAN bus error
- In one-shot transmission mode (RECREQ = 0, TRMREQ = 1, and ONESHOT = 1), when the CAN module detects CAN bus arbitration-lost or CAN bus error.

The TRMABT flag is not set to 1 when data transmission is complete. The TRMABT flag is set to 0 through a software write.

ONESHOT bit (One-Shot Enable)

When the ONESHOT bit is set to 1 in transmit mode (RECREQ = 0 and TRMREQ = 1), the CAN module transmits a message only one time. The CAN module does not transmit the message again if a CAN bus error or CAN bus arbitration-lost occurs. When transmission is complete, the SENTDATA flag is set to 1. If transmission is not complete because of a CAN bus error or CAN bus arbitration-lost, the TRMABT flag is set to 1. Set the ONESHOT bit to 0 after the SENTDATA or TRMABT bit is set to 1.

RECREQ bit (Receive Mailbox Request)

When the RECREQ bit is set to 1, the associated mailbox is configured for reception of a data or a remote frame.

When the RECREQ bit is set to 0, the associated mailbox is not configured for reception of a data or a remote frame.

Due to hardware protection, the RECREQ bit cannot be set to 0 through a software write during the following period:

- Hardware protection is started from acceptance filter processing (the beginning of the CRC field)
- Hardware protection is released:
  - For the mailbox that is specified to receive the incoming message, after the received data is stored in the mailbox or a CAN bus error occurs. This means that the maximum period of hardware protection is from the beginning of the CRC field to the end of the 7th bit of EOF.
  - For the other mailboxes, after the acceptance filter processing.
  - If no mailbox is specified to receive the message, after acceptance filter processing.

When setting the RECREQ bit to 1, do not set the TRMREQ bit to 1. To change the configuration of a mailbox from transmission to reception, first abort the transmission, then set the SENTDATA and TRMABT bits to 0 before changing to reception.

Note: MCTL_TXj.RECREQ is the mirror bit of MCTL_RXj.RECREQ.

TRMREQ bit (Transmit Mailbox Request)

When the TRMREQ bit is set to 1, the associated mailbox is configured for transmission of a data frame or a remote frame.

When the TRMREQ bit is set to 0, the associated mailbox is not configured for transmission of a data frame or a remote frame.

If the TRMREQ bit is changed from 1 to 0 to cancel the associated transmission request, either the TRMABT or SENTDATA flag is set to 1. When setting the TRMREQ bit to 1, do not set the RECREQ bit to 1. To change the configuration of a mailbox from reception to transmission, first abort the reception, then set the NEWDATA and MSGLOST bits to 0 before changing to transmission.

Note: MCTL_TXj.TRMREQ is the mirror bit of MCTL_RXj.TRMREQ.

32.2.10 Message Control Register for Receive (MCTL_RXj) (j = 0 to 31)

- Receive mode (when the TRMREQ bit is 0 and the RECREQ bit is 1)

Address(es): CAN0.MCTL_RX[0] 4005 0820h to CAN0.MCTL_RX[31] 4005 083Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	TRMREQ	RECREQ	—	ONESHOT	—	MSGLOST	INVALDATA	NEWDATA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	NEWDATA	Reception Complete Flag*1,*2	0: No data received, or 0 was written to the bit 1: New message is being stored or was stored to the mailbox.	R/W
b1	INVALDATA	Reception-in-Progress Status Flag	0: Message valid 1: Message updated.	R
b2	MSGLOST	Message Lost Flag*1,*2	0: Message not overwritten or overrun 1: Message overwritten or overrun.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	ONESHOT	One-Shot Enable*2,*3	0: Disable one-shot reception 1: Enable one-shot reception.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	RECREQ	Receive Mailbox Request *2,*3,*4,*5	0: Do not configure for reception 1: Configure for reception.	R/W
b7	TRMREQ	Transmit Mailbox Request *2,*4	0: Do not configure for transmission 1: Configure for transmission.	R/W

Note 1. Write 0 only. Writing 1 has no effect.
Note 2. When writing to bits of this register, write 1 to the NEWDATA and MSGLOST bits if they are not the write target.
Note 3. To enter one-shot receive mode, write 1 to the ONESHOT bit at the same time as setting the RECREQ bit to 1. To exit one-shot receive mode, write 0 to the ONESHOT bit after writing 0 to the RECREQ bit and confirming that it is set to 0.
Note 4. Do not set both the RECREQ and TRMREQ bits to 1.
Note 5. When setting the RECREQ bit to 0, set bits MSGLOST, NEWDATA, and RECREQ to 0 simultaneously.

The MCTL_RXj register sets mailbox j to transmit mode or receive mode. In receive mode, MCTL_RXj also controls and indicates the status of reception.

Do not access the MCTL_RXj register if mailbox j is in transmit mode. Only write to MCTL_RXj in CAN operation mode or CAN halt mode. Do not use the MCTL_RX24 to MCTL_RX31 registers in FIFO mailbox mode.

NEWDATA flag (Reception Complete Flag)

The NEWDATA flag is set to 1 when a new message is being stored or was stored in the mailbox. The timing for setting NEWDATA to 1 is simultaneous with the INVALDATA flag. The NEWDATA flag is set to 0 through a software write. The NEWDATA flag cannot be set to 0 through a software write when the associated INVALDATA flag is 1.

INVALDATA flag (Reception-in-Progress Status Flag)

After the completion of a message reception, the INVALDATA flag is set to 1 while the received message is updated in the associated mailbox. The INVALDATA flag is set to 0 immediately after the message is stored. If the mailbox is read when the INVALDATA flag is 1, the data is undefined.

MSGLOST flag (Message Lost Flag)

The MSGLOST flag is set to 1 when the mailbox is overwritten or overrun by a new received message when the NEWDATA flag is 1. The MSGLOST flag is set to 1 at the end of the 6th bit of EOF. The MSGLOST flag is set to 0 through a software write.

In both overwrite and overrun modes, the MSGLOST flag cannot be set to 0 by writing 0 with software during the five peripheral module clock (PCLKB) cycles following the sixth bit of EOF.

ONESHOT bit (One-Shot Enable)

When the ONESHOT bit is set to 1 in receive mode (RECREQ = 1 and TRMREQ = 0), the mailbox receives a message only one time. The mailbox does not behave as a receive mailbox after it receives the message. The behavior of the NEWDATA and INVALDATA flags is the same as in normal receive mode. In one-shot receive mode, the MSGLOST flag is not set to 1. To set the ONESHOT bit to 0, first write 0 to the RECREQ bit and ensure that it is 0.

RECREQ bit (Receive Mailbox Request)

When the RECREQ bit is set to 1, the associated mailbox is configured for reception of a data frame or a remote frame.

When the RECREQ bit is set to 0, the associated mailbox is not configured for reception of a data frame or remote frame.

Due to hardware protection, the RECREQ bit cannot be set to 0 through a software write during the following period:

- Hardware protection is started from acceptance filter processing (the beginning of the CRC field)
- Hardware protection is released:
  - For the mailbox that is specified to receive the incoming message, after the received data is stored into the mailbox or a CAN bus error occurs. The maximum period of hardware protection is from the beginning of the CRC field to the end of the 7th bit of EOF.
  - For the other mailboxes, after acceptance filter processing.
  - If no mailbox is specified to receive the message, after acceptance filter processing.

When setting the RECREQ bit to 1, do not set the TRMREQ bit to 1. To change the configuration of a mailbox from transmission to reception, first abort the transmission, then set the SENTDATA and TRMABT bits to 0 before changing to reception.

Note: MCTL_RXj.RECREQ is the mirror bit of MCTL_TXj.RECREQ.

TRMREQ bit (Transmit Mailbox Request)

When the TRMREQ bit is set to 1, the associated mailbox is configured for transmission of a data frame or a remote frame.

When the TRMREQ bit is set to 0, the associated mailbox is not configured for transmission of a data frame or remote frame.

If the TRMREQ bit changes from 1 to 0 to cancel the associated transmission request, either the TRMABT or SENTDATA flag is set to 1. When setting the TRMREQ bit to 1, do not set the RECREQ bit to 1. To change the configuration of a mailbox from reception to transmission, first abort the reception, then set the NEWDATA and MSGLOST bits to 0 before changing to transmission.

Note: MCTL_RXj.TRMREQ is the mirror bit of MCTL_TXj.TRMREQ.

32.2.11 Receive FIFO Control Register (RFCR)

Address(es): CAN0.RFCR 4005 0848h

	b7	b6	b5	b4	b3	b2	b1	b0
	RFEST	RFWST	RFFST	RFMLF	RFUST[2:0]			RFE
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RFE	Receive FIFO Enable	0: Disable receive FIFO 1: Enable receive FIFO.	R/W
b3 to b1	RFUST[2:0]	Receive FIFO Unread Message Number Status	b3 b1 0 0 0: No unread message 0 0 1: 1 unread message 0 1 0: 2 unread messages 0 1 1: 3 unread messages 1 0 0: 4 unread messages 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved.	R
b4	RFMLF	Receive FIFO Message Lost Flag	0: Receive FIFO message not lost 1: Receive FIFO message lost.	R/W
b5	RFFST	Receive FIFO Full Status Flag	0: Receive FIFO not full 1: Receive FIFO full (4 unread messages).	R
b6	RFWST	Receive FIFO Buffer Warning Status Flag	0: Receive FIFO is not buffer warning 1: Receive FIFO is buffer warning (3 unread messages).	R
b7	RFEST	Receive FIFO Empty Status Flag	0: Unread message in receive FIFO 1: No unread message in receive FIFO.	R

Write to the RFCR register in CAN operation mode or CAN halt mode.

RFE bit (Receive FIFO Enable)

When the RFE bit is set to 1, the receive FIFO is enabled.

When the RFE bit is set to 0, the receive FIFO is disabled for reception and becomes empty (RFEST = 1). Write 0 to the RFE bit simultaneously with the RFMLF bit setting.

Do not set the RFE bit to 1 in normal mailbox mode (CTLR.MBM = 0).

Due to hardware protection, the RFE bit cannot be set to 0 through a software write during the following period:

- Hardware protection is started from acceptance filter processing (the beginning of the CRC field)
- Hardware protection is released:
  - If the receive FIFO is specified to receive the incoming message, after the received data is stored into the receive FIFO or a CAN bus error occurs. The maximum period of hardware protection is from the beginning of the CRC field to the end of 7th bit of EOF.
  - If the receive FIFO is not specified to receive the message, after the acceptance filter processing.

RFUST[2:0] bits (Receive FIFO Unread Message Number Status)

The RFUST[2:0] bits indicate the number of unread messages in the receive FIFO. The value of the RFUST[2:0] bits is initialized to 000b when the RFE bit is set to 0.

RFMLF flag (Receive FIFO Message Lost Flag)

The RFMLF flag is set to 1 (receive FIFO message lost) when the receive FIFO receives a new message and is full. It is set to 1 at the end of the 6th bit of EOF.

The RFMLF flag is set to 0 through a software write (writing 1 has no effect). In both overwrite and overrun modes, if the receive FIFO is full and determined to have received a message, the RFMLF flag cannot be set to 0 (receive FIFO message was not lost) through a software write during 5 PCLKB cycles following the 6th bit of EOF due to hardware protection.

RFFST flag (Receive FIFO Full Status Flag)

The RFFST flag is set to 1 (receive FIFO is full) when the number of unread messages in the receive FIFO is 4. The RFFST flag is 0 (receive FIFO is not full) when the number of unread messages in the receive FIFO is less than 4. The RFFST flag is set to 0 when the RFE bit is 0.

RFWST flag (Receive FIFO Buffer Warning Status Flag)

The RFWST flag is set to 1 (receive FIFO is buffer warning) when the number of unread messages in the receive FIFO is 3. The RFWST flag is 0 (receive FIFO is not buffer warning) when the number of unread messages in the receive FIFO is less than 3 or equal to 4. The RFWST flag is set to 0 when the RFE bit is 0.

RFEST flag (Receive FIFO Empty Status Flag)

The RFEST flag is set to 1 (no unread message in receive FIFO) when the number of unread messages in the receive FIFO is 0. The RFEST flag is set to 1 when the RFE bit is set to 0. The RFEST flag is set to 0 (unread message in receive FIFO) when the number of unread messages in the receive FIFO is one or more.

Figure 32.2 shows the receive FIFO mailbox operation.

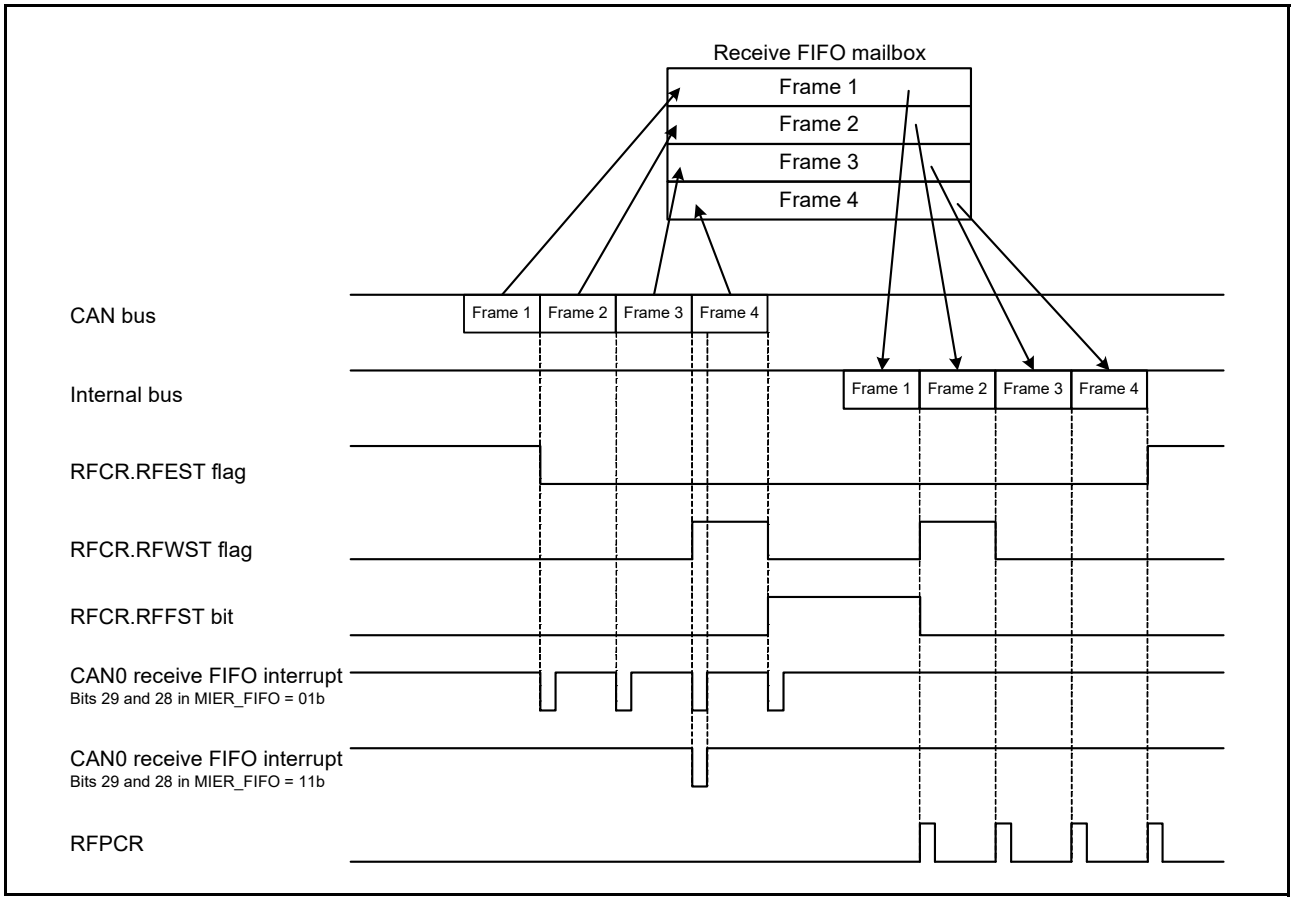


Figure 32.2 Receive FIFO mailbox operation when bits [29] and [28] in MIER_FIFO = 01b or 11b

32.2.12 Receive FIFO Pointer Control Register (RFPCR)

Address(es): CAN0.RFPCR 4005 0849h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Description	R/W
b7 to b0	The CPU pointer for the receive FIFO is incremented by writing FFh to RFPCR	W

When the receive FIFO is not empty, write FFh to the RFPCR register through software to increment the CPU pointer to the next mailbox location. Do not write to RFPCR when the RFCR.RFE bit is 0 (receive FIFO disabled).

Both the CAN and CPU pointers increment when a new message is received and the RFFST bit is 1 (receive FIFO is full) in overwrite mode. When the RFMLF bit is 1 in this state, the CPU pointer does not increment on a software write to the RFPCR register.

32.2.13 Transmit FIFO Control Register (TFCR)

Address(es): CAN0.TFCR 4005 084Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	TFEST	TFFST	—	—	TFUST[2:0]			TFE
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TFE	Transmit FIFO Enable	0: Disable transmit FIFO 1: Enable transmit FIFO.	R/W
b3 to b1	TFUST[2:0]	Transmit FIFO Unsent Message Number Status	b3 b1 0 0 0: 0 unsent messages 0 0 1: 1 unsent message 0 1 0: 2 unsent messages 0 1 1: 3 unsent messages 1 0 0: 4 unsent messages 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved.	R
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	TFFST	Transmit FIFO Full Status	0: Transmit FIFO not full 1: Transmit FIFO full (4 unsent messages).	R
b7	TFEST	Transmit FIFO Empty Status	0: Unsent message in transmit FIFO 1: No unsent message in transmit FIFO.	R

Write to the TFCR register in CAN operation mode or CAN halt mode.

TFE bit (Transmit FIFO Enable)

Setting the TFE bit to 1 enables the transmit FIFO. Setting the TFE bit to 0 empties the transmit FIFO (TFEST bit = 1), and unsent messages from the transmit FIFO are lost in the following ways:

- Immediately if a message from the transmit FIFO is not scheduled for the next transmission or already in transmission
- On completion of transmission, a CAN bus error, CAN bus arbitration-lost, or entry to CAN halt mode if a message from the transmit FIFO is scheduled for the next transmission or already in transmission.

Before setting the TFE bit to 1 again, ensure that the TFEST bit is set to 1. After setting the TFE bit to 1, write transmit data to mailbox 24.

Do not set the TFE bit to 1 in normal mailbox mode (CTLR.MBM bit = 0).

TFUST[2:0] bits (Transmit FIFO Unsent Message Number Status)

The TFUST[2:0] bits indicate the number of unsent messages in the transmit FIFO. The TFUST[2:0] bits are set to 000b after TFE bit is set to 0 and transmission aborts or completes.

TFFST bit (Transmit FIFO Full Status)

The TFFST bit is set to 1 (transmit FIFO is full) when the number of unsent messages in the transmit FIFO is 4. The TFFST bit is set to 0 (transmit FIFO is not full) when the number of unsent messages in the transmit FIFO is less than 4. The TFFST bit is set to 0 when transmission from the transmit FIFO is aborted.

TFEST bit (Transmit FIFO Empty Status)

The TFEST bit is set to 1 (no message in transmit FIFO) when the number of unsent messages in the transmit FIFO is 0. The TFEST bit is set to 1 when transmission from the transmit FIFO is aborted. The TFEST bit is set to 0 (message in transmit FIFO) when the number of unsent messages in the transmit FIFO is not 0.

Figure 32.3 shows the transmit FIFO mailbox operation.

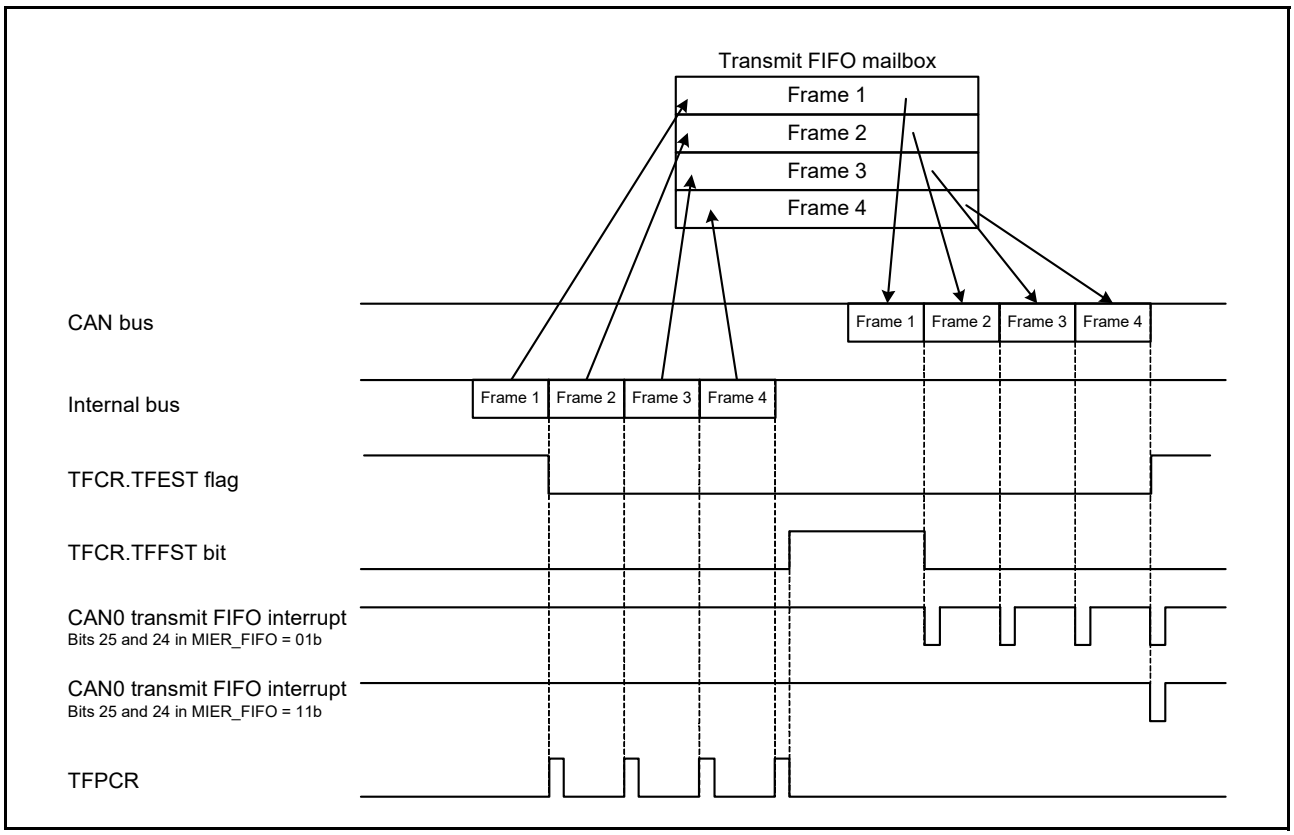


Figure 32.3 Transmit FIFO mailbox operation when bits [25] and [24] in MIER_FIFO = 01b or 11b

32.2.14 Transmit FIFO Pointer Control Register (TFPCR)

Address(es): CAN0.TFPCR 4005 084Bh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Description	R/W
b7 to b0	The CPU pointer for the transmit FIFO is incremented by writing FFh to TFPCR	W

When the transmit FIFO is not full, write FFh to the TFPCR register through software to increment the CPU pointer for the transmit FIFO to the next mailbox location.

Do not write to the TFPCR register when the TFCR.TFE bit is 0 (transmit FIFO disabled).

32.2.15 Status Register (STR)

Address(es): CAN0.STR 4005 0842h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	RECST	TRMST	BOST	EPST	SLPST	HLTST	RSTST	EST	TABST	FMLST	NMLST	TFST	RFST	SDST	NDST
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	NDST	NEWDATA Status Flag	0: No mailbox with NEWDATA bit = 1 1: 1 or more mailboxes with NEWDATA bit = 1.	R
b1	SDST	SENTDATA Status Flag	0: No mailbox with SENTDATA bit = 1 1: 1 or more mailboxes with SENTDATA bit = 1.	R
b2	RFST	Receive FIFO Status Flag	0: Receive FIFO empty 1: Message in receive FIFO.	R
b3	TFST	Transmit FIFO Status Flag	0: Transmit FIFO full 1: Transmit FIFO not full.	R
b4	NMLST	Normal Mailbox Message Lost Status Flag	0: No mailbox with MSGLOST bit = 1 1: 1 or more mailboxes with MSGLOST bit = 1.	R
b5	FMLST	FIFO Mailbox Message Lost Status Flag	0: RFMLF bit = 0 1: RFMLF bit = 1.	R
b6	TABST	Transmission Abort Status Flag	0: No mailbox with TRMABT bit = 1 1: 1 or more mailboxes with TRMABT bit = 1.	R
b7	EST	Error Status Flag	0: No error occurred 1: Error occurred.	R
b8	RSTST	CAN Reset Status Flag	0: Not in CAN reset mode 1: In CAN reset mode.	R
b9	HLTST	CAN Halt Status Flag	0: Not in CAN halt mode 1: In CAN halt mode.	R
b10	SLPST	CAN Sleep Status Flag	0: Not in CAN sleep mode 1: In CAN sleep mode.	R
b11	EPST	Error-Passive Status Flag	0: Not in error-passive state 1: In error-passive state.	R
b12	BOST	Bus-Off Status Flag	0: Not in bus-off state 1: In bus-off state.	R

Bit	Symbol	Bit name	Description	R/W
b13	TRMST	Transmit Status Flag	0: Bus idle or reception in progress 1: Transmission in progress or in bus-off state.	R
b14	RECST	Receive Status Flag	0: Bus idle or transmission in progress 1: Reception in progress.	R
b15	—	Reserved	The read value is 0.	R

NDST flag (NEWDATA Status Flag)

The NDST flag is set to 1 when at least one NEWDATA bit in MCTL_RXj (j = 0 to 31) is 1, regardless of the value of MIER or MIER_FIFO. The NDST flag is set to 0 when all NEWDATA bits are 0.

SDST flag (SENTDATA Status Flag)

The SDST flag is set to 1 when at least one SENTDATA bit in MCTL_TXj (j = 0 to 31) is 1, regardless of the value of MIER or MIER_FIFO. The SDST flag is set to 0 when all SENTDATA bits are 0.

RFST flag (Receive FIFO Status Flag)

The RFST flag is set to 1 when the receive FIFO is not empty. The RFST flag is set to 0 when the receive FIFO is empty or normal mailbox mode is selected.

TFST flag (Transmit FIFO Status Flag)

The TFST flag is set to 1 when the transmit FIFO is not full. The TFST flag is set to 0 when the transmit FIFO is full or normal mailbox mode is selected.

NMLST flag (Normal Mailbox Message Lost Status Flag)

The NMLST flag is set to 1 when at least one MSGLOST bit in MCTL_RXj (j = 0 to 31) is 1, regardless of the value of MIER or MIER_FIFO. The NMLST flag is set to 0 when all MSGLOST bits are 0.

FMLST flag (FIFO Mailbox Message Lost Status Flag)

The FMLST flag is set to 1 when the RFMLF bit in the RFCR register is 1, regardless of the value of MIER_FIFO. The FMLST bit is set to 0 when the RFMLF bit is 0.

TABST flag (Transmission Abort Status Flag)

The TABST flag is set to 1 when at least one TRMABT bit in MCTL_TXj (j = 0 to 31) is 1, regardless of the value of MIER or MIER_FIFO. The TABST flag is set to 0 when all TRMABT bits are 0.

EST flag (Error Status Flag)

The EST flag is set to 1 when at least one error is detected by the EIFR register, regardless of the value of EIER. The EST flag is set to 0 when no error is detected by the EIFR register.

RSTST flag (CAN Reset Status Flag)

The RSTST flag is set to 1 when the CAN module is in CAN reset mode. The RSTST flag is 0 when the CAN module is not in CAN reset mode. When the CAN module transition from CAN reset mode to CAN sleep mode, the RSTST flag remains 1.

HLTST flag (CAN Halt Status Flag)

The HLTST flag is set to 1 when the CAN module is in CAN halt mode. The HLTST flag is set to 0 when the CAN module is not in CAN halt mode. When the CAN module transition from CAN halt mode to CAN sleep mode, the HLTST flag remains 1.

SLPST flag (CAN Sleep Status Flag)

The SLPST flag is set to 1 when the CAN module is in CAN sleep mode. The SLPST flag is set to 0 when the CAN module is not in CAN sleep mode.

EPST flag (Error-Passive Status Flag)

The EPST flag is set to 1 when the value of the TECR or RECR register exceeds 127 and the CAN module is in the error-passive state (128 ≤ TEC < 256 or 128 ≤ REC < 256). The EPST flag is set to 0 when the CAN module is not in the error-passive state.

BOST flag (Bus-Off Status Flag)

The BOST flag is set to 1 when the value of the TECR register exceeds 255 and the CAN module is in the bus-off state (TEC ≥ 256). The BOST flag is set to 0 when the CAN module is not in the bus-off state.

TRMST flag (Transmit Status Flag)

The TRMST flag is set to 1 when the CAN module performs as a transmitter node or is in the bus-off state. The TRMST flag is set to 0 when the CAN module performs as a receiver node or is in the bus-idle state.

RECST flag (Receive Status Flag)

The RECST flag is set to 1 when the CAN module performs as a receiver node. The RECST flag is set to 0 when the CAN module performs as a transmitter node or is in the bus-idle state.

32.2.16 Mailbox Search Mode Register (MSMR)

Address(es): CAN0.MSMR 4005 0853h

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

Bit	Symbol	Bit name	Description	R/W
b1, b0	MBSM[1:0]	Mailbox Search Mode Select	b1 b0 0 0: Receive mailbox search mode 0 1: Transmit mailbox search mode 1 0: Message lost search mode 1 1: Channel search mode.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Write to the MSMR register in CAN operation mode or CAN halt mode.

MBSM[1:0] bits (Mailbox Search Mode Select)

The MBSM[1:0] bits select the search mode for the mailbox search function.

When the MBSM[1:0] bits are 00b, receive mailbox search mode is selected. In this mode, the search targets are the NEWDATA bit in MCTL_RXj (j = 0 to 31) for the normal mailbox and the RFCR.RFEST bit for the receive FIFO.

When the MBSM[1:0] bits are 01b, transmit mailbox search mode is selected. In this mode, the search target is the SENTDATA bit in MCTL_TXj.

When the MBSM[1:0] bits are 10b, message lost search mode is selected. In this mode, the search targets are the MSGLOST bit in MCTL_RXj for the normal mailbox and the RFCR.RFMLF bit for the receive FIFO.

When the MBSM[1:0] bits are 11b, channel search mode is selected. In this mode, the search target is CSSR. See section 32.2.18, Channel Search Support Register (CSSR).

32.2.17 Mailbox Search Status Register (MSSR)

Address(es): CAN0.MSSR 4005 0852h

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

Bit	Symbol	Bit name	Description	R/W
b4 to b0	MBNST[4:0]	Search Result Mailbox Number Status	These bits output the smallest mailbox number that is found in each search mode selected in the MSMR register.	R
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SEST	Search Result Status	0: Search result found 1: No search result.	R

MBNST[4:0] bits (Search Result Mailbox Number Status)

In all mailbox search modes, the MBNST[4:0] bits output the smallest found mailbox number. In receive mailbox search mode, transmit mailbox search mode, and message lost search mode, the value of the mailbox (search result to be output) is updated under the following conditions:

- When the respective NEWDATA, SENTDATA or MSGLOST bit is set to 0 for the mailbox output by MBNST[4:0]
- When the respective NEWDATA, SENTDATA or MSGLOST bit is set to 1 for a mailbox with a smaller number than that in MBNST[4:0].

If the MBSM[1:0] bits are set to 00b (receive mailbox search mode) or 10b (message lost search mode), the receive FIFO (mailbox 28) is output when it is not empty and there are no unread received messages or no lost messages in any of the normal mailboxes (0 to 23). If the MBSM[1:0] bits are set to 01b (transmit mailbox search mode), the transmit FIFO (mailbox 24) is not output. Table 32.6 lists the behavior of the MBNST[4:0] bits in FIFO mailbox mode.

In channel search mode, the MBNST[4:0] bits output the associated channel number. After the MSSR register is read by software, the next target channel number is output.

SEST bit (Search Result Status)

The SEST bit is set to 1 (no search result) when no associated mailbox is found after searching all mailboxes. For example, in transmit mailbox search mode, the SEST bit is set to 1 when no SENTDATA bit for any mailbox is 1. The SEST bit is set to 0 when at least one SENTDATA bit is 1. When the SEST bit is 1, the value of the MBNST[4:0] bits is undefined.

Table 32.6 Behavior of MBNST[4:0] bits in FIFO mailbox mode

MBSM[1:0] bits	Mailbox 24 (transmit FIFO)	Mailbox 28 (receive FIFO)
00b	Mailbox 24 is not output	Mailbox 28 is output when no MCTL_RXj.NEWDATA bit for the normal mailboxes is set to 1 (new message is being stored or was stored to the mailbox) and the receive FIFO is not empty
01b		Mailbox 28 is not output
10b		Mailbox 28 is output when no MCTL_RXj.MSGLOST bit for the normal mailboxes is set to 1 (message is overwritten or overrun) and the RFCR.RFMLF bit is set to 1 (receive FIFO message was lost) in the receive FIFO
11b		Mailbox 28 is not output

32.2.18 Channel Search Support Register (CSSR)

Address(es): CAN0.CSSR 4005 0851h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Description	R/W
b7 to b0	When the value for the channel search is input, the channel number is output to the MSSR register	R/W

The bits that are set to 1 in the CSSR register are encoded by an 8/3 encoder (the LSB position has the higher priority) and output to the MBNST[4:0] bits in the MSSR register. The MSSR register outputs the updated value whenever the MSSR register is read by software.

Write to the CSSR register only when the MSMR.MBSM[1:0] bits are 11b (channel search mode). Write to the CSSR register in CAN operation mode or CAN halt mode.

Figure 32.4 shows writes to and reads from the CSSR and MSSR registers.

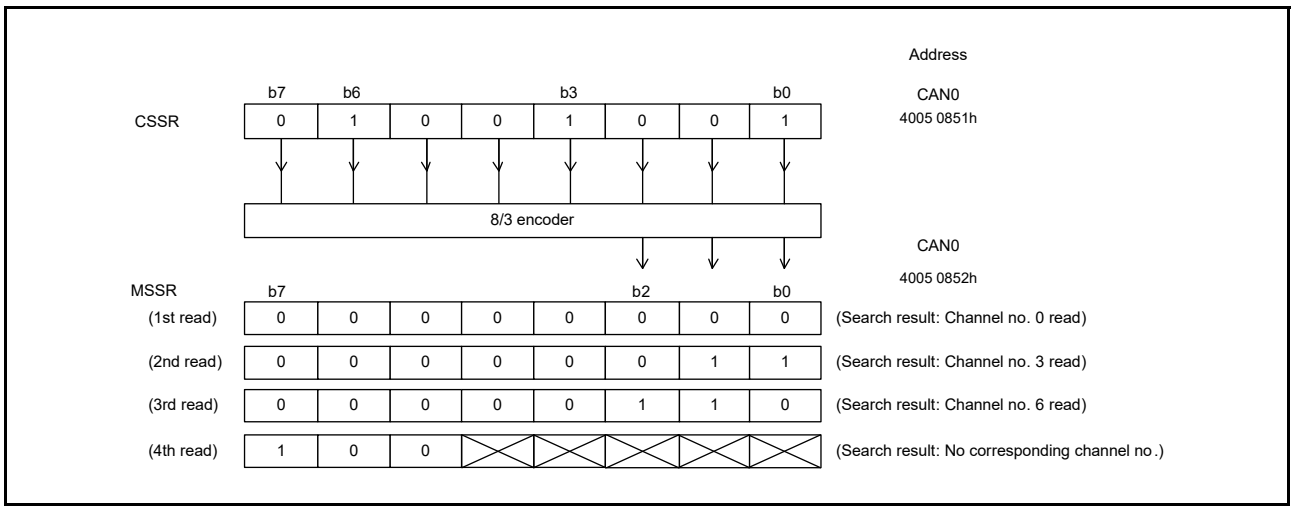


Figure 32.4 Writes to and reads from the CSSR and MSSR registers

The value of the CSSR register is also updated whenever the MSSR register is read. On this read, the value prior to conversion by the 8/3 encoder can be read.

32.2.19 Acceptance Filter Support Register (AFSR)

Address(es): CAN0.AFSR 4005 0856h

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

x: Undefined

Bit	Description	R/W
b15 to b0	After the standard ID of a received message is written, the value converted for data table search can be read	R/W

Note: Write to AFSR in CAN operation mode or CAN halt mode.

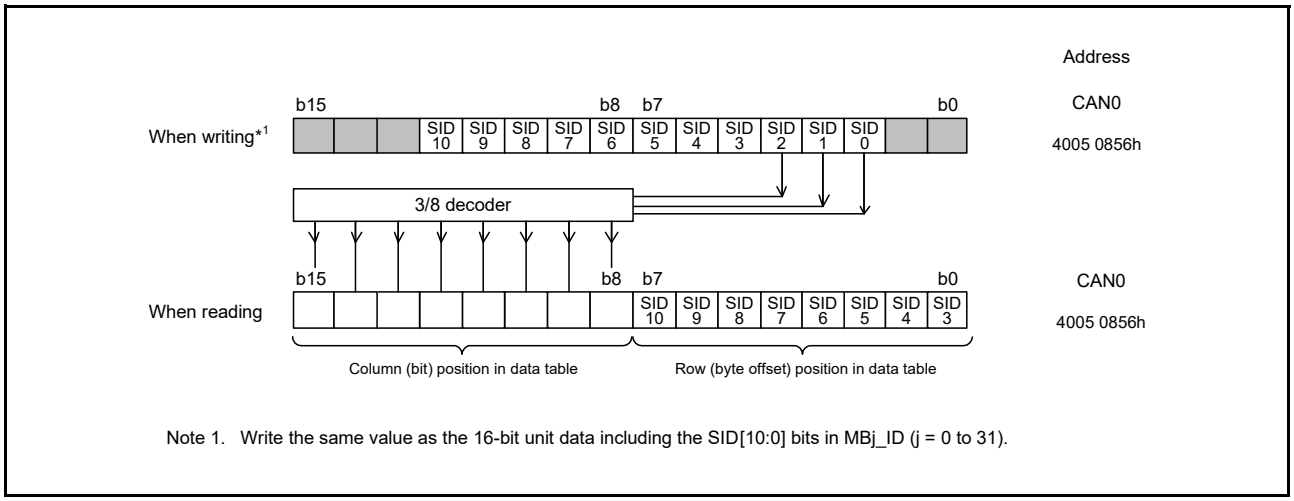
The acceptance filter support unit (ASU) can be used for data table (8 bits × 256) search. In the data table, all standard IDs that you create are set to be valid or invalid in bit units. When AFSR is written with data in 16-bit units including the SID[10:0] bit in MBj_ID (j = 0 to 31), in which a received standard ID is stored, a decoded row (byte offset) position and column (bit) position for data table search can be read. The ASU can be used for standard (11-bit) IDs only.

The ASU is enabled in the following cases:

- When the ID to be received cannot be masked by the acceptance filter. For example, if the IDs to be received are 078h, 087h, and 111h
- When there are too many IDs to receive and the software filtering time is expected to be shortened.

Note: The AFSR register cannot be set in CAN reset mode.

Figure 32.5 shows writes to and reads from the AFSR register.



Note 1. Write the same value as the 16-bit unit data including the SID[10:0] bits in MBj_ID (j = 0 to 31).

Figure 32.5 Writes to and reads from the AFSR register

32.2.20 Error Interrupt Enable Register (EIER)

Address(es): CAN0.EIER 4005 084Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	BLIE	OLIE	ORIE	BORIE	BOEIE	EPIE	EWIE	BEIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	BEIE	Bus Error Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b1	EWIE	Error-Warning Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b2	EPIE	Error-Passive Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b3	BOEIE	Bus-Off Entry Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b4	BORIE	Bus-Off Recovery Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b5	ORIE	Overrun Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b6	OLIE	Overload Frame Transmit Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W
b7	BLIE	Bus Lock Interrupt Enable	0: Disable interrupt 1: Enable interrupt.	R/W

The EIER register independently enables or disables the interrupt request for each error interrupt source in the EIFR register. Write to the EIER register in CAN reset mode.

BEIE bit (Bus Error Interrupt Enable)

When the BEIE bit is 0, no error interrupt request is generated even if the EIFR.BEIF bit is 1. When the BEIE bit is 1, an error interrupt request is generated if the EIFR.BEIF bit is set to 1.

EWIE bit (Error-Warning Interrupt Enable)

When the EWIE bit is 0, no error interrupt request is generated even if the EIFR.EWIF bit is 1. When the EWIE bit is 1, an error interrupt request is generated if the EIFR.EWIF bit is set to 1.

EPIE bit (Error-Passive Interrupt Enable)

When the EPIE bit is 0, no error interrupt request is generated even if the EIFR.EPIF bit is 1. When the EPIE bit is 1, an error interrupt request is generated if the EIFR.EPIF bit is set to 1.

BOEIE bit (Bus-Off Entry Interrupt Enable)

When the BOEIE bit is 0, no error interrupt request is generated even if the EIFR.BOEIF bit is 1. When the BOEIE bit is 1, an error interrupt request is generated if the EIFR.BOEIF bit is set to 1.

BORIE bit (Bus-Off Recovery Interrupt Enable)

When the BORIE bit is 0, an error interrupt request is not generated even if the EIFR.BORIF bit is 1. When the BORIE bit is set to 1, an error interrupt request is generated if the EIFR.BORIF bit is set to 1.

ORIE bit (Overrun Interrupt Enable)

When the ORIE bit is 0, an error interrupt request is not generated even if the EIFR.ORIF bit is 1. When the ORIE bit is 1, an error interrupt request is generated if the EIFR.ORIF bit is set to 1.

OLIE bit (Overload Frame Transmit Interrupt Enable)

When the OLIE bit is 0, no error interrupt request is generated even if the EIFR.OLIF bit is 1. When the OLIE bit is 1, an error interrupt request is generated if the EIFR.OLIF bit is set to 1.

BLIE bit (Bus Lock Interrupt Enable)

When the BLIE bit is 0, no error interrupt request is generated even if the EIFR.BLIF bit is 1. When the BLIE bit is 1, an error interrupt request is generated if the EIFR.BLIF bit is set to 1.

32.2.21 Error Interrupt Factor Judge Register (EIFR)

Address(es): CAN0.EIFR 4005 084Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	BLIF	OLIF	ORIF	BORIF	BOEIF	EPIF	EWIF	BEIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	BEIF	Bus Error Detect Flag	0: No bus error detected 1: Bus error detected.	R/W
b1	EWIF	Error-Warning Detect Flag	0: No error-warning detected 1: Error-warning detected.	R/W
b2	EPIF	Error-Passive Detect Flag	0: No error-passive detected 1: Error-passive detected.	R/W
b3	BOEIF	Bus-Off Entry Detect Flag	0: No bus-off entry detected 1: Bus-off entry detected.	R/W
b4	BORIF	Bus-Off Recovery Detect Flag	0: No bus-off recovery detected 1: Bus-off recovery detected.	R/W
b5	ORIF	Receive Overrun Detect Flag	0: No receive overrun detected 1: Receive overrun detected.	R/W
b6	OLIF	Overload Frame Transmission Detect Flag	0: No overload frame transmission detected 1: Overload frame transmission detected.	R/W
b7	BLIF	Bus Lock Detect Flag	0: No bus lock detected 1: Bus lock detected.	R/W

If an event associated with one of these bits occurs, the associated bit in the EIFR register is set to 1, regardless of the EIER register setting.

Clear the bits to 0 through a software write. If a bit is set to 1 at the same time that the software clears it, the bit becomes 1. When setting a single bit to 0 in software, use the transfer instruction (MOV) to ensure that only the specified bit is set to 0 and the other bits are set to 1. Writing 1 has no effect to these bit values.

BEIF flag (Bus Error Detect Flag)

The BEIF flag is set to 1 when a bus error is detected.

EWIF flag (Error-Warning Detect Flag)

The EWIF flag is set to 1 when the value of the receive error counter (REC) or transmit error counter (TEC) exceeds 95. It is set to 1 only when REC or TEC initially exceeds 95. If 0 is written to the EWIF flag by software while REC or TEC remains greater than 95, the EWIF flag is not set to 1 until REC or TEC goes below 95 then exceeds 95 again.

EPIF flag (Error-Passive Detect Flag)

The EPIF flag is set to 1 when the CAN error state becomes error-passive, when the receive error counter (REC) or transmit error counter (TEC) value exceeds 127. The EPIF flag is set to 1 only when the REC or TEC initially exceeds 127. If 0 is written to the EPIF flag by software while the REC or TEC remains greater than 127, the EPIF flag is not set to 1 until REC or TEC goes below 127, then exceeds 127 again.

BOEIF flag (Bus-Off Entry Detect Flag)

The BOEIF flag is set to 1 when the CAN error state becomes bus-off, when the transmit error counter (TEC) value exceeds 255. The BOEIF flag is also set to 1 when the CTLR.BOM[1:0] bits are 01b (automatic entry to CAN halt mode on bus-off entry) and the CAN module enters the bus-off state.

BORIF flag (Bus-Off Recovery Detect Flag)

The BORIF flag is set to 1 when the CAN module recovers from the bus-off state normally by detecting 11 consecutive recessive bits 128 times in the following conditions:

- When the CTLR.BOM[1:0] bits are 00b
- When the CTLR.BOM[1:0] bits are 10b
- When the CTLR.BOM[1:0] bits are 11b.

However, the BORIF flag is not set to 1 if the CAN module recovers from the bus-off state in the following conditions:

- When the CTLR.CANM[1:0] bits are set to 01b or 11b (CAN reset mode)
- When the CTLR.RBOC bit is set to 1 (forcible return from bus-off)
- When the CTLR.BOM[1:0] bits are set to 01b
- When the CTLR.BOM[1:0] bits are set to 11b and the CTLR.CANM[1:0] bits are set to 10b (CAN halt mode) before normal recovery occurs.

Table 32.7 lists the behavior of the BOEIF and BORIF bits for each CTLR.BOM[1:0] setting.

Table 32.7 Behavior of BOEIF and BORIF flags for each CTLR.BOM[1:0] setting

BOM[1:0] bits	BOEIF bit	BORIF bit
00b	Set to 1 on entry to the bus-off state	Set to 1 on exit from the bus-off state
01b		Do not set to 1
10b		Set to 1 on exit from the bus-off state
11b		Set to 1 if normal bus-off recovery occurs before the CANM[1:0] bits are set to 10b (CAN halt mode)

ORIF flag (Receive Overrun Detect Flag)

The ORIF flag is set to 1 when a receive overrun occurs. This bit is not set to 1 in overwrite mode.

In overwrite mode, a reception complete interrupt request is generated if an overwrite condition occurs and the ORIF bit is not set to 1.

In overrun mode with normal mailbox mode, if an overrun occurs in any of mailboxes 0 to 31, the ORIF flag is set to 1. In overrun mode with FIFO mailbox mode, if an overrun occurs in any of mailboxes 0 to 23 or the receive FIFO, the ORIF flag is set to 1.

OLIF flag (Overload Frame Transmission Detect Flag)

The OLIF flag is set to 1 if the transmitting condition of an overload frame is detected when the CAN module is transmitting or receiving.

BLIF flag (Bus Lock Detect Flag)

The BLIF flag is set to 1 if 32 consecutive dominant bits are detected on the CAN bus while the CAN module is in CAN operation mode. After the BLIF bit is set to 1, 32 consecutive dominant bits are detected again under either of the following conditions:

- Recessive bits are detected after the BLIF flag changes to 0 from 1
- The CAN module enters CAN reset mode or CAN halt mode and then enters CAN operation mode again after the BLIF flag changes to 0 from 1.

32.2.22 Receive Error Count Register (RECR)

Address(es): CAN0.RECR 4005 084Eh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Description	R/W
b7 to b0	Receive error count function. RECR increments or decrements the counter value based on the error status of the CAN module during reception.	R

The RECR register indicates the value of the receive error counter. See the CAN Specification (ISO11898-1) for the increment and decrement conditions of the receive error counter.

The value of the RECR register in the bus-off state is undefined.

32.2.23 Transmit Error Count Register (TECR)

Address(es): CAN0.TECR 4005 084Fh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Description	R/W
b7 to b0	Transmit error count function. TECR increments or decrements the counter value based on the error status of the CAN module during transmission.	R

The TECR register indicates the value of the transmit error counter. See the CAN Specification (ISO11898-1) for the increment and decrement conditions of the transmit error counter.

The value of the TECR register in the bus-off state is undefined.

32.2.24 Error Code Store Register (ECSR)

Address(es): CAN0.ECSR 4005 0850h

	b7	b6	b5	b4	b3	b2	b1	b0
	EDPM	ADEF	BE0F	BE1F	CEF	AEF	FEF	SEF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SEF	Stuff Error Flag*1,*2	0: No stuff error detected 1: Stuff error detected.	R/W
b1	FEF	Form Error Flag*1,*2	0: No form error detected 1: Form error detected.	R/W
b2	AEF	ACK Error Flag*1,*2	0: No ACK error detected 1: ACK error detected.	R/W
b3	CEF	CRC Error Flag*1,*2	0: No CRC error detected 1: CRC error detected.	R/W

Bit	Symbol	Bit name	Description	R/W
b4	BE1F	Bit Error (recessive) Flag*1,*2	0: No bit error (recessive) detected 1: Bit error (recessive) detected.	R/W
b5	BE0F	Bit Error (dominant) Flag*1,*2	0: No bit error (dominant) detected 1: Bit error (dominant) detected.	R/W
b6	ADEF	ACK Delimiter Error Flag*1,*2	0: No ACK delimiter error detected 1: ACK delimiter error detected.	R/W
b7	EDPM	Error Display Mode Select*3,*4	0: Output of first detected error code 1: Output of accumulated error code.	R/W

Note 1. Writing 1 has no effect on these bit values.
Note 2. To write 0 to the SEF, FEF, AEF, CEF, BE1F, BE0F, and ADEF bits, use the transfer (MOV) instruction to ensure that only the specified bit is set to 0 and the other bits are set to 1.
Note 3. Write to the EDPM bit in CAN reset mode or CAN halt mode.
Note 4. If more than one error condition is detected simultaneously, all related bits are set to 1.

The ECSR register indicates whether an error occurs on the CAN bus. See the CAN Specification (ISO11898-1) to check the conditions when each error occurs.

Clear all of the bits except for the EDPM bit to 0 through a software write. If an ECSR bit is set to 1 at the same time that software clears it, the bit becomes 1.

SEF flag (Stuff Error Flag)

The SEF flag is set to 1 when a stuff error is detected.

FEF flag (Form Error Flag)

The FEF flag is set to 1 when a form error is detected.

AEF flag (ACK Error Flag)

The AEF flag is set to 1 when an ACK error is detected.

CEF flag (CRC Error Flag)

The CEF flag is set to 1 when a CRC error is detected.

BE1F flag (Bit Error (recessive) Flag)

The BE1F flag is set to 1 when a recessive bit error is detected.

BE0F flag (Bit Error (dominant) Flag)

The BE0F flag is set to 1 when a dominant bit error is detected.

ADEF flag (ACK Delimiter Error Flag)

The ADEF flag is set to 1 when a form error is detected with the ACK delimiter during transmission.

EDPM bit (Error Display Mode Select)

The EDPM bit selects the output mode of the ECSR register. When the EDPM bit is set to 0, the ECSR register outputs the first error code. When the EDPM bit is set to 1, the ECSR register outputs the accumulated error code.

32.2.25 Time Stamp Register (TSR)

Address(es): CAN0.TSR 4005 0854h

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

Bit	Description	R/W
b15 to b0	Free-running counter value for the time stamp function	R

Note 1. Read the TSR register in 16-bit units.

Reading the TSR register returns the current value of the 16-bit free-running time stamp counter. The time stamp counter reference clock is configured in the TSPS[1:0] bits in the CTLR register. The counter stops in CAN sleep mode and CAN halt mode, and is initialized in CAN reset mode. The time stamp counter value is stored to bits TSL[7:0] and TSH[7:0] in the MBj_TS register when a received message is stored in a receive mailbox.

32.2.26 Test Control Register (TCR)

Address(es): CAN0.TCR 4005 0858h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TSTM[1:0]		TSTE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	TSTE	CAN Test Mode Enable	0: Disable CAN test mode 1: Enable CAN test mode.	R/W
b2, b1	TSTM[1:0]	CAN Test Mode Select	b2 b1 0 0: Not CAN test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback) 1 1: Self-test mode 1 (internal loopback).	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TCR register controls the CAN test mode. Write to TCR register in CAN halt mode only.

(1) Listen-Only Mode

The CAN Specification (ISO11898-1) recommends an optional bus monitoring mode. In listen-only mode, valid data frames and valid remote frames can be received. However, only recessive bits can be sent on the CAN bus. The ACK bit, overload flag, and active error flag cannot be sent.

Listen-only mode can be used for baud rate detection. Do not request transmission from any mailboxes in listen-only mode.

Figure 32.6 shows the connection when listen-only mode is selected.

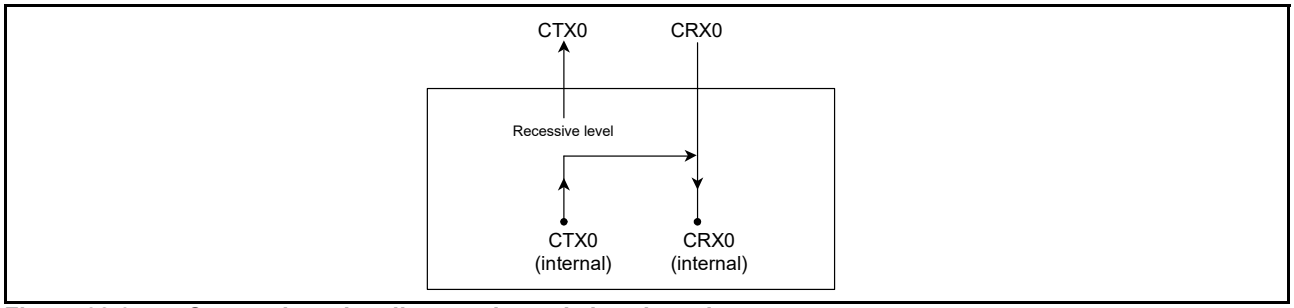


Figure 32.6 Connection when listen-only mode is selected

(2) Self-Test Mode 0 (External Loopback)

Self-test mode 0 is provided for CAN transceiver tests. In this mode, the protocol module treats its own transmitted messages as those received by the CAN transceiver and stores them into the receive mailbox. To be independent from external stimulation, the protocol module generates the ACK bit. Connect the CTX0 and CRX0 pins to the transceiver.

Figure 32.7 shows the connection when self-test mode 0 is selected.

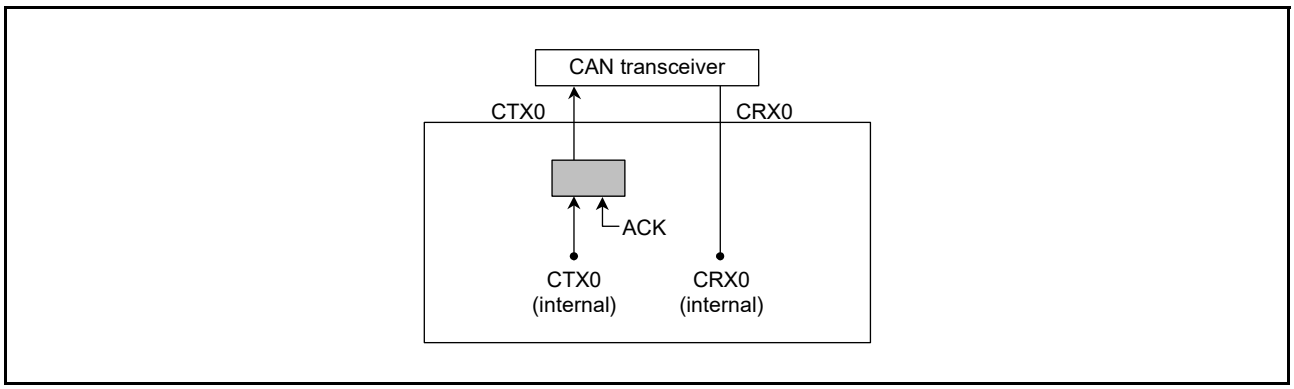


Figure 32.7 Connection when self-test mode 0 is selected

(3) Self-Test Mode 1 (Internal Loopback)

Self-test mode 1 is provided for self-test functions.

In self-test mode 1, the protocol controller treats its transmitted messages as received messages and stores them into the receive mailbox. To be independent from external stimulation, the protocol controller generates the ACK bit.

In self-test mode 1, the protocol controller performs internal loopback from the internal CTX0 pin to the internal CRX0 pin. The input value of the external CRX0 pin is ignored. The external CTX0 pin outputs only recessive bits. The CTX0 and CRX0 pins are not required to be connected to the CAN bus or any external device.

Figure 32.8 shows the connection when self-test mode 1 is selected.

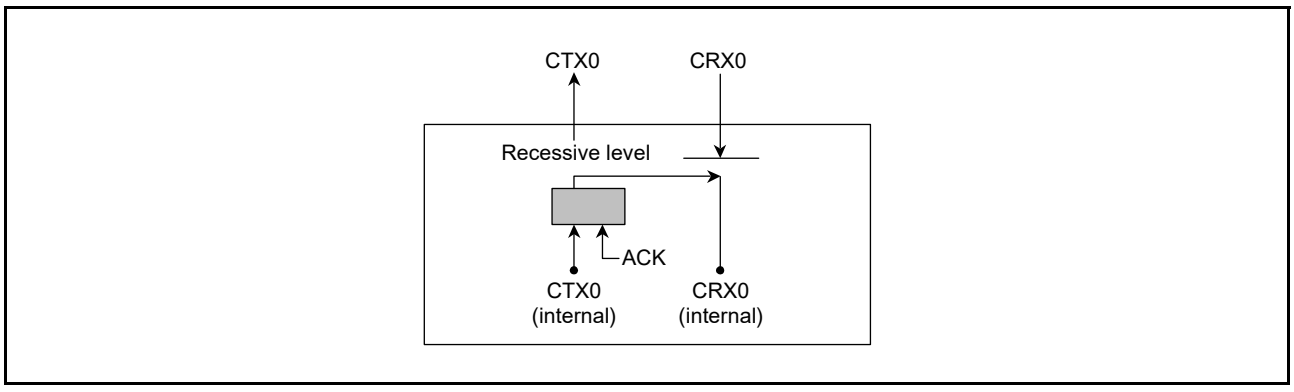


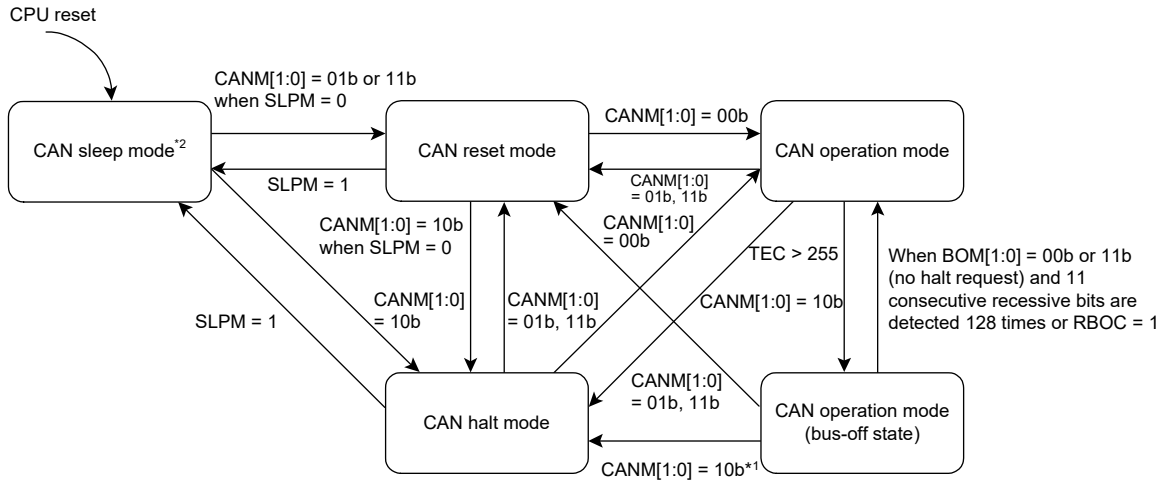
Figure 32.8 Connection when self-test mode 1 is selected

32.3 Modes of Operation

The CAN module has four modes of operation:

- CAN reset mode
- CAN halt mode
- CAN operation mode
- CAN sleep mode.

Figure 32.9 shows the transition between different modes of operation.



CANM, SLPM, BOM, RBOC: Bits in CTLR

Note 1. The transition timing from the bus-off state to CAN halt mode depends on the setting of the CTLR.BOM[1:0] bits.
When the CTLR.BOM[1:0] bits are 01b, the state transition occurs immediately after entering the bus-off state.
When the CTLR.BOM[1:0] bits are 10b, the state transition occurs at the end of the bus-off state.
When the CTLR.BOM[1:0] bits are 11b, the state transition occurs at the setting of the CTLR.CANM[1:0] bits to 10b (CAN halt mode).
Note 2. Change the CTLR.SLPM bit to set or cancel CAN sleep mode.

Figure 32.9 Transition between different modes of operation

32.3.1 CAN Reset Mode

CAN reset mode is provided for CAN communication configuration.

When the CTLR.CANM[1:0] bits are set to 01b or 11b, the CAN module enters CAN reset mode. The STR.RSTST bit is then set to 1. Do not change the CTLR.CANM[1:0] bits until the RSTST bit is set to 1. Set the BCR register before exiting CAN reset mode to any other mode.

The following registers are initialized to their reset values after entering CAN reset mode, and their initial values are retained during CAN reset mode:

- MCTL_TXj and MCTL_RXj
- STR (except for the SLPST and TFST bits)
- EIFR
- RECR
- TECR
- TSR
- MSSR
- MSMR
- RFCR
- TFCR
- TCR
- ECSR (except for the EDPM bit).

The following registers retain their previous values even after entering CAN reset mode:

- CTLR
- STR (only the SLPST and TFST bits)
- MIER and MIER_FIFO
- EIER
- BCR
- CSSR
- ECSR (only the EDPM bit)
- MBj_ID, MBj_DL, MBj_Dm and MBj_TS
- MKRk
- FIDCR0 and FIDCR1
- MKIVLR
- AFSR
- RFPCR
- TFPCR.

32.3.2 CAN Halt Mode

CAN halt mode is used for mailbox configuration and test mode setting.

When the CTLR.CANM[1:0] bits are set to 10b, CAN halt mode is selected and the STR.HLTST bit is set to 1. Do not change the CTLR.CANM[1:0] bits until the HLTST bit is 1. See Table 32.8 for the state transition conditions when transmitting or receiving.

All registers except for bits RSTST, HLTST, and SLPST in the STR register remain unchanged when the CAN enters CAN halt mode.

Do not change the CTLR register (except for bits CANM[1:0] and SLPM) and EIER register in CAN halt mode. The BCR register can be changed in CAN halt mode only when listen-only mode is selected for automatic baud rate detection.

Table 32.8 Operation in CAN reset mode and CAN halt mode

Operation mode	Receiver	Transmitter	Bus-off
CAN reset mode (forcible transition) CANM[1:0] = 11b	CAN module enters CAN reset mode without waiting for the end of message reception	CAN module enters CAN reset mode without waiting for the end of message transmission	CAN module enters CAN reset mode without waiting for the end of bus-off recovery
CAN reset mode CANM[1:0] = 01b	CAN module enters CAN reset mode without waiting for the end of message reception	CAN module enters CAN reset mode after waiting for the end of message transmission*1,*4	CAN module enters CAN reset mode without waiting for the end of bus-off recovery
CAN halt mode	CAN module enters CAN halt mode after waiting for the end of message reception*2,*3	CAN module enters CAN halt mode after waiting for the end of message transmission*1,*4	• When the BOM[1:0] bits are 00b: A halt request from the software is accepted only after bus-off recovery. • When the BOM[1:0] bits are 01b: CAN module automatically enters CAN halt mode without waiting for the end of bus-off recovery, regardless of a halt request from software. • When the BOM[1:0] bits are 10b: CAN module automatically enters CAN halt mode after waiting for the end of bus-off recovery, regardless of a halt request from software. • When the BOM[1:0] bits are 11b: CAN module enters CAN halt mode without waiting for the end of bus-off recovery, if a halt is requested by software during bus-off.

Note 1. If transmission of multiple messages is requested, a mode transition occurs after completion of the first transmission. If the CAN reset mode is being requested during suspend transmission, mode transition occurs when the bus is idle, the next transmission ends, or the CAN module becomes a receiver.

Note 2. If the CAN bus is locked at the dominant level, the program can detect this state by monitoring the BLIF bit in the EIFR register.

Note 3. If a CAN bus error occurs during reception after CAN halt mode is requested, the CAN module transitions to CAN halt mode.

Note 4. If a CAN bus error or arbitration-lost occurs during transmission after CAN reset mode or CAN halt mode is requested, the CAN module transitions to the requested CAN mode.

32.3.3 CAN Sleep Mode

CAN sleep mode reduces current consumption by stopping the clock supply to the CAN module. After a reset from an MCU pin or a software reset, the CAN module starts from CAN sleep mode.

When the SLPM bit in CTLR is set to 1, the CAN module enters CAN sleep mode and the STR.SLPST bit is then set to 1. Do not change the value of the SLPM bit until the SLPST bit is 1. The other registers remain unchanged when the CAN module enters CAN sleep mode.

Write to the SLPM bit in CAN reset mode and CAN halt mode. Do not change any registers (except for the SLPM bit) during CAN sleep mode. Read operation is still allowed.

When the SLPM bit is set to 0, the CAN module is released from CAN sleep mode. When the CAN module exits CAN sleep mode, the other registers remain unchanged.

32.3.4 CAN Operation Mode (Excluding Bus-Off State)

CAN operation mode is used for CAN communication.

When the CTLR.CANM[1:0] bits are set to 00b, the CAN module enters CAN operation mode. The RSTST and HLTST bits are then set to 0. Do not change the value of the CANM[1:0] bits until bits the RSTST and HLTST bits are set to 0. If 11 consecutive recessive bits are detected after entering CAN operation mode:

- The CAN module becomes an active node on the network, which enables transmission and reception of CAN messages
- Error monitoring of the CAN bus, such as receive and transmit error counters, is performed.

During CAN operation mode, the CAN module is in one of the following three sub-modes, depending on the status of the CAN bus:

- Idle mode: No transmission or reception is occurring
- Receive mode: A CAN message sent by another node is being received
- Transmit mode: A CAN message is being transmitted. The CAN module receives a message transmitted by the local node simultaneously when self-test mode 0 (TSTM[1:0] bits in TCR = 10b) or self-test mode 1 (TSTM[1:0] bits = 11b) is selected.

Figure 32.10 shows the sub-modes of CAN operation mode.

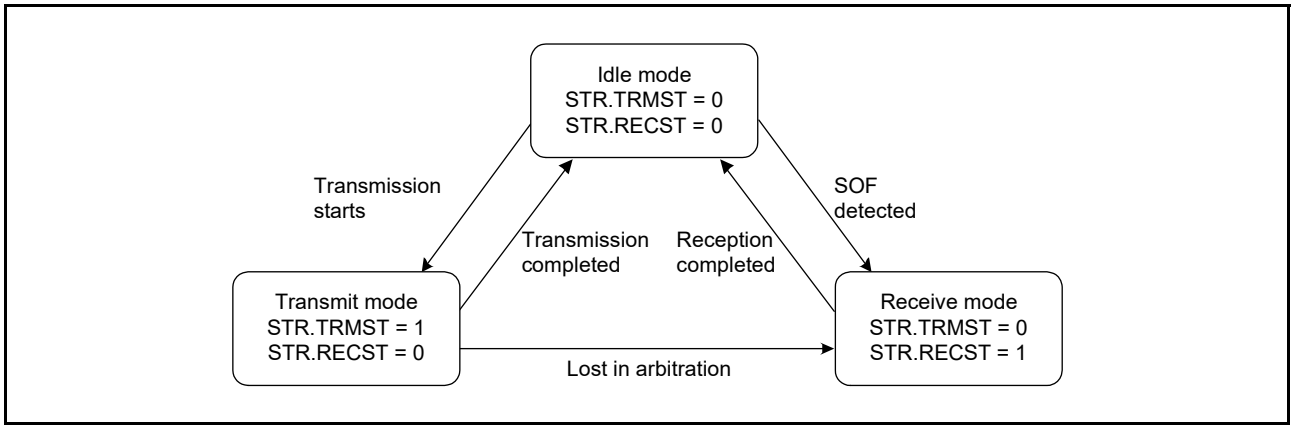


Figure 32.10 Sub-modes of CAN operation mode

32.3.5 CAN Operation Mode (Bus-Off State)

The CAN module enters the bus-off state based on the increment or decrement rules for the transmit or receive error counters, as defined in the CAN specification.

The following cases apply when the CAN module is recovering from the bus-off state. When the CAN module is in the bus-off state, the values of the CAN module registers, except for STR, EIFR, RECR, TECR and TSR, remain unchanged.

(1) When CTLR.BOM[1:0] = 00b (Normal Mode)

The CAN module enters the error-active state after it completes recovery from the bus-off state and CAN communication is enabled. The EIFR.BORIF flag is set to 1 (bus-off recovery detected).

(2) When CTLR.RBOC = 1 (Forced Return from Bus-Off)

The CAN module enters the error-active state when it is in the bus-off state and the RBOC bit is set to 1. CAN communication is enabled again after 11b consecutive recessive bits are detected. The BORIF bit is not set to 1.

(3) When CTLR.BOM[1:0] = 01b (Automatic Transition to CAN Halt Mode on Bus-Off Entry)

The CAN module enters CAN halt mode when it reaches the bus-off state. The BORIF bit is not set to 1.

(4) When CTLR.BOM[1:0] = 10b (Automatic Transition to CAN Halt Mode on Bus-Off End)

The CAN module enters CAN halt mode when it completes the recovery from bus-off. The BORIF bit is set to 1.

(5) When CTLR.BOM[1:0] = 11b (Automatic Transition to CAN Halt Mode through Software) and CTLR.CANM[1:0] = 10b (CAN Halt Mode) during Bus-Off State

The CAN module enters CAN halt mode when it is in the bus-off state and the CANM[1:0] bits are set to 10b (CAN halt mode). The EIFR.BORIF flag is not set to 1.

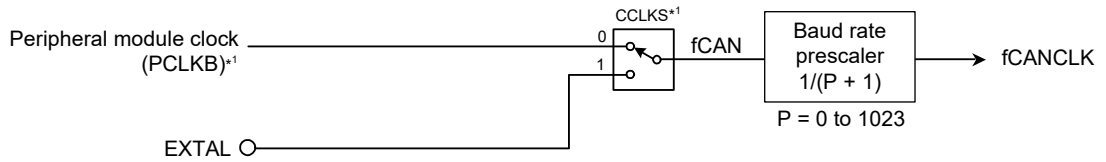
If the CANM[1:0] bits are not set to 10b during bus-off, the same behavior as (1) applies.

32.4 Data Transfer Rate Configuration

This section describes how to configure the data transfer rate.

32.4.1 Clock Setting

The CAN module includes a CAN clock generator, as shown in Figure 32.11. The CAN clock can be set by the CCLKS bit and the BRP[9:0] bits in the BCR register.



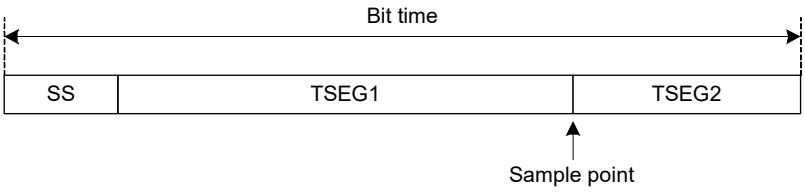
CCLKS: Bit in the BCR register
fCAN: CAN system clock
P: Value selected by BRP[9:0] bits in BCR (P = 0 to 1023)
fCANCLK: CAN communication clock (fCANCLK = fCAN/(P + 1))

Note 1. See section 9, Clock Generation Circuit. When using the CAN module while the CCLKS bit is set to 0, the PLL must be selected as the source of the peripheral module clock.

Figure 32.11 CAN clock generator block diagram

32.4.2 Bit Time Setting

The bit time consists of three segments shown in Figure 32.12.



The range of each segment: Bit time = 8 Tq to 25 Tq
SS = 1 Tq
TSEG1 = 4 Tq to 16 Tq
TSEG2 = 2 Tq to 8 Tq
SJW = 1 Tq to 4 Tq
Setting of TSEG1 and TSEG2: TSEG1 > TSEG2 ≥ SJW

Figure 32.12 Bit timing

32.4.3 Data Transfer Rate

The data transfer rate depends on the division value of fCAN (CAN system clock), the division value of the baud rate prescaler, and the number of Tq count for 1 bit time.

$$\text{Data transfer rate [bps]} = \frac{\text{fCAN}}{\text{Baud rate prescaler division value*1 x number of Tq of 1 bit time}} = \frac{\text{fCANCLK}}{\text{Number of Tq of 1 bit time}}$$

Note 1. Division value of baud rate prescaler = P + 1 (P: 0 to 1023) where P is the BRP[9:0] setting in the BCR register.

Table 32.9 lists data transfer rate examples.

Table 32.9 Data transfer rate examples when fCAN = 32 MHz

Data transfer rate	Tq count	P + 1
1 Mbps	8Tq 16Tq	4 2
500 kbps	8Tq 16Tq	8 4
250 kbps	8Tq 16Tq	16 8
125 kbps	8Tq 16Tq	32 16
83.3 kbps	8Tq 16Tq	48 24
33.3 kbps	8Tq 10Tq 16Tq 20Tq	120 96 60 48

32.5 Mailbox and Mask Register Structure

Figure 32.13 shows the structure of the 32 mailbox registers (MBj_ID, MBj_DL, MBj_Dm, and MBj_TS).

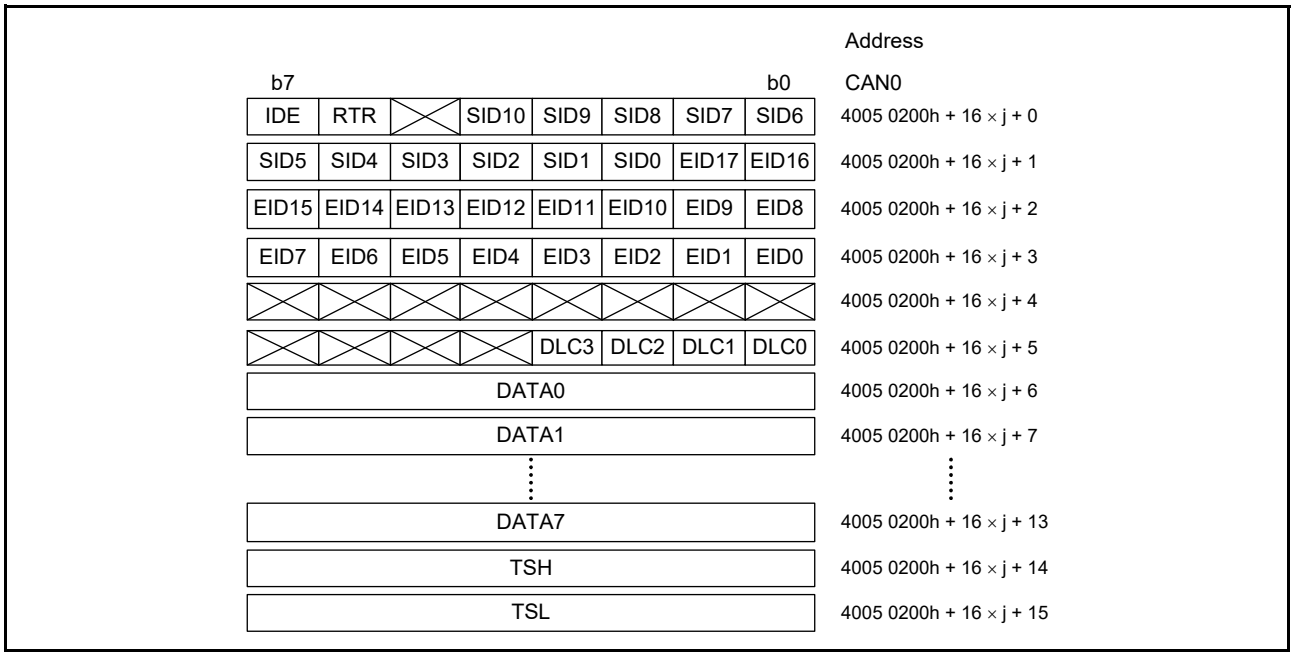


Figure 32.13 Structure of mailbox register (j = 0 to 31)

Figure 32.14 shows the structure of the eight mask registers MKRk.

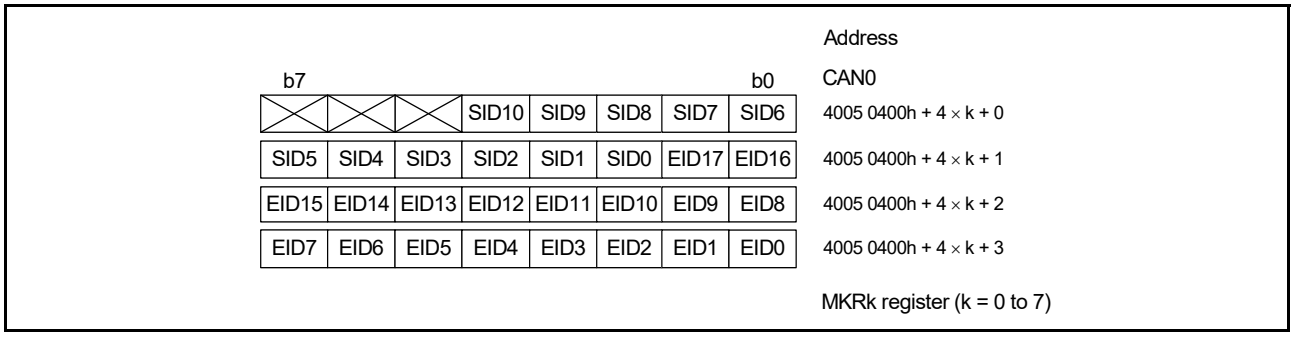


Figure 32.14 Structure of MKRk registers (k = 0 to 7)

Figure 32.15 shows the structure of the two FIFO receive ID compare registers, FIDCR0 and FIDCR1.

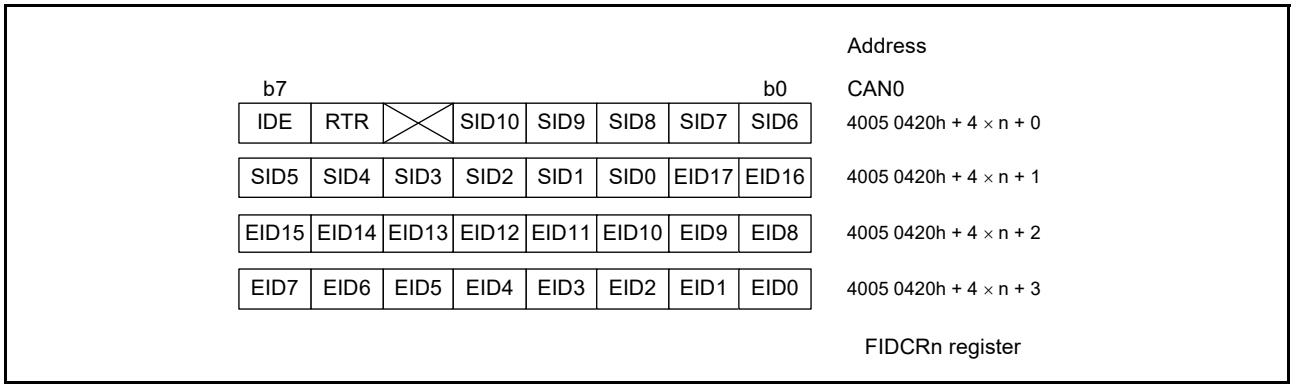


Figure 32.15 Structure of the FIDCRn registers (n = 0, 1)

32.6 Acceptance Filtering and Masking Functions

The acceptance filtering and masking functions allows you to select and receive messages with multiple IDs for mailboxes within a specified range.

The MKRk registers can mask the standard ID and the extended ID of 29 bits:

- MKR0 is the mask register for mailboxes 0 to 3
- MKR1 is the mask register for mailboxes 4 to 7
- MKR2 is the mask register for mailboxes 8 to 11
- MKR3 is the mask register for mailboxes 12 to 15
- MKR4 is the mask register for mailboxes 16 to 19
- MKR5 is the mask register for mailboxes 20 to 23
- MKR6 is the mask register for mailboxes 24 to 27 in normal mailbox mode and receive FIFO mailboxes 28 to 31 in FIFO mailbox mode
- MKR7 is the mask register for mailboxes 28 to 31 in normal mailbox mode and receive FIFO mailboxes 28 to 31 in FIFO mailbox mode.

The MKIVLR register disables acceptance filtering independently for each mailbox.

The IDE bit in the MBj_ID register is valid when the CTLR.IDFM[1:0] bits are 10b (mixed ID mode).

The RTR bit in the MBj_ID register selects a data frame or a remote frame.

In FIFO mailbox mode, normal mailboxes (0 to 23) use the associated register (MKR0 to MKR5) for acceptance filtering. The receive FIFO mailboxes (28 to 31) use two registers, MKR6 and MKR7, for acceptance filtering.

The receive FIFO also uses two registers, FIDCR0 and FIDCR1, for ID comparison. The EID[17:0], SID[10:0], RTR, and IDE bits in mailbox 28 to mailbox 31 for the receive FIFO are disabled. As acceptance filtering depends on the result of two logic OR operations, two ranges of IDs can be received into the receive FIFO.

The MKIVLR register is disabled for the receive FIFO.

If different values are set in the IDE bits in the FIDCR0 and FIDCR1 registers, both ID formats are received.

If different values are set in the RTR bits in the FIDCR0 and FIDCR1 registers, both data and remote frames are received.

When a combination of two ranges of IDs is not necessary, set the same mask value and the same ID into both the FIFO ID and mask registers.

Figure 32.16 shows the associations between mask registers and mailboxes. Figure 32.17 shows acceptance filtering.

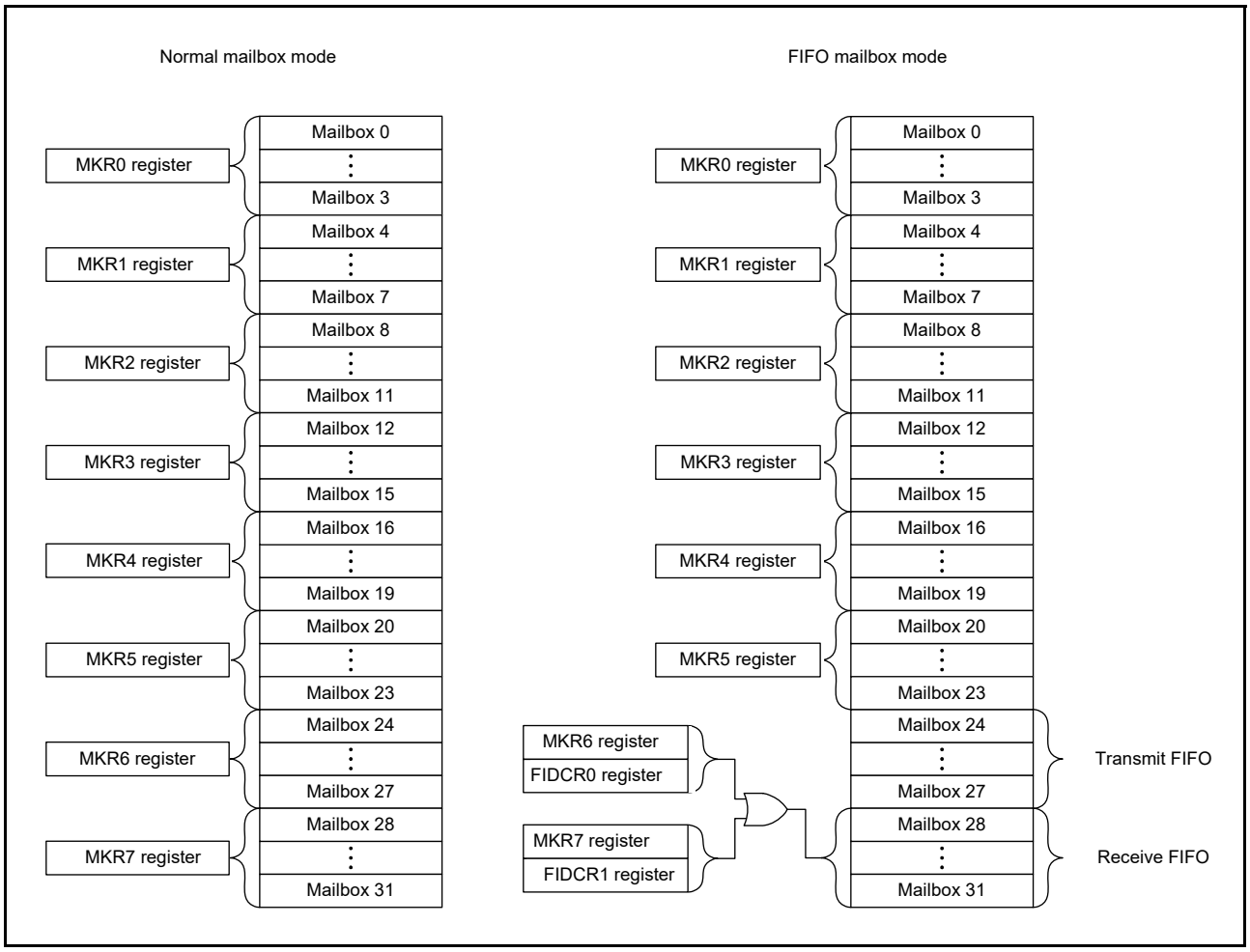


Figure 32.16 Associations between mask registers and mailboxes

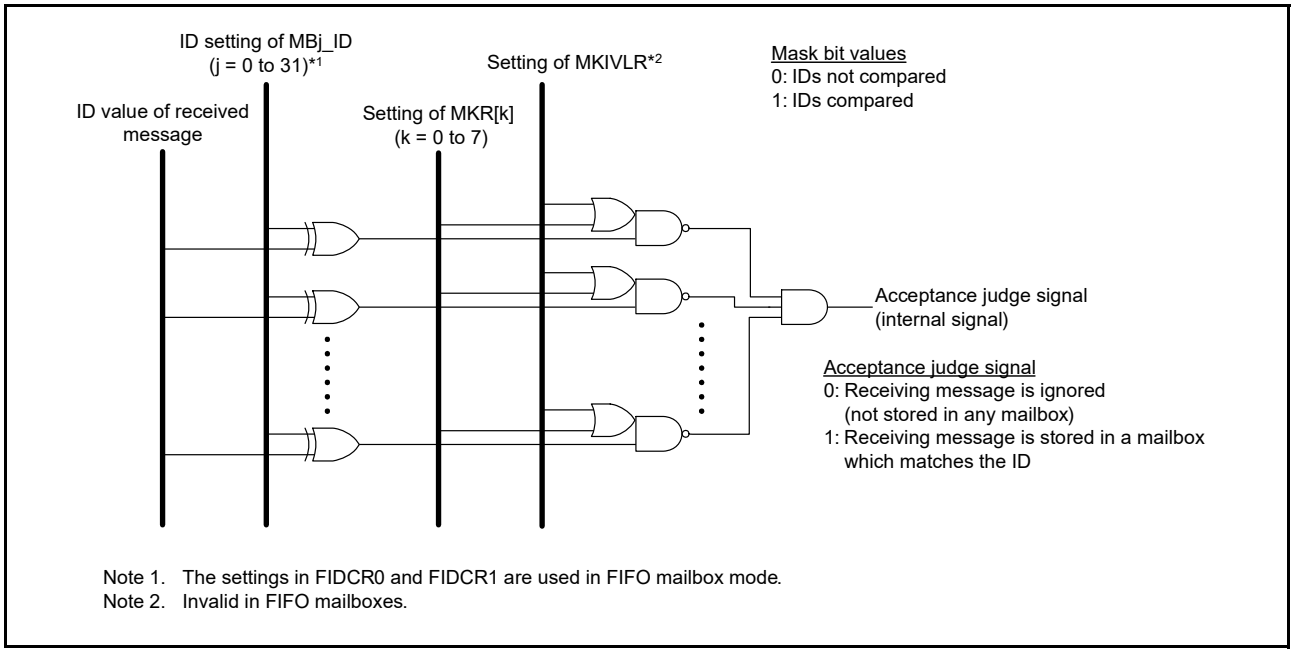


Figure 32.17 Acceptance filtering

32.7 Reception and Transmission

Table 32.10 lists the CAN communication mode settings.

Table 32.10 Settings for CAN receive and transmit modes

MCTL_TXj and MCTL_RXjTRMREQ	MCTL_TXj and MCTL_RXjRECREQ	MCTL_TXj and MCTL_RXjONESHOT	Mailbox communication mode
0	0	0	Mailbox disabled or transmission being aborted
0	0	1	Can be configured only when transmission or reception from a mailbox programmed in one-shot mode is aborted
0	1	0	Configured as a receive mailbox for a data frame or a remote frame
0	1	1	Configured as a one-shot receive mailbox for a data frame or a remote frame
1	0	0	Configured as a transmit mailbox for a data frame or a remote frame
1	0	1	Configured as a one-shot transmit mailbox for a data frame or a remote frame
1	1	0	Do not set
1	1	1	Do not set

j = 0 to 31

When a mailbox is configured as a receive mailbox or a one-shot receive mailbox:

1. Before configuring a mailbox, set the MCTL_RXj register to 00h.
2. A received message is stored into the first mailbox that matches the condition resulting from the receive mode settings and acceptance filtering. The matching mailbox with the smaller number takes priority for storing the received message.
3. In CAN operation mode, the CAN module does not receive its own transmitted data even when ID is a match. In self-test mode, however, the CAN module receives its own transmitted data and returns ACK.

When configuring a mailbox as a transmit mailbox or a one-shot transmit mailbox:

1. Before configuring the mailbox, ensure that the MCTL_TXj register is 00h and that there is no pending abort process.

32.7.1 Reception

Figure 32.18 shows an operation example of data frame reception in overwrite mode.

The example shows the overwriting of the first message when the CAN module receives two consecutive CAN messages that match the receiving conditions in MCTL_RXj (j = 0 to 31).

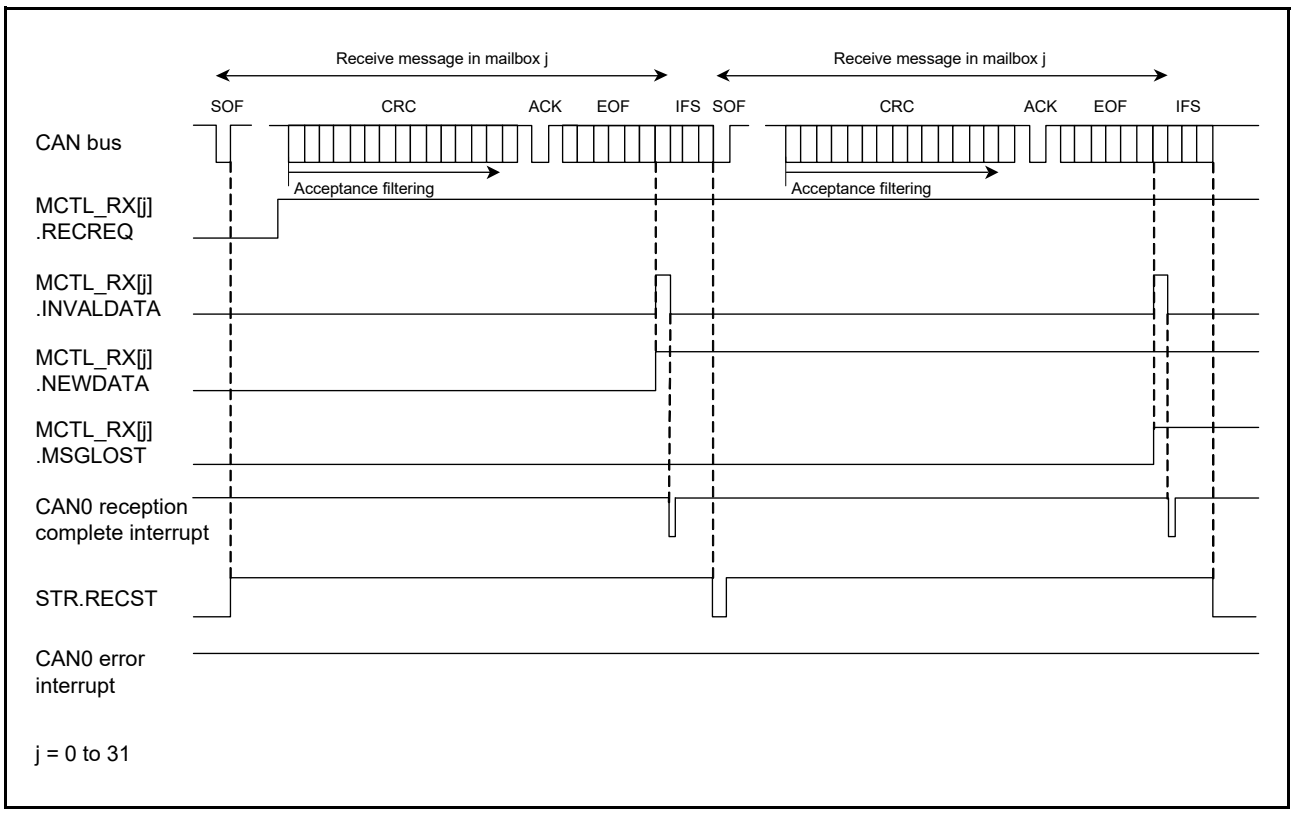


Figure 32.18 Operation example of data frame reception in overwrite mode

1. When an SOF is detected on the CAN bus, the RECST bit in the STR register is set to 1 (reception in progress) if the CAN module has no message ready to start transmission.
2. Acceptance filtering starts at the beginning of the CRC field to select the receive mailbox.
3. After a message is received, the MCTL_RXj.NEWDATA bit for the receive mailbox is set to 1 (new message is being stored or was stored to the mailbox). The MCTL_RXj.INVALDATA flag is set to 1 (message is being updated) at the same time. The INVALDATA flag is set to 0 (message valid) again after the complete message is transferred to the mailbox.
4. If the interrupt enable bit in the MIER register for the receive mailbox is 1 (interrupt enabled), the CAN0 reception complete interrupt request is generated when the INVALDATA flag is set to 0.
5. After the message is read from the mailbox, the NEWDATA bit must be set to 0 by software.
6. In overwrite mode, if the next CAN message is received while the NEWDATA bit in MCTL_RXj is set to 1, the MSGLOST bit in MCTL_RXj is set to 1 (message was overwritten). The new received message is transferred to the mailbox. The CAN0 reception complete interrupt request is generated in the same way as in step 4.

Figure 32.19 shows an operation example of data frame reception in overrun mode. The example shows the overrunning of the second message when the CAN module receives two consecutive CAN messages that match the receiving conditions of MCTL_RXj (j = 0 to 31).

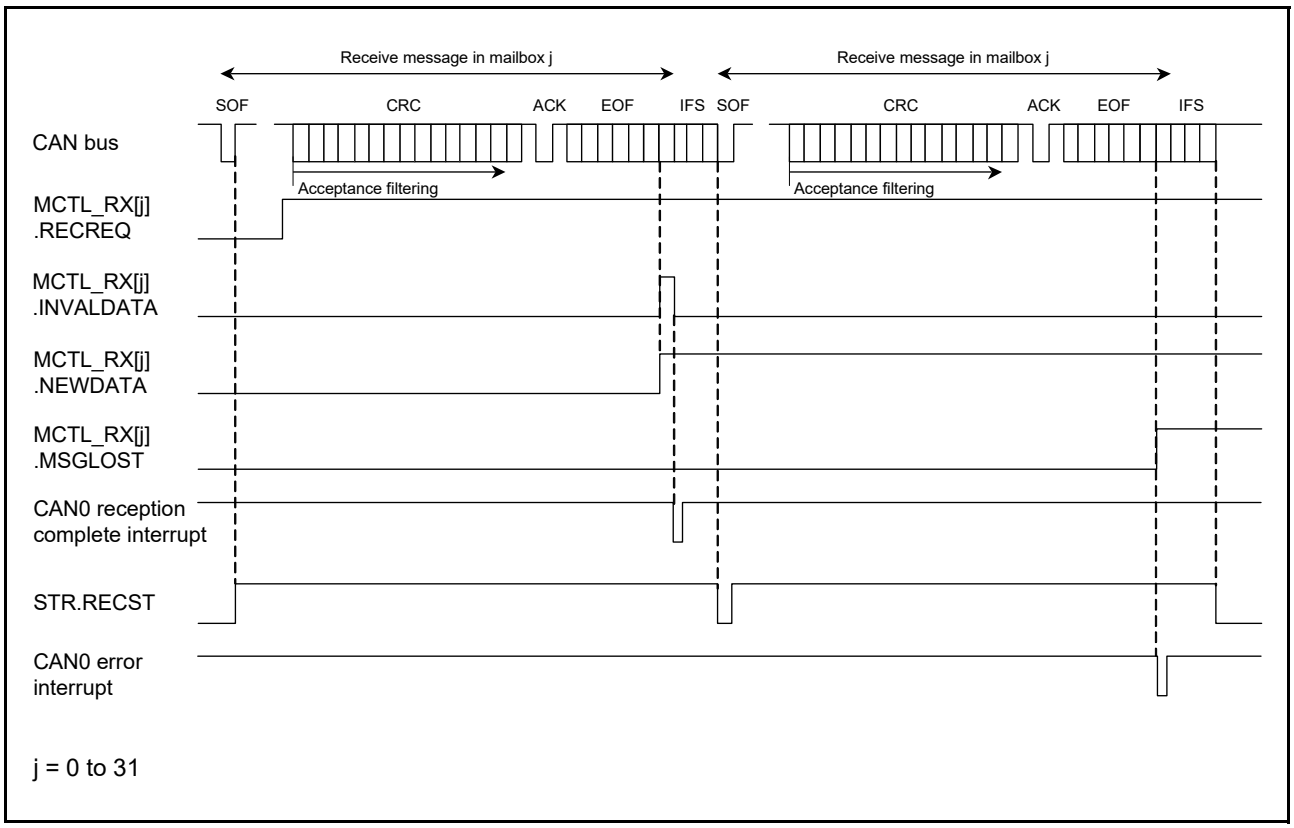


Figure 32.19 Operation example of data frame reception in overrun mode

Steps 1. to 5. are the same as in overwrite mode.

6. In overrun mode, if the next CAN message is received before the NEWDATA bit in MCTL_RXj is set to 0, the MSGLOST bit in MCTL_RXj is set to 1 (message overrun). The new received message is discarded and a CAN0 error interrupt request is generated when the associated interrupt enable bit in the EIER register is 1 (interrupt enabled).

32.7.2 Transmission

Figure 32.20 shows an operation example of data frame transmission.

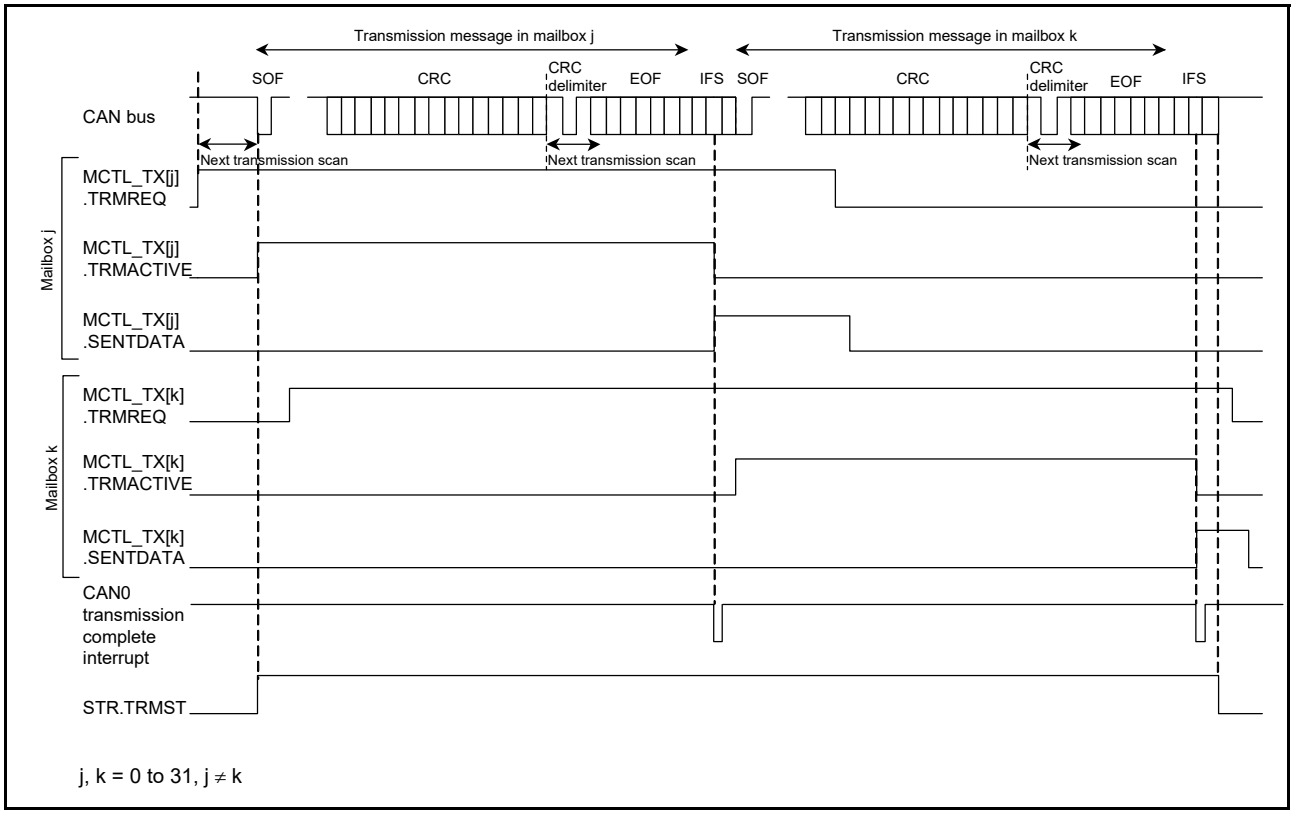


Figure 32.20 Operation example of data frame transmission

1. When a TRMREQ bit in MCTL_TXj (j = 0 to 31) is set to 1 (transmit mailbox) in the bus-idle state, mailbox scanning starts to determine the highest-priority mailbox for transmission. When the transmit mailbox is determined, the TRMACTIVE flag in MCTL_TXj is set to 1 (from acceptance of transmission request to completion of transmission, or until error or arbitration-lost), the STR.TRMST bit is set to 1 (transmission in progress), and the CAN module starts transmission.*1
2. If other TRMREQ bits are set, the transmission scanning starts with the CRC delimiter for the next transmission.
3. If transmission is complete without losing arbitration, the SENTDATA bit in MCTL_TXj is set to 1 (transmission complete) and the TRMACTIVE flag is set to 0 (transmission is pending or transmission is not requested). If the interrupt enable bit in the MIER register is 1 (interrupt enabled), the CAN0 transmission complete interrupt request is generated.
4. When requesting the next transmission from the same mailbox, set bits SENTDATA and TRMREQ to 0, then set the TRMREQ bit to 1 after checking that the SENTDATA and TRMREQ bits are set to 0.

Note 1. If arbitration is lost after the CAN module starts transmission, the TRMACTIVE flag is set to 0. Transmission scanning is performed again to search for the highest-priority transmit mailbox from the beginning of the CRC delimiter. If an error occurs either during transmission or following arbitration-lost, transmission scanning is performed again to search for the highest-priority transmit mailbox from the start of the CRC delimiter.

32.8 Interrupt

The CAN module provides the following interrupts for each channel:

- CAN0 reception complete interrupt for mailboxes 0 to 31 (CAN0_RXM)
- CAN0 transmission complete interrupt for mailboxes 0 to 31 (CAN0_TXM)
- CAN0 receive FIFO interrupt (CAN0_RXF)
- CAN0 transmit FIFO interrupt (CAN0_TXF)
- CAN0 error interrupt (CAN0_ERS).

Eight interrupt sources are available for CAN0 error interrupts. Check the EIFR register to determine the interrupt source:

- Bus error
- Error-warning
- Error-passive
- Bus-off entry
- Bus-off recovery
- Receive overrun
- Overload frame transmission
- Bus lock.

Table 32.11 lists the CAN interrupts.

Table 32.11 CAN interrupts

Module	Interrupt symbol	Interrupt source	Source flag
CAN0	CAN0_ERS	Bus lock detected	EIFR.BLIF
		Overload frame transmission detected	EIFR.OLIF
		Overrun detected	EIFR.ORIF
		Bus-off recovery detected	EIFR.BORIF
		Bus-off entry detected	EIFR.BOEIF
		Error-passive detected	EIFR.EPIF
		Error-warning detected	EIFR.EWIF
		Bus error detected	EIFR.BEIF
	CAN0_RXF	Receive FIFO message received (MIER_FIFO.MB29 = 0)	RFCR.RFUST[2:0]
		Receive FIFO warning (MIER_FIFO.MB29 = 1)	
	CAN0_TXF	Transmit FIFO message transmission completed (MIER_FIFO.MB25 = 0)	TFCR.TFUST[2:0]
		FIFO last message transmission completed (MIER_FIFO.MB25 = 1)	
	CAN0_RXM	Mailbox 0 to 31 message received	MCTL_RX0.NEWDATA to MCTL_RX31.NEWDATA
	CAN0_TXM	Mailbox 0 to 31 message transmission completed	MCTL_TX0.SENTDATA to MCTL_TX31.SENTDATA

32.9 Usage Notes

32.9.1 Settings for Module-Stop State

The Module Stop Control Register B (MSTPCRB) can enable or disable CAN module operation. The CAN module is initially stopped after a reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

32.9.2 Settings for the Operating Clock

- The following clock constraint must be satisfied for the CAN module when the CCLKS bit is 1:

  fPCLKB ≥ fCANMCLK

- The source of the peripheral module clocks must be PLL for the CAN module when the CCLKS bit is 0
- The clock frequency ratio of PCLKA and PCLKB must be 2:1 when using the CAN module. Operation is not guaranteed for other settings.

33. Serial Peripheral Interface (SPI)

33.1 Overview

The MCU provides two independent channels of the Serial Peripheral Interface (SPI). The SPI channels are capable of high-speed, full-duplex synchronous serial communications with multiple processors and peripheral devices. Table 33.1 lists the specifications of the SPI, and Figure 33.1 shows the block diagram.

In this section, PCLK is used to refer to PCLKA. Additionally, *n* indicates A or B, and *i* indicates 0 or 1. A lower-case letter *i* in pin and signal names indicates a value from 0 to 3, and a lower-case letter *m* in SPI command register m (SPCMDm) indicates a value from 0 to 7.

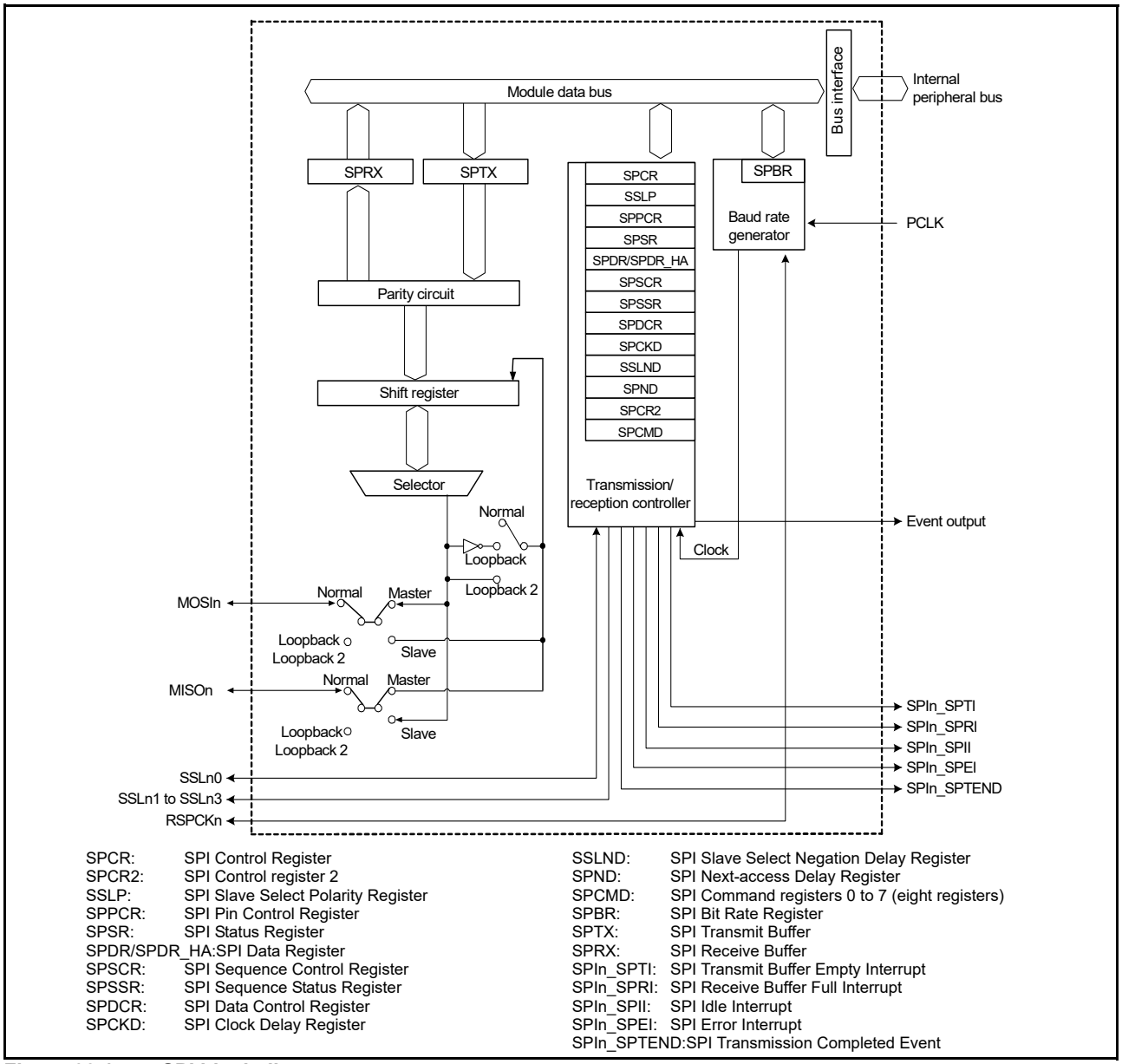
Table 33.1 SPI specifications (1 of 2)

Parameter	Description
Number of channels	Two channels
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 is available • Communication mode: Full-duplex or transmit-only can be selected • Switching of RSPCK polarity • Switching of RSPCK phase.
Data format	• MSB first or LSB first is selectable • Transfer bit length is selectable as 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits • 128-bit transmit and receive buffers • Up to four frames can be transferred in one round of transmission/reception (each frame consisting of up to 32 bits).
Bit rate	• In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing PCLK (the division ratio ranges from divided by 2 to divided by 4096) • In slave mode, the minimum PCLK clock divided by 6 can be input as RSPCK (the maximum frequency of RSPCK is that of PCLK divided by 6). Width at high level: 3 cycles of PCLK Width at low level: 3 cycles of PCLK
Buffer configuration	• Double buffer configuration for the transmit/receive buffers • 128 bits for the transmit/receive buffers.
Error detection	• Mode fault error detection • Underrun error detection • Overrun error detection*1 • Parity error detection.
SSL control function	• Four SSL pins (SSLn0 to SSLn3) for each channel • In single-master mode, SSLn0 to SSLn3 pins for output • In multi-master mode: SSLn0 pin for input and SSLn1 to SSLn3 pins for either output or unused. • In slave mode: SSLn0 pin for input and SSLn1 to SSLn3 pins 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	• A transfer of up to eight commands can be executed sequentially in looped execution. • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity/phase, transfer data length, MSB first/LSB first, burst, RSPCK delay, SSL negation delay, and next-access delay • A transfer can be initiated by writing to the transmit buffer • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function.

Table 33.1 SPI specifications (2 of 2)

Parameter	Description
Interrupt sources	• Receive buffer full interrupt • Transmit buffer empty interrupt • SPI error interrupt (mode fault, overrun, parity error) • SPI idle interrupt (SPI idle) • Transmission-completed interrupt.
Event link function (output)	The following events can be output to the event link controller: • Receive buffer full signal • Transmit buffer empty signal • Mode fault, underrun, overrun, or parity error signal • SPI idle signal • Transmission-completed signal.
Others	• Function for switching between CMOS output and open-drain output • Function for initializing the SPI • Loopback mode.
Module stop function	Module-stop state can be set

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.



SPCR: SPI Control Register
SPCR2: SPI Control register 2
SSLP: SPI Slave Select Polarity Register
SPPCR: SPI Pin Control Register
SPSR: SPI Status Register
SPDR/SPDR_HA:SPI Data Register
SPSCR: SPI Sequence Control Register
SPSSR: SPI Sequence Status Register
SPDCR: SPI Data Control Register
SPCKD: SPI Clock Delay Register
SSLND: SPI Slave Select Negation Delay Register
SPND: SPI Next-access Delay Register
SPCMD: SPI Command registers 0 to 7 (eight registers)
SPBR: SPI Bit Rate Register
SPTX: SPI Transmit Buffer
SPRX: SPI Receive Buffer
SPIn_SPTI: SPI Transmit Buffer Empty Interrupt
SPIn_SPRI: SPI Receive Buffer Full Interrupt
SPIn_SPII: SPI Idle Interrupt
SPIn_SPEI: SPI Error Interrupt
SPIn_SPTEND:SPI Transmission Completed Event

Figure 33.1 SPI block diagram

Table 33.2 lists the I/O pins used in the SPI. The SPI automatically switches the I/O direction of the SSLn0 pin. SSLn0 is an output when the SPI is a single master and 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 33.3.2, Controlling SPI Pins.

Table 33.2 SPI pin configuration

Channel	Pin name	I/O	Function
SPI0	RSPCKA	I/O	Clock I/O
	MOSIA	I/O	Master transmit data I/O
	MISOA	I/O	Slave transmit data I/O
	SSLA0	I/O	Slave selection I/O
	SSLA1	Output	Slave selection output
	SSLA2	Output	Slave selection output
	SSLA3	Output	Slave selection output
SPI1	RSPCKB	I/O	Clock I/O
	MOSIB	I/O	Master transmit data I/O
	MISOB	I/O	Slave transmit data I/O
	SSLB0	I/O	Slave selection I/O
	SSLB1	Output	Slave selection output
	SSLB2	Output	Slave selection output
	SSLB3	Output	Slave selection output

33.2 Register Descriptions

33.2.1 SPI Control Register (SPCR)

Address(es): SPI0.SPCR 4007 2000h, SPI1.SPCR 4007 2100h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SPMS	SPI Mode Select	0: SPI operation (4-wire method) 1: Clock synchronous operation (3-wire method).	R/W
b1	TXMD	Communications Operating Mode Select	0: Full-duplex synchronous serial communications 1: Serial communications consisting of only transmit operations.	R/W
b2	MODFEN	Mode Fault Error Detection Enable	0: Disables the detection of mode fault error 1: Enables the detection of mode fault error.	R/W
b3	MSTR	SPI Master/Slave Mode Select	0: Slave mode 1: Master mode.	R/W
b4	SPEIE	SPI Error Interrupt Enable	0: Disables the generation of SPI error interrupt requests 1: Enables the generation of SPI error interrupt requests.	R/W
b5	SPTIE	Transmit Buffer Empty Interrupt Enable	0: Disables the generation of transmit buffer empty interrupt requests 1: Enables the generation of transmit buffer empty interrupt requests.	R/W
b6	SPE	SPI Function Enable	0: Disables the SPI function 1: Enables the SPI function.	R/W
b7	SPRIE	SPI Receive Buffer Full Interrupt Enable	0: Disables the generation of SPI receive buffer full interrupt requests 1: Enables the generation of SPI receive buffer full interrupt requests.	R/W

If the SPCR.MSTR, SPCR.MODFEN, or SPCR.TXMD bit is changed while the SPCR.SPE bit is 1, do not perform subsequent operations.

SPMS bit (SPI Mode Select)

The SPMS bit selects SPI operation (4-wire method) or clock synchronous operation (3-wire method).

The SSLn0 to SSLn3 pins are not used in clock synchronous operation. The RSPCKn, MOSIn, and MISOn pins handle communications. If clock synchronous operation is in master mode (SPCR.MSTR = 1), the SPCMDm.CPHA bit can be set to either 0 or 1. Set the CPHA bit to 1 if clock synchronous operation is in slave mode (SPCR.MSTR = 0). Do not perform operations if the CPHA bit is set to 0 when clock synchronous operation is in slave mode (SPCR.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 33.3.6, 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 error (see section 33.3.8, Error Detection). In addition, the SPI determines the I/O direction of the SSLn0 to SSLn3 pins based on combinations of the MODFEN and MSTR bit settings (see section 33.3.2, Controlling 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 SSLn0 to SSLn3 pins.

SPEIE bit (SPI Error Interrupt Enable)

The SPEIE bit enables or disables the generation of SPI error interrupt requests when:

- 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.

See section 33.3.8, 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.

A transmit buffer empty interrupt request on transmission start is generated by setting the SPE and SPTIE bits to 1 at the same time or by setting the SPE bit to 1 after setting the SPTIE bit to 1. The interrupt is generated when the SPTIE bit is 1 even if the SPI function is disabled (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 33.3.8, Error Detection.

Setting the SPE bit to 0 disables the SPI function and initializes a part of the module function. For details, see section 33.3.9, Initializing the SPI. In addition, a state change on the SPE bit, from 0 to 1 or 1 to 0, triggers a transmit buffer empty interrupt request.

SPRIE bit (SPI Receive Buffer Full Interrupt Enable)

The SPRIE bit enables or disables the generation of an interrupt request if the SPI detects a receive buffer full write after completing a serial transfer.

33.2.2 SPI Slave Select Polarity Register (SSLP)

Address(es): SPI0.SSLP 4007 2001h, SPI1.SSLP 4007 2101h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SSL0P	SSL0 Signal Polarity Setting	0: SSL0 signal is active-low 1: SSL0 signal is active-high.	R/W
b1	SSL1P	SSL1 Signal Polarity Setting	0: SSL1 signal is active-low 1: SSL1 signal is active-high.	R/W
b2	SSL2P	SSL2 Signal Polarity Setting	0: SSL2 signal is active-low 1: SSL2 signal is active-high.	R/W
b3	SSL3P	SSL3 Signal Polarity Setting	0: SSL3 signal is active-low 1: SSL3 signal is active-high.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the contents of SSLP are changed when the SPCR.SPE bit is 1, do not perform subsequent operations.

33.2.3 SPI Pin Control Register (SPPCR)

Address(es): SPI0.SPPCR 4007 2002h, SPI1.SPPCR 4007 2102h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SPLP	SPI Loopback	0: Normal mode 1: Loopback mode, with data inverted for transmission.	R/W
b1	SPLP2	SPI Loopback 2	0: Normal mode 1: Loopback mode, with data not inverted for transmission.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	MOIFV	MOSI Idle Fixed Value	0: The level output on the MOSIn pin during MOSI idling is defined as low 1: The level output on the MOSIn pin during MOSI idling is defined as high.	R/W
b5	MOIFE	MOSI Idle Value Fixing Enable	0: MOSI output value equals final data from previous transfer 1: MOSI output value equals the value set in the MOIFV bit.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the contents of SPPCR are changed when the SPCR.SPE bit is 1, do not perform subsequent operations.

SPLP bit (SPI Loopback)

The SPLP bit selects the mode of the SPI pins.

When the SPLP 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 input path and output path for the shift register, establishing loopback mode.

SPLP2 bit (SPI Loopback 2)

The SPLP2 bit selects the mode of the SPI pins.

When the SPLP2 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 input path and output path for the shift register (loopback mode).

MOIFV bit (MOSI Idle Fixed Value)

If the MOIFE bit is 1 in master mode, the MOIFV bit determines the MOSIn pin output value during the SSL negation period, including the SSL retention period during a burst transfer.

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, including the SSL retention period during a burst transfer. 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.

33.2.4 SPI Status Register (SPSR)

Address(es): SPI0.SPSR 4007 2003h, SPI1.SPSR 4007 2103h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRF	—	SPTEF	UDRF	PERF	MODF	IDLNF	OVRF
Value after reset:	0	0	1	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	OVRF	Overrun Error Flag	0: No overrun error occurs 1: An overrun error occurs.	R/(W)*1
b1	IDLNF	SPI Idle Flag	0: SPI is in the idle state 1: SPI is in the transfer state.	R
b2	MODF	Mode Fault Error Flag	0: No mode fault error or underrun error occurs 1: A mode fault error or an underrun error occurs.	R/(W)*1
b3	PERF	Parity Error Flag	0: No parity error occurs 1: A parity error occurs.	R/(W)*1
b4	UDRF	Underrun Error Flag	0: A mode fault error occurs (MODF = 1) 1: An underrun error occurs (MODF = 1). This bit is invalid when MODF flag is 0.	R/W*1,*2
b5	SPTEF	SPI Transmit Buffer Empty Flag	0: Data found in the transmit buffer 1: No data in the transmit buffer.	R/(W)*3
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	SPRF	SPI Receive Buffer Full Flag	0: No valid data in SPDR/SPDR_HA 1: Valid data found 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 that you clear 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 = 1) and when the RSPCK clock auto-stop function is enabled (SPCR2.SCKASE = 1), an overrun error does not occur and this flag does not become 1. For details, see section 33.3.8.1, Overrun Errors.

[Setting condition]

- When the next serial transfer ends when the SPCR.TXMD bit is 0 and the receive buffer is full.

[Clearing condition]

- When SPSR is read while the OVRF flag is 1, and then 0 is written to this flag.

IDLNF flag (SPI Idle Flag)

The IDLNF flag indicates the transfer status of the SPI.

[Setting condition]

Master mode

- When conditions 1. and 2. in the master mode [Clearing condition] are not satisfied.

Slave mode

- The SPCR.SPE bit is 1 (SPI function is enabled).

[Clearing condition]

Master mode

- When either condition 1. is satisfied or conditions 2., 3., and 4. are satisfied.

1. The SPCR.SPE bit is 0, for SPI initialization.
2. The transmit buffer (SPTX) is empty, meaning data for the next transfer is not set.

3. The SPSSR.SPCP[2:0] bits are 000b, indicating the beginning of sequence control.
4. The SPI internal sequencer is in the idle state, indicating that operations up to the next-access delay are complete.

Slave mode

- When the SPCR.SPE bit is 0, for SPI initialization.

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), the SPI detects a mode fault error.

Slave mode

- When either condition 1. or 2. is satisfied.
  1. The SSLni pin is negated before the RSPCK cycle necessary 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.
  2. The serial transfer begins with the SPCR.MSTR bit set to 0 (slave mode), the SPCR.SPE bit 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 when the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1, triggering a parity error.

[Clearing condition]

- When SPSR is read when the PERF 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 set to 0 (slave mode), the SPCR.SPE bit set to 1, and the transmission data not prepared, triggering an underrun error.

[Clearing condition]

- When SPSR is read when the UDRF flag is 1, 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]

- The following 1. is satisfied or the following condition 2. is satisfied.
  1. The SPCR.SPE bit is 0, for SPI initialization.
  2. Transmit data is transferred from the transmit buffer to the shift register.

[Clearing condition]

- Written data to SPDR/SPDR_HA equals the number of frames set by the number of frames specification bits (SPFC[1:0]) in the SPI Data Control Register (SPDCR).

Data can be written to SPDR/SPDR_HA only when the SPTEF bit is 1. If data is written to the transmit buffer of SPDR/SPDR_HA when the SPTEF bit is 0, the 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 conditions]

- When a serial transfer ends while the communication operating mode select bit (TXMD) in the SPI Control Register (SPCR) is 0 and the SPRF bit is 0, the SPI transfers the receive data from the shift register to SPDR/SPDR_HA.
- When the OVRF flag is 1, however, SPRF is not changed from 0 into 1.

[Clearing condition]

- When received data is read from SPDR/SPDR_HA.

33.2.5 SPI Data Register (SPDR/SPDR_HA)

Address(es): SPI0.SPDR 4007 2004h, SPI1.SPDR 4007 2104h

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

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

Address(es): SPI0.SPDR_HA 4007 2004h, SPI1.SPDR_HA 4007 2104h

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

SPDR/SPDR_HA is the interface with the buffers that hold data for transmission and reception by the SPI.

When accessing in words (SPLW = 1), access SPDR. When accessing in halfwords (SPLW = 0), access SPDR_HA.

The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to SPDR/SPDR_HA. Figure 33.2 shows the configuration of SPDR/SPDR_HA.

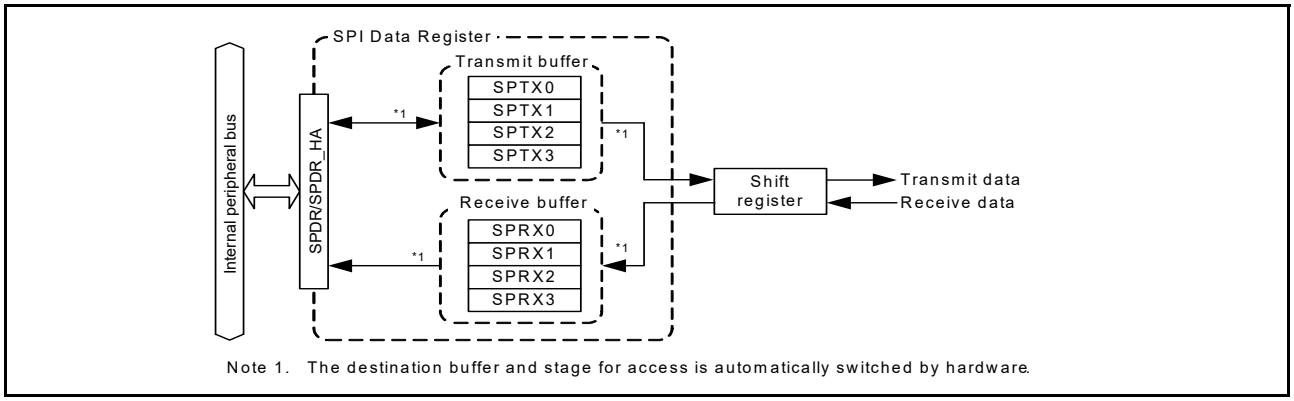


Figure 33.2 Configuration of SPDR/SPDR_HA

The transmit and receive buffers each have four stages. The number of stages is selectable by the number of frames specification bits in the SPI Data Control Register (SPDCR.SPFC[1:0]). The eight stages of the buffer are all mapped to the single address of SPDR/SPDR_HA.

Data written to SPDR/SPDR_HA is written to a transmit-buffer stage (SPTXn) (n = 0 to 3) 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.

If the data length is not 32 bits, bits not referred to in SPTXn (n = 0 to 3) are stored in the associated bits in SPRXn (n = 0 to 3). For example, if the data length is 9 bits, received data are 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 is the interface with 32-bit wide transmit and receive buffers, each of which has four stages, for a total of 32 bytes. The 32 bytes are mapped to the 4-byte address space for SPDR/SPDR_HA. The unit of access for SPDR/SPDR_HA is selected in the SPI word access/halfword access specification bit in the SPI Data Control Register (SPDCR.SPLW).

Flush data for transmission at the LSB end of the register and store received data at the LSB end.

This section describes operations involved in writing to and reading from SPDR/SPDR_HA.

(a) Writing

Data written to SPDR/SPDR_HA 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.

The transmit buffer includes a transmit buffer write pointer that is automatically updated to reference the next stage each time data is written to SPDR/SPDR_HA.

Figure 33.3 shows the configuration of the bus interface with the transmit buffer, for writes to SPDR.

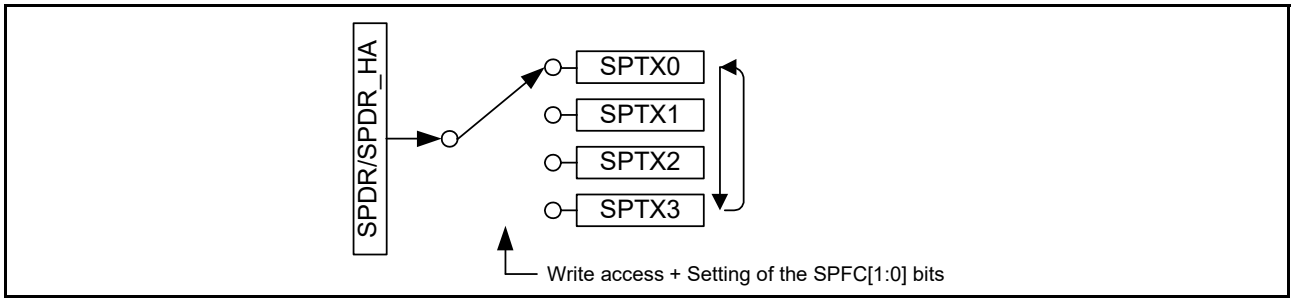


Figure 33.3 Configuration of SPDR/SPDR_HA for write access

The sequence for switching the transmit buffer write pointer changes with the setting of the number of frames specification bits in the SPI Data Control Register (SPDCR.SPFC[1:0]).

- Settings of the SPFC[1:0] bits and sequence of switching the pointer from SPTX0 to SPTX3.
  When the SPFC[1:0] bits are 00b: SPTX0 → SPTX0 → SPTX0 → …
  When the SPFC[1:0] bits are 01b: SPTX0 → SPTX1 → SPTX0 → SPTX1 → …
  When the SPFC[1:0] bits are 10b: SPTX0 → SPTX1 → SPTX2 → SPTX0 → SPTX1 → …
  When the SPFC[1:0] bits are 11b: SPTX0 → SPTX1 → SPTX2 → SPTX3 → SPTX0 → SPTX1 → …

When 1 is written to the SPI function enable bit in the SPI Control Register (SPCR.SPE) while the bit is 0, SPTX0 is the destination for the next write.

When writing to the transmit buffer (SPTXn) after generation of the transmit buffer empty interrupt (SPSR.SPTEF = 1), write the number of frames set in the number of frames specification bits (SPFC[1:0]) in the SPI Data Control Register (SPDCR). The value of the buffer is not updated after completion of the writing and before the generation of the next transmit buffer empty interrupt (SPSR.SPTEF = 0) even when the number of frames is written to the transmit buffer (SPTXn).

(b) Reading

SPDR/SPDR_HA can be accessed to read the value of receive buffer (SPRXn) or transmit buffer (SPTXn). The setting of the SPI receive or transmit data select bit in the SPI Data Control Register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer.

The sequence of reading the SPDR/SPDR_HA register is controlled by independent pointers, receive buffer read pointer, and transmit buffer read pointer.

Figure 33.4 shows the configuration of a bus interface with the receive and transmit buffers for reading from SPDR/SPDR_HA.

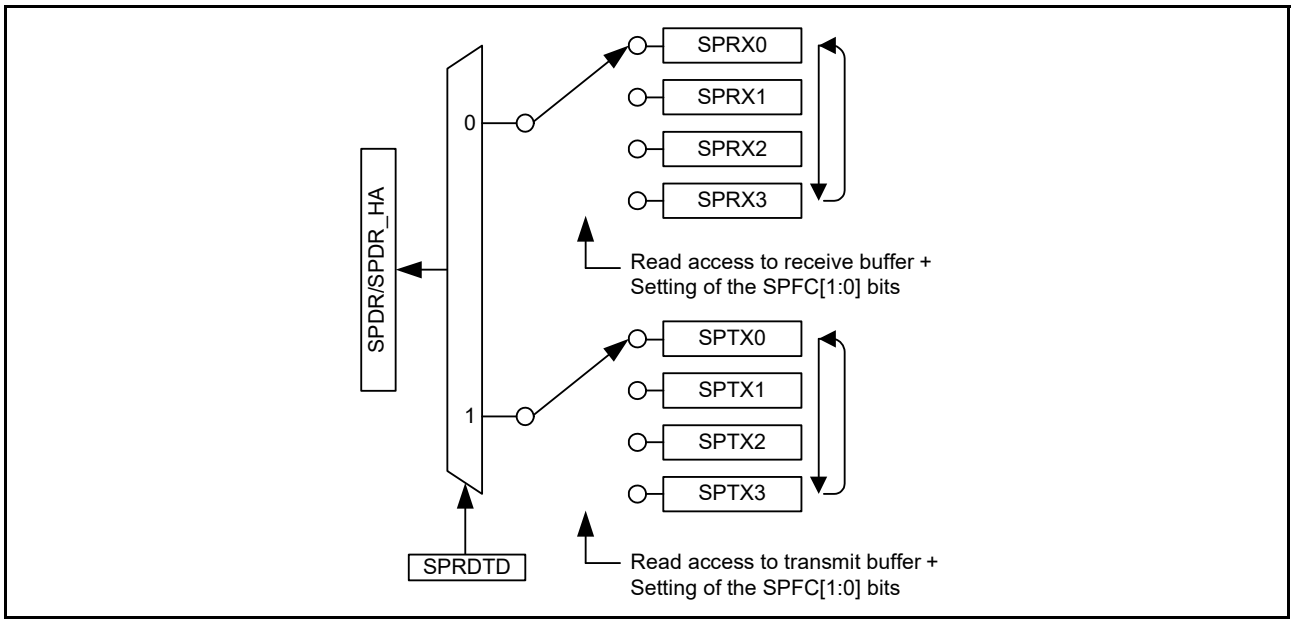


Figure 33.4 Configuration of SPDR/SPDR_HA for read access

Reading the receive buffer switches the receive buffer read pointer to the next buffer automatically.

The sequence of switching the receive buffer read pointer is the same as that for the transmit buffer write pointer.

However, when 1 is written to the SPI function enable bit in the SPI Control Register (SPCR.SPE) while the bit is 1, SPRX0 is referenced by the buffer read pointer for the next read.

The transmit buffer read pointer is updated when writing to SPDR/SPDR_HA, but not when reading from the transmit buffer. When reading from the transmit buffer, the value most recently written to SPDR/SPDR_HA is read. However, after the transmit buffer empty interrupt is generated, and when the transmit buffer becomes full again (the number of frames of data specified in the (SPDCR.SPFC[1:0] bits are written to the transmit buffer), reading from the transmit buffer returns all 0s until the next transmit buffer empty interrupt is generated.

33.2.6 SPI Sequence Control Register (SPSCR)

Address(es): SPI0.SPSCR 4007 2008h, SPI1.SPSCR 4007 2108h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPSLN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	SPSLN[2:0]	SPI Sequence Length Specification	b2 b0 Sequence Length — Referenced SPCMD0 to SPCMD7 (No.) 0 0 0: 1 — 0→0→… 0 0 1: 2 — 0→1→0→… 0 1 0: 3 — 0→1→2→0→… 0 1 1: 4 — 0→1→2→3→0→… 1 0 0: 5 — 0→1→2→3→4→0→… 1 0 1: 6 — 0→1→2→3→4→5→0→… 1 1 0: 7 — 0→1→2→3→4→5→6→0→… 1 1 1: 8 — 0→1→2→3→4→5→6→7→0→… The order in which the SPCMD0 to SPCMD07 registers are referenced is changed based on the sequence length that is set in these bits. The relationship between the setting of these bits, sequence length, and SPCMD0 to SPCMD7 registers referenced by the SPI is shown. However, the SPI in slave mode references SPCMD0.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPSCR register sets the sequence length when the SPI operates in master mode. When changing the SPSCR.SPSLN[2:0] bits while both the SPCR.MSTR and SPCR.SPE bits are 1, always check that the SPSR.IDLNF flag is 0.

SPSLN[2:0] bits (SPI Sequence Length Specification)

The SPSLN[2:0] bits specify a sequence length when the SPI in master mode performs sequential operations. The SPI in master mode changes SPCMD0 to SPCMD7 registers to be referenced and the order in which they are referenced is based on the sequence length that is set in the SPSLN[2:0] bits. In slave mode, SPCMD0 is referenced.

33.2.7 SPI Sequence Status Register (SPSSR)

Address(es): SPI0.SPSSR 4007 2009h, SPI1.SPSSR 4007 2109h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SPECM[2:0]			—	SPCP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	SPCP[2:0]	SPI Command Pointer	b2 b0 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7.	R
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	SPECM[2:0]	SPI Error Command	b6 b4 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7.	R
b7	—	Reserved	This bit is read as 0.	R

The SPSSR register indicates the sequence control status when the SPI operates in master mode. Any writing to SPSSR is ignored.

SPCP[2:0] bits (SPI Command Pointer)

The SPCP[2:0] bits indicate the SPCMDm register that is referenced to by the pointer during sequence control by the SPI. For the SPI sequence control, see section 33.3.10.1, Master Mode Operation.

SPECM[2:0] bits (SPI Error Command)

The SPECM[2:0] bits indicate the SPCMDm register that is specified by the SPCP[2:0] bits when an error is detected during sequence control by the SPI. The SPI updates the SPECM[2:0] bits only when an error is detected. If both the SPSR.OVRF and SPSR.MODF flags are 0 and there is no error, the values of the SPECM[2:0] bits have no meaning.

For the SPI error detection function, see section 33.3.8, Error Detection. For the SPI sequence control, see section 33.3.10.1, Master Mode Operation.

33.2.8 SPI Bit Rate Register (SPBR)

Address(es): SPI0.SPBR 4007 200Ah, SPI1.SPBR 4007 210Ah

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

The SPBR register sets the bit rate in master mode. If the contents of the SPBR register are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, do not perform subsequent operations.

When the SPI is used in slave mode, the bit rate depends on the bit rate of the input clock regardless of the settings of SPBR and the SPCMDm.BRDV[1:0] bits (bit rate division setting bits). Use bit rates that satisfy the electrical characteristics.

The bit rate is determined by the combination of the SPBR setting and the SPCMDm.BRDV[1:0] bit setting. The equation for calculating the bit rate is given as follows:

$$\text{Bit rate} = \frac{f(\text{PCLK})}{2 \times (n + 1) \times 2^{N}}$$

In the equation, n denotes an SPBR setting (0, 1, 2, …, 255), and N denotes a BRDV[1:0] bit setting (0, 1, 2, 3).

Table 33.3 lists examples of the relationship between the SPBR settings, the BRDV[1:0] settings, and the bit rates.

Table 33.3 Relationship between SPBR settings, BRDV[1:0] settings, and bit rates

SPBR (n)	BRDV[1:0] bits (N)	Division ratio	Bit rate	
			PCLK = 32 MHz	PCLK = 48 MHz
0	0	2	16.0 Mbps	-
1	0	4	8.00 Mbps	12.00 Mbps
2	0	6	5.33 Mbps	8.00 Mbps
3	0	8	4.00 Mbps	6.00 Mbps
4	0	10	3.20 Mbps	4.80 Mbps
5	0	12	2.67 Mbps	4.00 Mbps
5	1	24	1.33 Mbps	2.00 Mbps
5	2	48	667 kbps	1.00 Mbps
5	3	96	333 kbps	500 kbps
255	3	4096	7.81 kbps	11.7 kbps

33.2.9 SPI Data Control Register (SPDCR)

Address(es): SPI0.SPDCR 4007 200Bh, SPI1.SPDCR 4007 210Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	SPLW	SPRDTD	—	—	SPFC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	SPFC[1:0]	Number of Frames Specification	b1 b0 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SPRDTD	SPI Receive/Transmit Data Select	0: SPDR/SPDR_HA values are read from the receive buffer 1: SPDR/SPDR_HA values are read from the transmit buffer (but only if the transmit buffer is empty).	R/W
b5	SPLW	SPI Word Access/Halfword Access Specification	0: SPDR_HA is valid to access in halfwords 1: SPDR is valid (to access in words).	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Up to four frames can be transmitted or received in one round of transmission or reception. The amount of data in each transfer is controlled by the combination of the SPCMDm.SPB[3:0] bits, the SPSCR.SPSLN[2:0] bits, and the SPDCR.SPFC[1:0] bits.

When changing the SPDCR.SPFC[1:0] bits while the SPCR.SPE bit is 1, always check that the SPSR.IDLNF flag is 0.

SPFC[1:0] bits (Number of Frames Specification)

The SPFC[1:0] bits specify the number of frames that can be stored in SPDR/SPDR_HA (per transfer activation). Up to four frames can be transmitted or received in one round of transmission or reception.

When the transmission data of the number of frames specified by the SPFC[1:0] bits are written to the SPDR/SPDR_HA register, the SPI clears the SPSR.SPTEF flag to 0 and begins transmitting. After that, when the transmission data of the number of frames specified by the SPFC[1:0] bits are transmitted to the shift register, the SPI generates the transmission buffer empty interrupt (SPSR.SPTEF is set to 1).

When the data of the number of frames specified by the SPFC[1:0] bits is received, the SPI generates the receive buffer full interrupt (SPSR.SPRF is set to 1).

Table 33.4 Settable combinations of SPSLN[2:0] bits and SPFC[1:0] bits

Setting	SPSLN[2:0]	SPFC[1:0]	Number of frames in a single sequence	Number of frames at which transmission or reception buffer is filled
1-1	000b	00b	1	1
1-2	000b	01b	2	2
1-3	000b	10b	3	3
1-4	000b	11b	4	4
2-1	001b	01b	2	2
2-2	001b	11b	4	4
3	010b	10b	3	3
4	011b	11b	4	4
5	100b	00b	5	1
6	101b	00b	6	1
7	110b	00b	7	1
8	111b	00b	8	1

SPRDTD bit (SPI Receive/Transmit Data Select)

The SPRDTD bit selects whether the SPDR/SPDR_HA read 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. Reading the transmit buffer must take place before the writing of the number of frames set in the SPFC[1:0] bits is finished and after generation of the transmit buffer empty interrupt (SPSR.SPTEF is 1).

For details, see section 33.2.5, SPI Data Register (SPDR/SPDR_HA).

SPLW bit (SPI Word Access/Halfword Access Specification)

The SPLW bit specifies the access width for the SPDR register. Access to SPDR_HA in halfwords is valid when the SPLW bit is 0 and access to the SPDR register in words is valid when the SPLW bit is 1. In addition, when the SPLW bit is 0, set the SPCMDm.SPB[3:0] bits (SPI data length setting bits) from 8 to 16 bits. When 20, 24, or 32 bits is specified, do not perform any operations.

33.2.10 SPI Clock Delay Register (SPCKD)

Address(es): SPI0.SPCKD 4007 200Ch, SPI1.SPCKD 4007 210Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	SCKDL[2:0]	RSPCK Delay Setting	b2 b0 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
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPCKD register sets a period from the beginning of SSLni signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMDm.SCKDEN bit is 1. If the contents of the SPCKD register are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, do not perform subsequent operations.

SCKDL[2:0] bits (RSPCK Delay Setting)

The SCKDL[2:0] bits set an RSPCK delay value when the SPCMDm.SCKDEN bit is 1. When using the SPI in slave mode, set the SCKDL[2:0] bits to 000b.

33.2.11 SPI Slave Select Negation Delay Register (SSLND)

Address(es): SPI0.SSLND 4007 200Dh, SPI1.SSLND 4007 210Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SLNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	SLNDL[2:0]	SSL Negation Delay Setting	b2 b0 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
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SSLND register sets a period (SSL negation delay) 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 the SSLND register 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 set an SSL negation delay value when the SPI is in master mode. When using the SPI in slave mode, set the SLNDL[2:0] bits to 000b.

33.2.12 SPI Next-Access Delay Register (SPND)

Address(es): SPI0.SPND 4007 200Eh, SPI1.SPND 4007 210Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	SPNDL[2:0]	SPI Next-Access Delay Setting	b2 b0 0 0 0: 1 RSPCK + 2 PCLK 0 0 1: 2 RSPCK + 2 PCLK 0 1 0: 3 RSPCK + 2 PCLK 0 1 1: 4 RSPCK + 2 PCLK 1 0 0: 5 RSPCK + 2 PCLK 1 0 1: 6 RSPCK + 2 PCLK 1 1 0: 7 RSPCK + 2 PCLK 1 1 1: 8 RSPCK + 2 PCLK.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPND register sets a non-active period (next-access delay) of the SSLni signal after termination of a serial transfer when the SPCMDm.SPNDEN bit is 1. If the contents of the SPND register are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, do not perform subsequent operations.

SPNDL[2:0] bits (SPI Next-Access Delay Setting)

The SPNDL[2:0] bits set a next-access delay when the SPCMDm.SPNDEN bit is 1. When using the SPI in slave mode, set the SPNDL[2:0] bits to 000b.

33.2.13 SPI Control Register 2 (SPCR2)

Address(es): SPI0.SPCR2 4007 200Fh, SPI1.SPCR2 4007 210Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SCKASE	PTE	SPIIE	SPOE	SPPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	SPPE	Parity Enable	0: No parity bit added to transmit data and parity bit in receive data not checked 1: (When SPCR.TXMD = 0) Parity bit added to transmit data and parity bit of receive data checked (When SPCR.TXMD = 1) Parity bit added to transmit data but parity bit of receive data not checked.	R/W
b1	SPOE	Parity Mode	0: Even parity selected for transmission and reception 1: Odd parity selected for transmission and reception.	R/W
b2	SPIIE	SPI Idle Interrupt Enable	0: Disables the generation of idle interrupt requests 1: Enables the generation of idle interrupt requests.	R/W
b3	PTE	Parity Self-Testing	0: Self-diagnosis function of the parity circuit disabled 1: Self-diagnosis function of the parity circuit enabled.	R/W
b4	SCKASE	RSPCK Auto-Stop Function Enable	0: RSPCK auto-stop function disabled 1: RSPCK auto-stop function enabled.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the SPPE, SPOE, or SCKASE bit in the SPCR2 register is changed while the SPCR.SPE bit is 1, do not perform subsequent operations.

SPPE bit (Parity Enable)

The SPPE bit enables or disables the parity function.

The parity bit is added to transmit data, and parity checking is performed for receive data when the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1. The parity bit is added to transmit data but parity checking is not performed for receive data when the SPCR.TXMD bit is 1 and the SPCR2.SPPE bit is 1.

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/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/receive character plus the parity bit is odd.

The SPOE bit is valid only 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 the SPI being in the idle state is detected and the SPSR.IDLNF flag is set to 0.

PTE bit (Parity Self-Testing)

The PTE bit enables the self-diagnosis function of the parity circuit in order 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 33.3.8.1, Overrun Errors.

33.2.14 SPI Command Registers 0 to 7 (SPCMD0 to SPCMD7)

Address(es): SPI0.SPCMD0 4007 2010h, SPI0.SPCMD1 4007 2012h, SPI0.SPCMD2 4007 2014h, SPI0.SPCMD3 4007 2016h, SPI0.SPCMD4 4007 2018h, SPI0.SPCMD5 4007 201Ah, SPI0.SPCMD6 4007 201Ch, SPI0.SPCMD7 4007 201Eh, SPI1.SPCMD0 4007 2110h, SPI1.SPCMD1 4007 2112h, SPI1.SPCMD2 4007 2114h, SPI1.SPCMD3 4007 2116h, SPI1.SPCMD4 4007 2118h, SPI1.SPCMD5 4007 211Ah, SPI1.SPCMD6 4007 211Ch, SPI1.SPCMD7 4007 211Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SCKDEN	SLNDEN	SPNDEN	LSBF	SPB[3:0]				SSLKP	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	Bit name	Description	R/W
b0	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
b1	CPOL	RSPCK Polarity Setting	0: RSPCK low when idle 1: RSPCK high when idle.	R/W
b3, b2	BRDV[1:0]	Bit Rate Division Setting	b3 b2 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
b6 to b4	SSLA[2:0]	SSL Signal Assertion Setting	b6 b4 0 0 0: SSL0 0 0 1: SSL1 0 1 0: SSL2 0 1 1: SSL3 1 x x: Setting prohibited x: Don't care.	R/W
b7	SSLKP	SSL Signal Level Keeping	0: Negates all SSL signals on completion of transfer 1: Keeps the SSL signal level from the end of transfer until the beginning of the next access.	R/W
b11 to b8	SPB[3:0]	SPI Data Length Setting	b11 b8 0100 to 0111: 8 bits 1 0 0 0: 9 bits 1 0 0 1: 10 bits 1 0 1 0: 11 bits 1 0 1 1: 12 bits 1 1 0 0: 13 bits 1 1 0 1: 14 bits 1 1 1 0: 15 bits 1 1 1 1: 16 bits 0 0 0 0: 20 bits 0 0 0 1: 24 bits 0010, 0011: 32 bits.	R/W
b12	LSBF	SPI LSB First	0: MSB first 1: LSB first.	R/W
b13	SPNDEN	SPI Next-Access Delay Enable	0: Next-access delay is 1 RSPCK + 2 PCLK 1: Next-access delay equals to the setting in the SPI Next-access Delay register (SPND).	R/W
b14	SLNDEN	SSL Negation Delay Setting Enable	0: An SSL negation delay of 1 RSPCK 1: An SSL negation delay equal to the setting of the SPI Slave Select Negation Delay register (SSLND).	R/W
b15	SCKDEN	RSPCK Delay Setting Enable	0: An RSPCK delay of 1 RSPCK 1: An RSPCK delay equals the setting in the SPI Clock Delay register (SPCKD).	R/W

The SPCMDm register sets a transfer format for the SPI in master mode. Each channel has 8 SPI command registers (SPCMD0 to SPCMD7). Some of the bits in the SPCMD0 register are used to set a transfer mode for the SPI in slave mode. The SPI in master mode sequentially references the SPCMDm register according to the settings in the SPSCR.SPSLN[2:0] bits and executes the serial transfer that is set in the referenced SPCMDm register.

Set the SPCMDm 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 the SPCMDm register is referenced.

The SPCMDm register that the SPI in master mode references can be checked with the SPSSR.SPCP[2:0] bits. If the contents of the SPCMDm register are changed while the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1, do not perform subsequent operations.

CPHA bit (RSPCK Phase Setting)

The CPHA bit sets the RSPCK phase for the SPI in master mode 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 sets the RSPCK polarity for the SPI in master mode 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 combination with the settings in the BRDV[1:0] bits and SPBR. See section 33.2.8, SPI Bit Rate Register (SPBR). The SPBR settings determine the base bit rate. The BRDV[1:0] settings select a bit rate that is obtained by dividing the base bit rate by 1, 2, 4, or 8. In the SPCMDm register, different BRDV[1:0] bit settings can be specified, enabling execution of serial transfers at a different bit rate 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 set value 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.

SSLKP bit (SSL Signal Level Keeping)

When the SPI in master mode performs a serial transfer, the SSLKP bit specifies whether the SSLni signal level for the current command is to be kept or negated between the SSL negation timing associated with the current command and the SSL assertion timing associated with the next command.

Setting the SSLKP bit to 1 enables a burst transfer. For details, see section 33.3.10.1, Master Mode Operation (4) Burst Transfer. When using the SPI in slave mode, set the SSLKP bit to 0.

SPB[3:0] bits (SPI Data Length Setting)

The SPB[3:0] bits set a transfer data length for the SPI in master mode or slave mode.

When the SPLW bit is 0, set the SPCMDm.SPB[3:0] bits (SPI data length setting bits) from 8 to 16 bits.

LSBF bit (SPI LSB First)

The LSBF bit sets the data format of the SPI in master mode or slave mode to MSB first or LSB first.

SPNDEN bit (SPI Next-Access Delay Enable)

The SPNDEN bit sets 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 (next-access delay). If the SPNDEN bit is 0, the SPI sets the next-access delay to 1 RSPCK + 2 PCLK. 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 sets the period from the time the SPI in master mode stops RSPCK oscillation until the SPI sets the SSLni signal inactive (SSL negation delay). 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 an 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 sets the period from the point when the SPI in master mode activates the SSLni signal until the RSPCK starts oscillation (SPI clock delay). 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.

33.3 Operation

In this section, the *serial transfer period* means a period from the beginning of driving valid data to the fetching of the final valid data.

33.3.1 Overview of SPI Operations

The SPI is capable of synchronous serial transfers in slave mode (SPI operation), single-master mode (SPI operation), multi-master mode (SPI operation), slave mode (clock synchronous operation), and master mode (clock synchronous operation). The mode is selectable in the MSTR, MODFEN, and SPMS bits in SPCR. Table 33.5 lists the relationship between SPI modes and SPCR settings, and a description of each mode.

Table 33.5 Relationship between SPI modes and SPCR settings and description of each mode

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 signal	Input	Output	Output/Hi-Z	Input	Output
MOSIn signal	Input	Output	Output/Hi-Z	Input	Output
MISOn signal	Output/Hi-Z	Input	Input	Output	Input
SSLn0 signal	Input	Output	Input	Hi-Z*1	Hi-Z*1
SSLn1 to SSLn3 signals	Hi-Z*1	Output	Output/Hi-Z	Hi-Z*1	Hi-Z*1
SSL polarity change function	Supported	Supported	Supported	-	-
Transfer rate	Up to PCLK/6	Up to PCLK/2	Up to PCLK/2	Up to PCLK/6	Up to PCLK/2
Clock source	RSPCKn input	On-chip baud rate generator	On-chip baud rate generator	RSPCKn 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				
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0,1)	Possible (CPHA = 0,1)	-	-
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 activation method	SSL input active or RSPCK oscillation	Transmit buffer is written to at generation of a transmit buffer empty interrupt request (SPTEF is 1)	Transmit buffer is written to at generation of a transmit buffer empty interrupt request (SPTEF is 1)	RSPCK oscillation	Transmit buffer is written to at generation of a transmit buffer empty interrupt request (SPTEF is 1)
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported				
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*2,*3				
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.

33.3.2 Controlling SPI Pins

Based on the MSTR, MODFEN, and SPMS bit settings in SPCR and the PmnPFS.NCODR bit for I/O Ports, the SPI can switch pin states. Table 33.6 lists the relationship between pin states and bit settings. Setting the PmnPFS.NCODR bit for an I/O port to 0 selects CMOS output, setting it to 1 selects open-drain output. The I/O port settings must follow this relationship.

Table 33.6 Relationship between pin states and bit settings

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 to SSLn3	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 to SSLn3*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 to SSLn3*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 to SSLn3*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
	SSLn0 to SSLn3*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 the multiplex pins for which the SPI function is not selected.
Note 3. When SSLn0 is at the active level, the pin state is Hi-Z.
Note 4. When SSLn0 is at the non-active level or the SPCR.SPE bit is 0, the pin state is Hi-Z.
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 MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) based on the MOIFE and MOIFV bit settings in SPPCR, as listed in Table 33.7.

Table 33.7 MOSI signal value determination during SSL negation period

MOIFE bit	MOIFV bit	MOSIn signal value during SSL negation period
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

33.3.3 SPI System Configuration Examples

33.3.3.1 Single Master and Single Slave with the MCU Configured as a Master

Figure 33.5 shows a single-master and single-slave SPI system configuration example where the MCU is the master. In the single-master and single-slave configuration, the SSLn0 to SSLn3 output of the MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave stays selected.*1

The MCU (master) drives the RSPCKn and MOSIn signals. The SPI slave drives the MISO signals.

Note 1. In the transfer format used when SPCMDm.CPHA is 0, the SSL signal for some slave devices cannot be fixed to the active level. In this case, always connect the SSLni output of the MCU to the SSL input of the slave device.

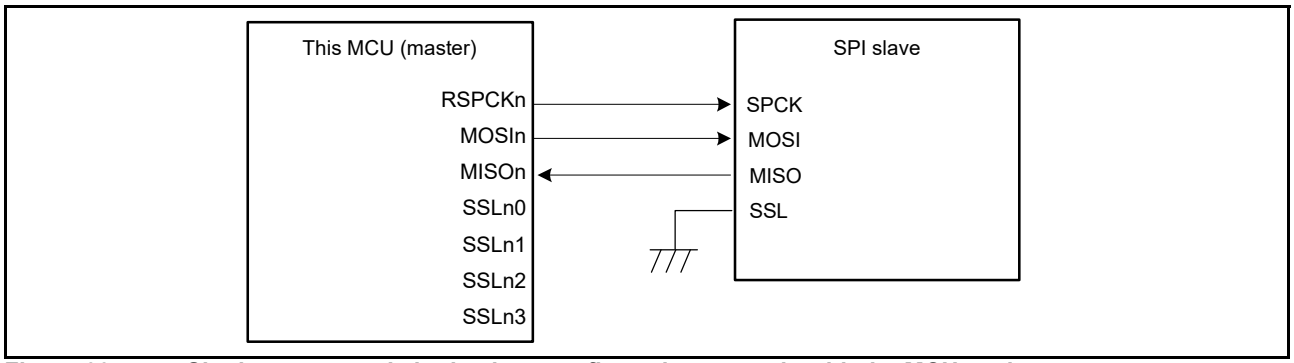


Figure 33.5 Single-master and single-slave configuration example with the MCU as the master

33.3.3.2 Single Master and Single Slave with the MCU Configured as a Slave

Figure 33.6 shows a single-master and single-slave SPI system configuration example where the MCU is the slave. When the MCU is to operate 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 signals.*1

In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, the SSLn0 input of the MCU (slave) is fixed to the low level, the MCU (slave) stays selected. This enables serial transfer (Figure 33.7).

Note 1. When SSLn0 is at the non-active level, the pin state is Hi-Z.

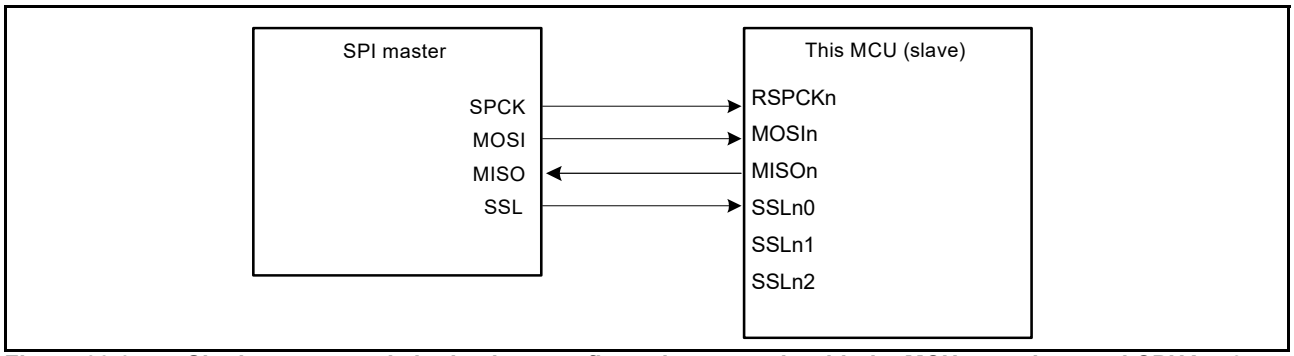


Figure 33.6 Single-master and single-slave configuration example with the MCU as a slave and CPHA = 0

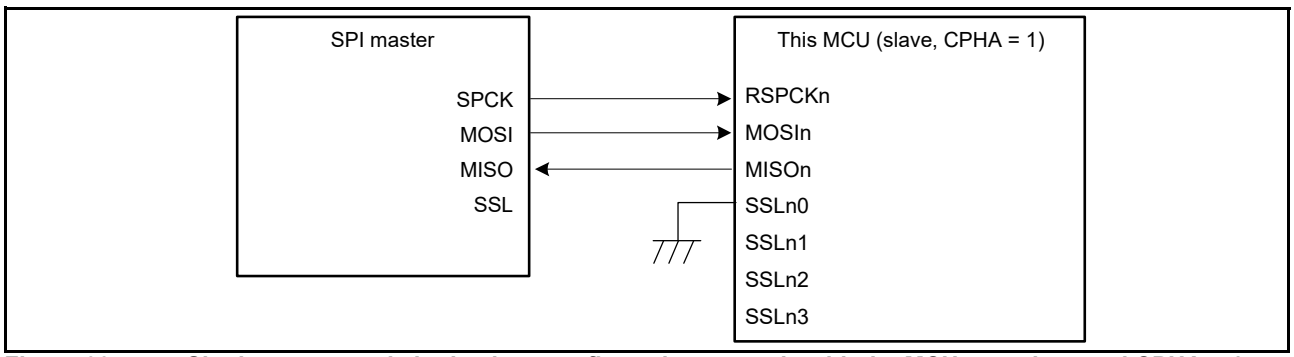


Figure 33.7 Single-master and single-slave configuration example with the MCU as a slave and CPHA = 1

33.3.3.3 Single Master and Multi-Slave with the MCU configured as a Master

Figure 33.8 shows a single-master and multi-slave SPI system configuration example where the MCU is the master. In the example of Figure 33.8, the SPI system comprises the MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCKn and MOSIn outputs of the MCU (master) are connected to the RSPCK and MOSI inputs of SPI slaves 0 to 3. The MISO outputs of SPI slaves 0 to 3 are all connected to the MISOn input of the MCU (master). SSLn0 to SSLn3 outputs of the MCU (master) are connected to the SSL inputs of SPI slave 0 to SPI slave 3, respectively.

The MCU (master) drives RSPCKn, MOSIn, and SSLn0 to SSLn3. Of the SPI slaves 0 to 3, the slave that receives low-level input into the SSL input drives the MISO signal.

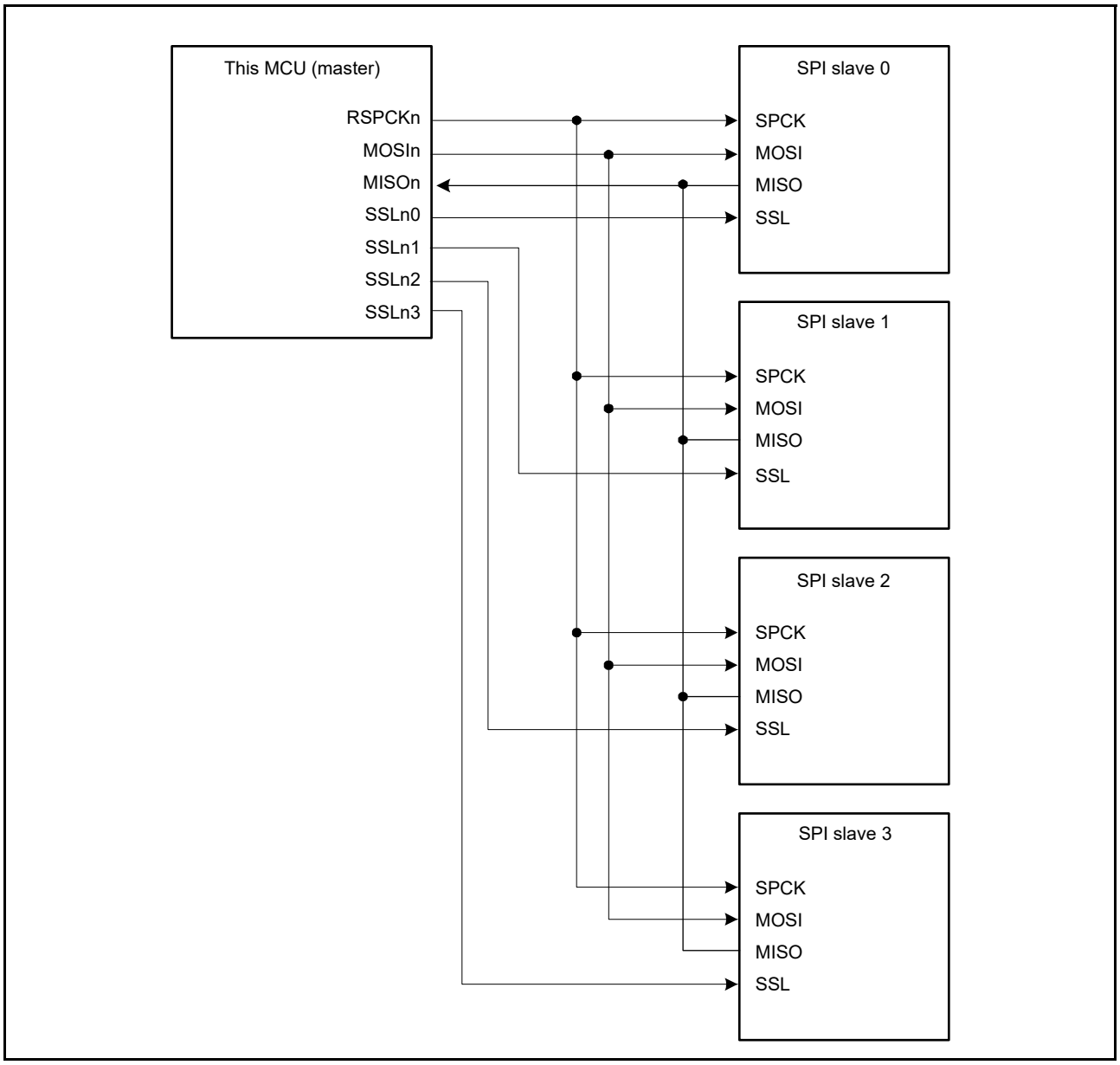


Figure 33.8 Single-master and multi-slave configuration example with the MCU as a master

33.3.3.4 Single Master and Multi-Slave with the MCU Configured as a Slave

Figure 33.9 shows a single-master and multi-slave SPI system configuration example where the MCU is a slave. In the example of Figure 33.9, the SPI system comprises an SPI master and two MCUs (slave X and slave Y).

The SPCK and MOSI outputs of the SPI master are connected to the RSPCKn and MOSIn inputs of the MCUs (slave X and slave Y). The MISOn outputs of the MCUs (slave X and slave Y) are all connected to the MISO input of the SPI master. SSLX and SSLY outputs of the SPI master are connected to the SSLn0 inputs of the MCUs (slave X and slave Y), respectively.

The SPI master drives SPCK, MOSI, SSLX, and SSLY. The MCU slave (X or Y) that receives low-level input into the SSLn0 input drives MISOn.

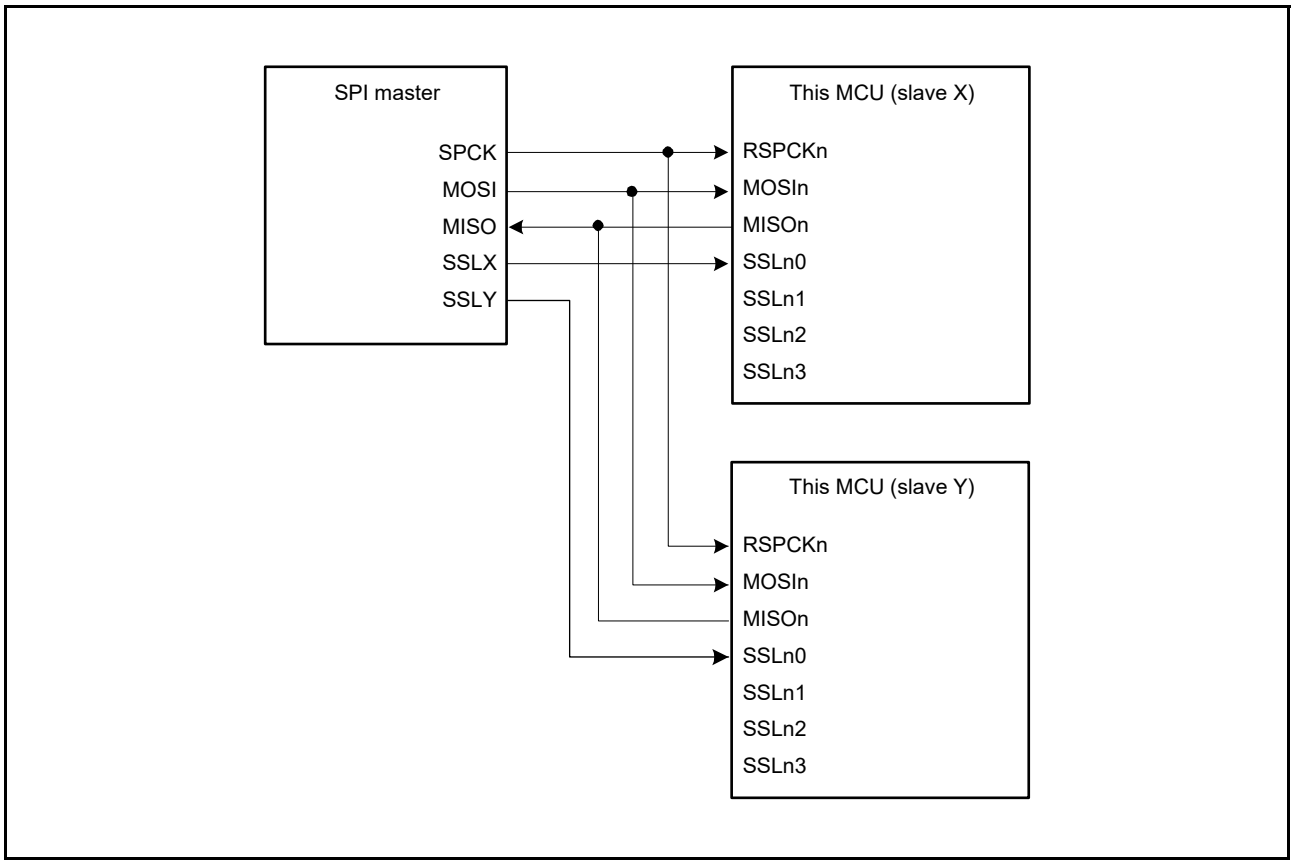


Figure 33.9 Single-master and multi-slave configuration example when the MCU is a slave

33.3.3.5 Multi-Master and Multi-Slave with the MCU Configured as a Master

Figure 33.10 shows a multi-master and multi-slave SPI system configuration example where the MCU is a master. In the example of Figure 33.10, the SPI system comprises two MCUs (master X and master Y) and two SPI slaves (SPI slave 1 and SPI slave 2).

The RSPCKn and MOSIn outputs of the MCUs (master X and master Y) are connected to the RSPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISOn inputs of the MCUs (master X and master Y). Any generic port Y output from the MCU (master X) is connected to the SSLn0 input of the MCU (master Y). Any generic port X output of the MCU (master Y) is connected to the SSLn0 input of the MCU (master X). The SSLn1 and SSLn2 outputs of the MCUs (master X and master Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, because the system can be comprised solely of SSLn0 input, and SSLn1 and SSLn2 outputs for slave connections, the SSLn3 output of the MCU is not required.

The MCU drives RSPCKn, MOSIn, SSLn1, and SSLn2 when the SSLn0 input level is high. When the SSLn0 input level is low, the MCU detects a mode fault error, sets RSPCKn, MOSIn, SSLn1, and SSLn2 to Hi-Z, and releases the SPI bus directly to the other master. The SPI slave 1 or 2 that receives low-level input into the SSL input drives MISO.

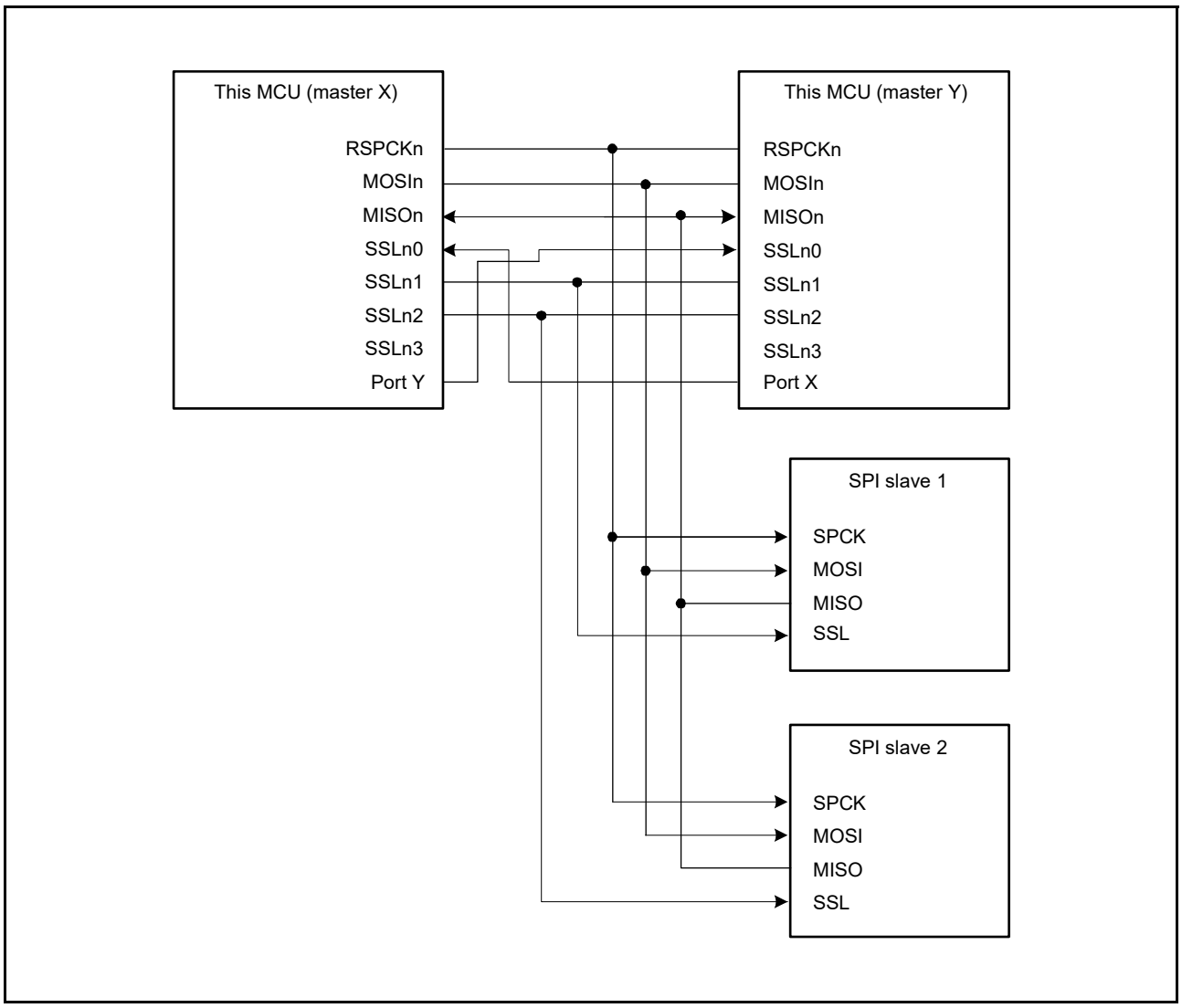


Figure 33.10 Multi-master and multi-slave configuration example with the MCU as a master

33.3.3.6 Master and Slave in Clock Synchronous Mode with the MCU Configured as a Master

Figure 33.11 shows a master and slave clock synchronous mode configuration where the MCU is a master. In the master and slave clock synchronous mode, SSLn0 to SSLn3 of the MCU (master) are not used.

The MCU (master) drives the RSPCKn and MOSIn signals. The SPI slave drives the MISO signal.

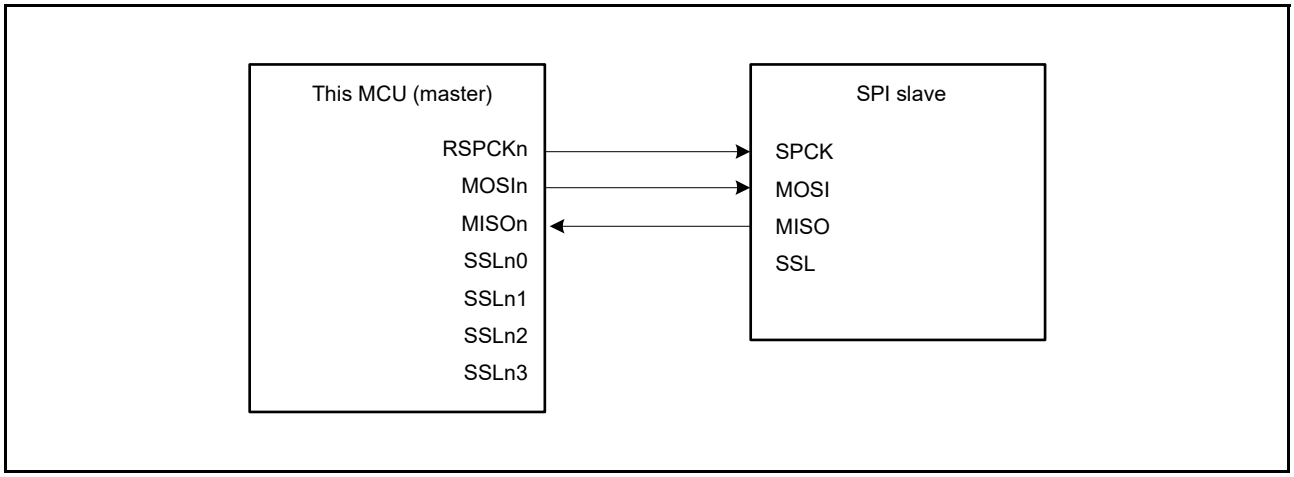


Figure 33.11 Configuration example of master and slave in clock synchronous mode with the MCU as a master

33.3.3.7 Master and Slave in Clock Synchronous Mode with the MCU Configured as a Slave

Figure 33.12 shows a master and slave in clock synchronous mode configuration where the MCU is a slave. When the MCU is to operate as a slave in clock synchronous mode, the MCU (slave) drives the MISOn signal and the SPI master drives the SPCK and MOSI signals. In addition, SSLn0 to SSLn3 of the MCU (slave) are not used.

The MCU (slave) can only execute serial transfer in the single-slave configuration when SPCMDm.CPHA is set to 1.

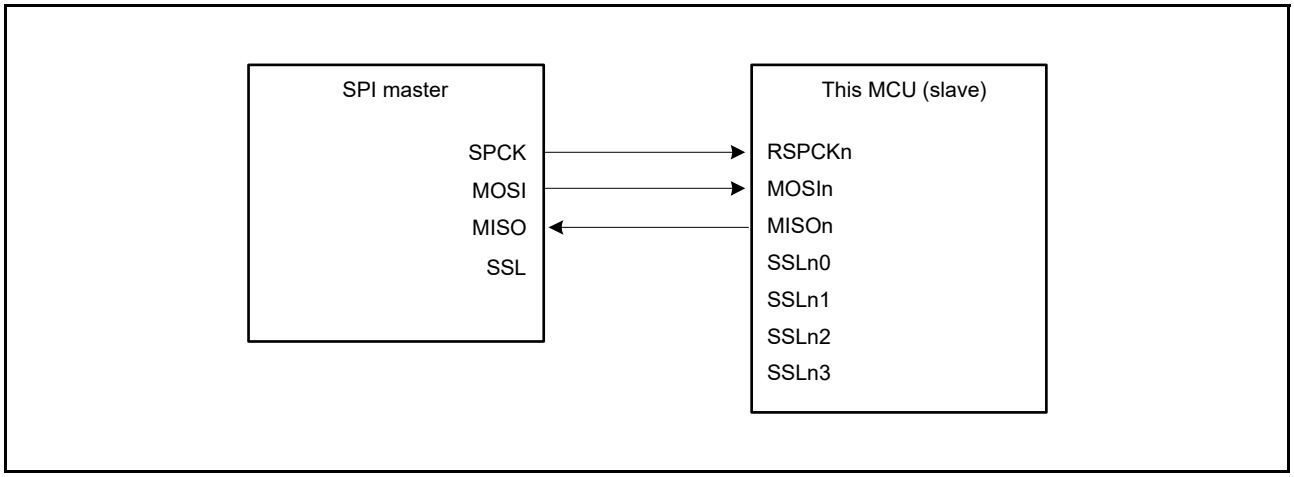


Figure 33.12 Configuration example of master and slave in clock synchronous mode with the MCU as a slave and CPHA = 1

33.3.4 Data Format

The data format of the SPI depends on the settings in the SPI command register m (SPCMDm) (m = 0 to 7) and the parity enable bit in the SPI Control Register 2 (SPCR2.SPPE). Regardless of whether the ordering is MSB first or LSB 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.

(a) 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 bits in the SPI command register m (SPCMDm.SPB[3:0]).

(b) 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 bits in SPI command register m (SPCMDm.SPB[3:0]). In this case, however, the last bit is a parity bit.

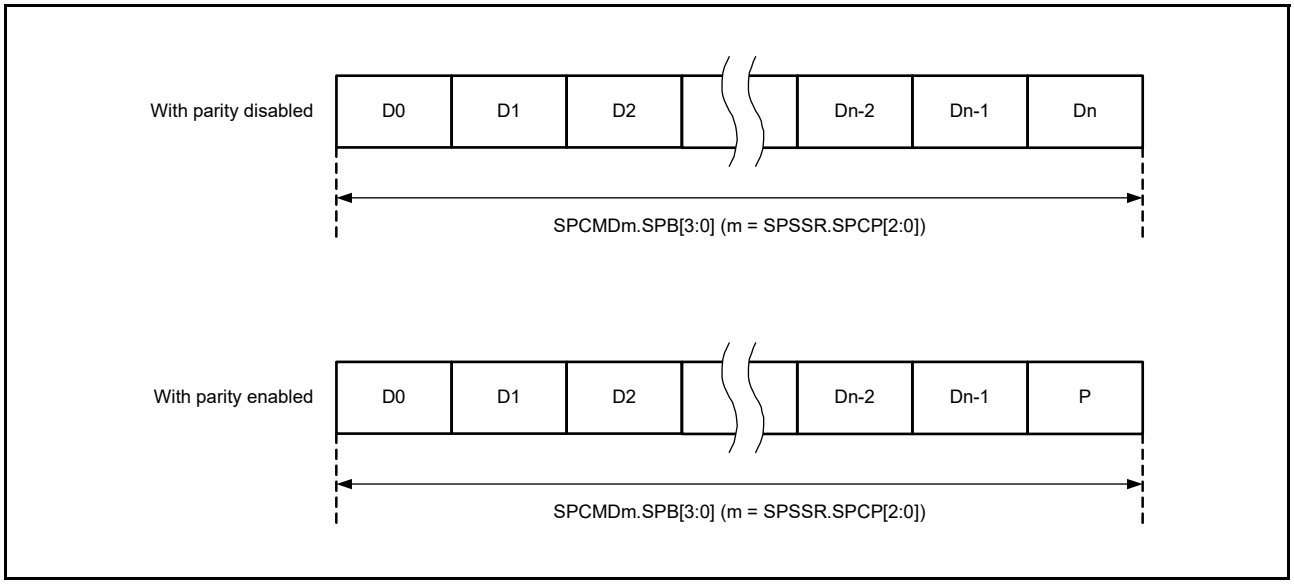


Figure 33.13 Data format with parity disabled and enabled

33.3.4.1 When Parity is Disabled (SPCR2.SPPE = 0)

When parity is disabled, data for transmission is copied to the shift register with no prior 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-first or LSB-first order and data length.

(1) MSB-First Transfer with 32-Bit Data

Figure 33.14 shows operations by the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an 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 cycles of RSPCK, the value in the shift register is copied to the receive buffer.

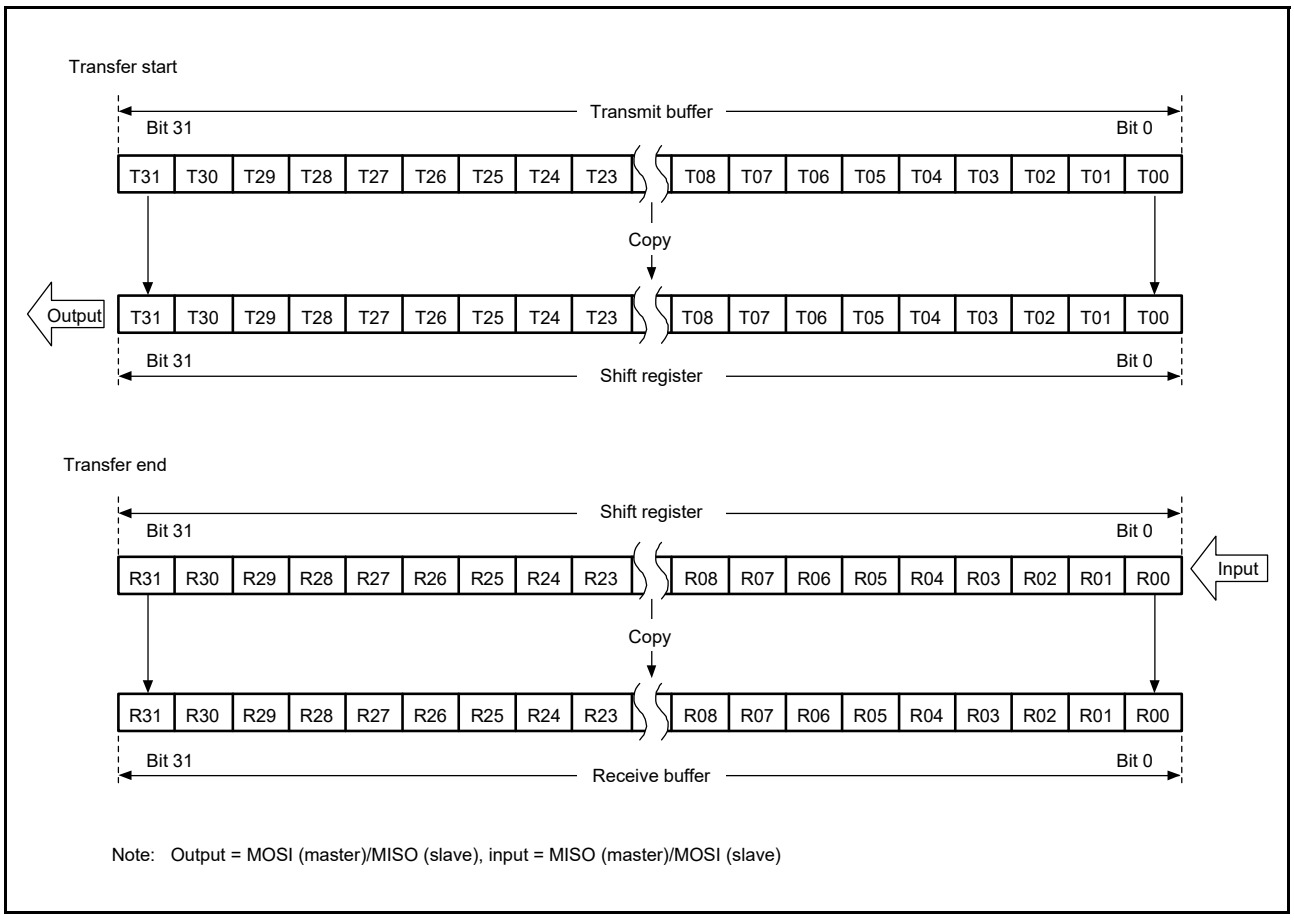


Figure 33.14 MSB-first transfer with 32-bit data and parity disabled

(2) MSB-First Transfer with 24-Bit Data

Figure 33.15 shows operations by the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, 24 bits as the SPI data length 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 bits R23 to R00 are collected after input of the required number of cycles of RSPCK, 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 at the time of transmission leads to 0 being inserted in the upper 8 bits of the receive buffer.

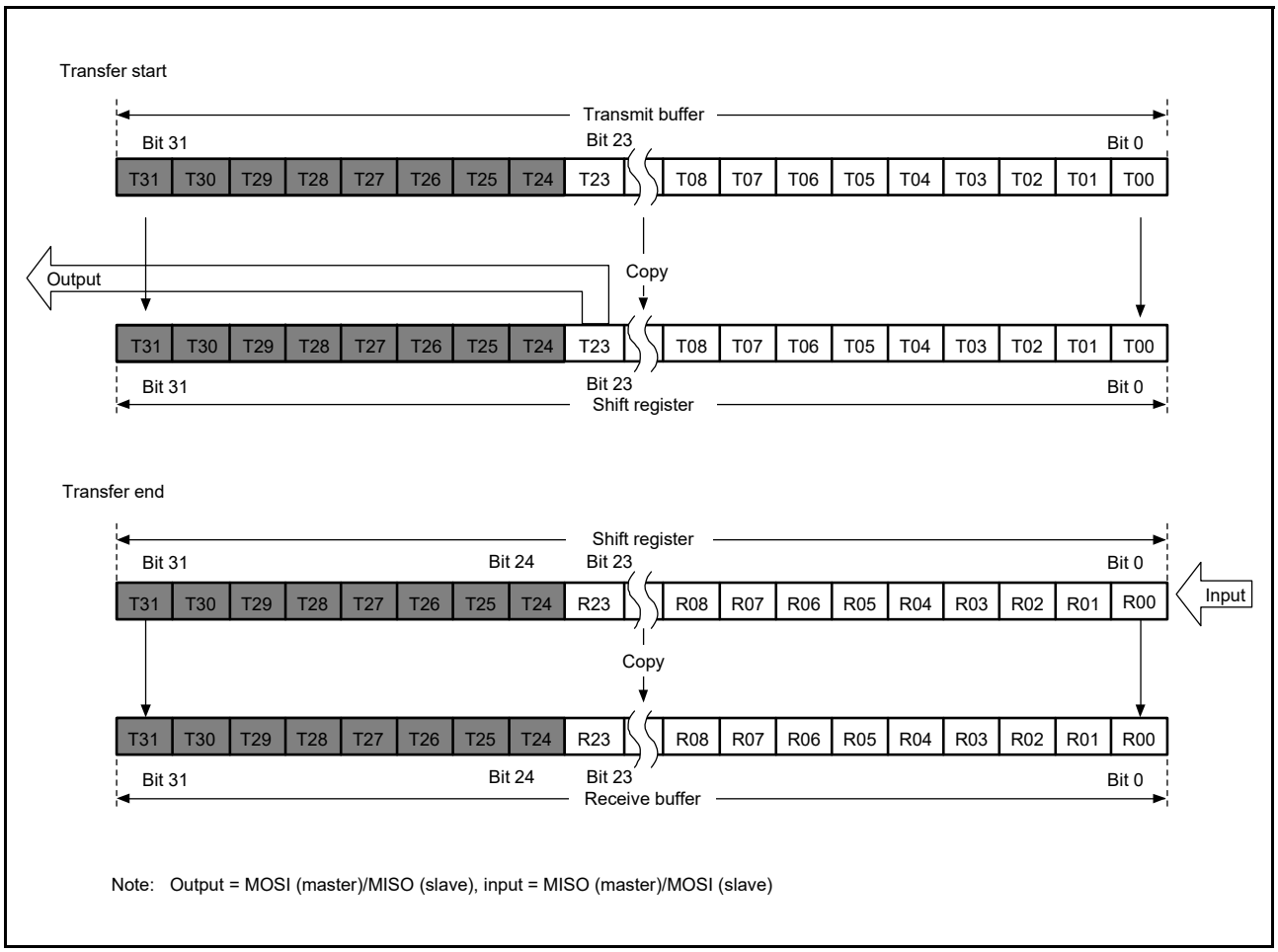


Figure 33.15 MSB-first transfer with 24-bit data and parity disabled

(3) LSB-First Transfer with 32-Bit Data

Figure 33.16 shows operations by 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 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 bits R00 to R31 are collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

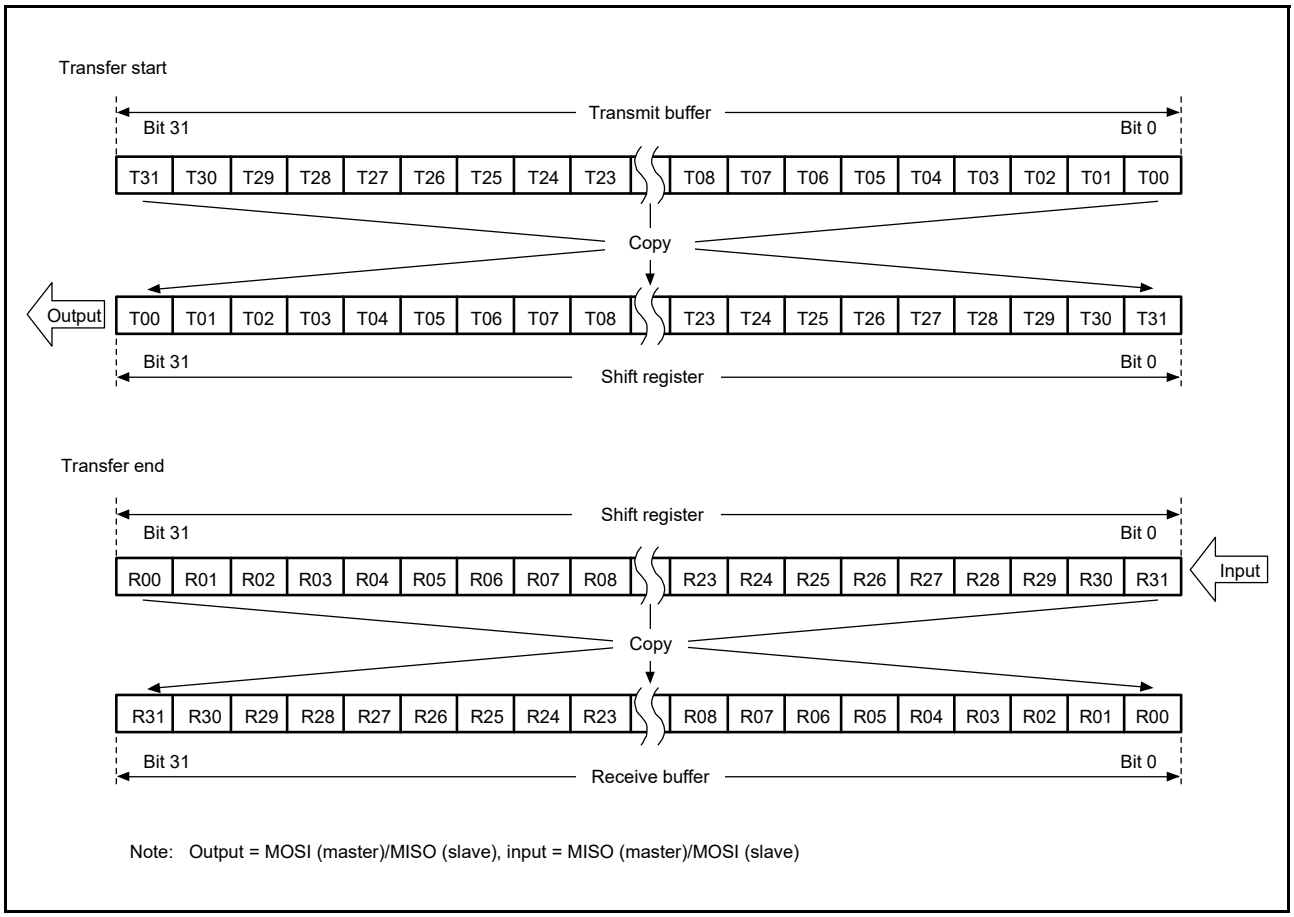


Figure 33.16 LSB-first transfer with 32-bit data and parity disabled

(4) LSB-First Transfer with 24-Bit Data

Figure 33.17 shows operations by the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, 24 bits as the SPI data length for an example that is not 32 bits, 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 bits R00 to R23 are collected after input of the required number of cycles of RSPCK, 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.

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 at the time of transmission leads to 0 being inserted in the upper 8 bits of the receive buffer.

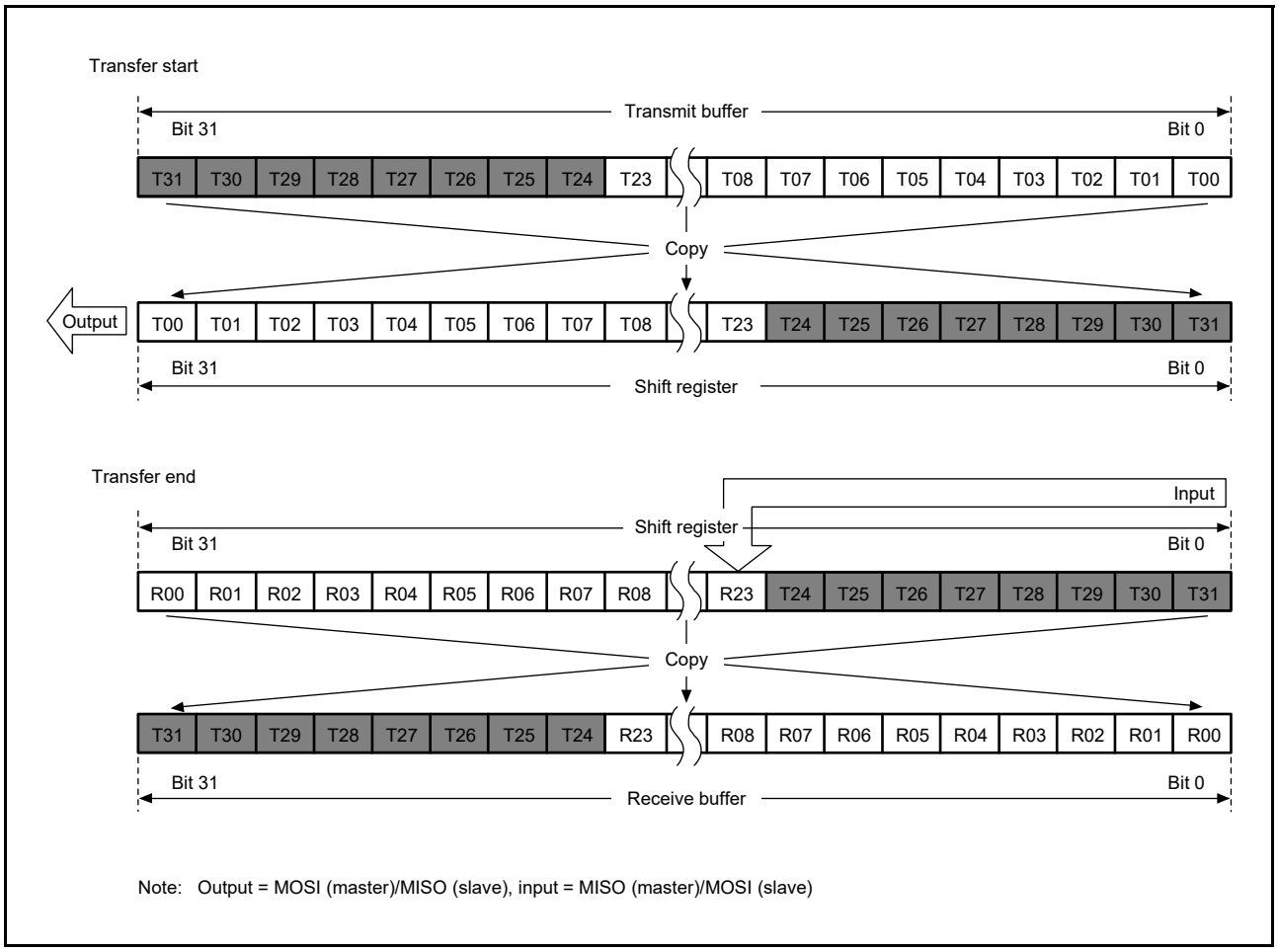


Figure 33.17 LSB-first transfer with 24-bit data and parity disabled

33.3.4.2 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 33.18 shows operations by 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 from T31, T30, …, T01, and P.

In reception, received data is shifted in bit by bit through bit 0 of the shift register. When bits R31 to P are collected after input of the required number of cycles of RSPCK, 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.

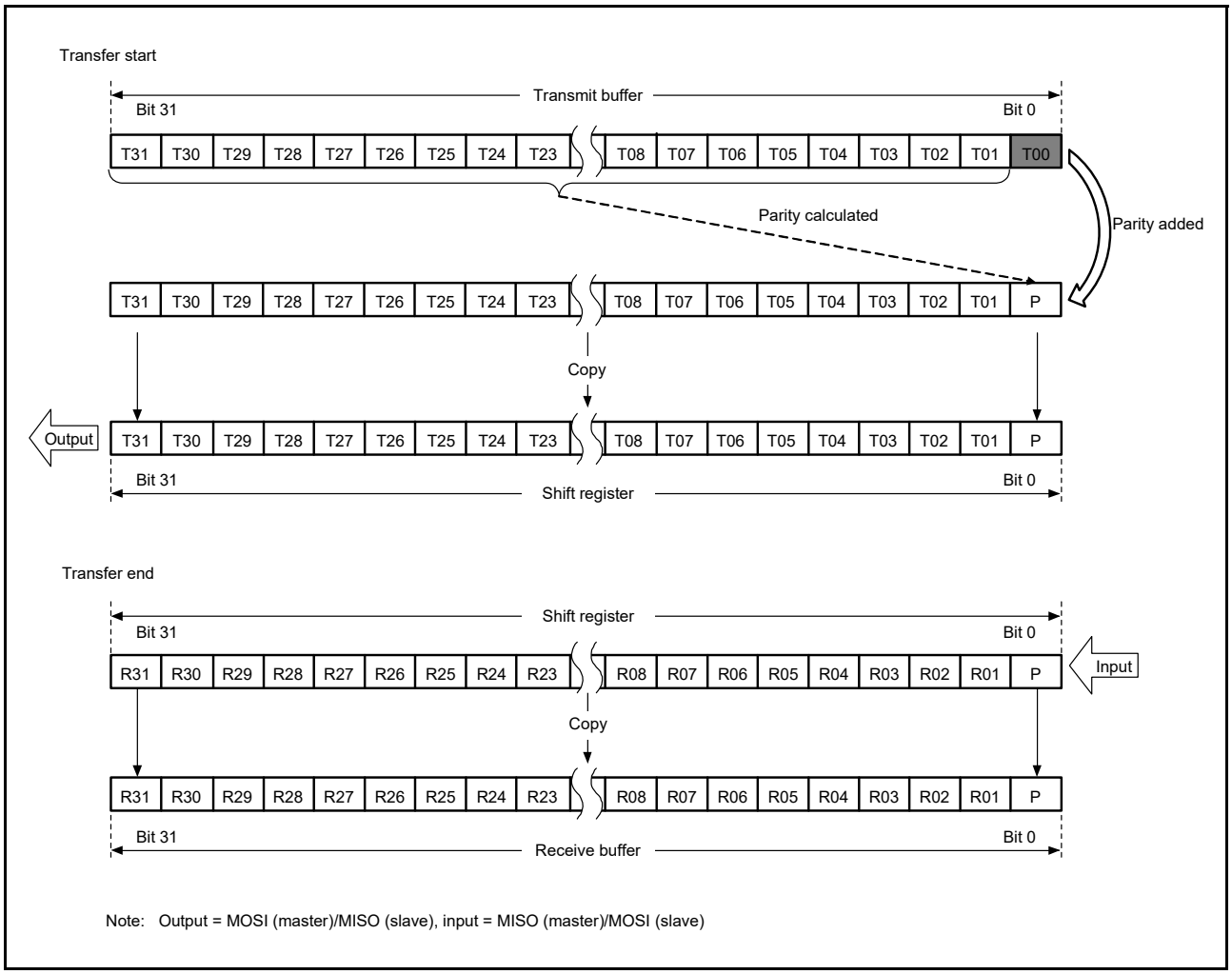


Figure 33.18 MSB-first transfer with 32-bit data and parity enabled

(2) MSB-First Transfer with 24-Bit Data

Figure 33.19 shows operations by the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, 24 bits as the SPI data length for an example that is not 32 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 bits R23 to P are collected after input of the required number of cycles of RSPCK, 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 are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the upper 8 bits of the receive buffer.

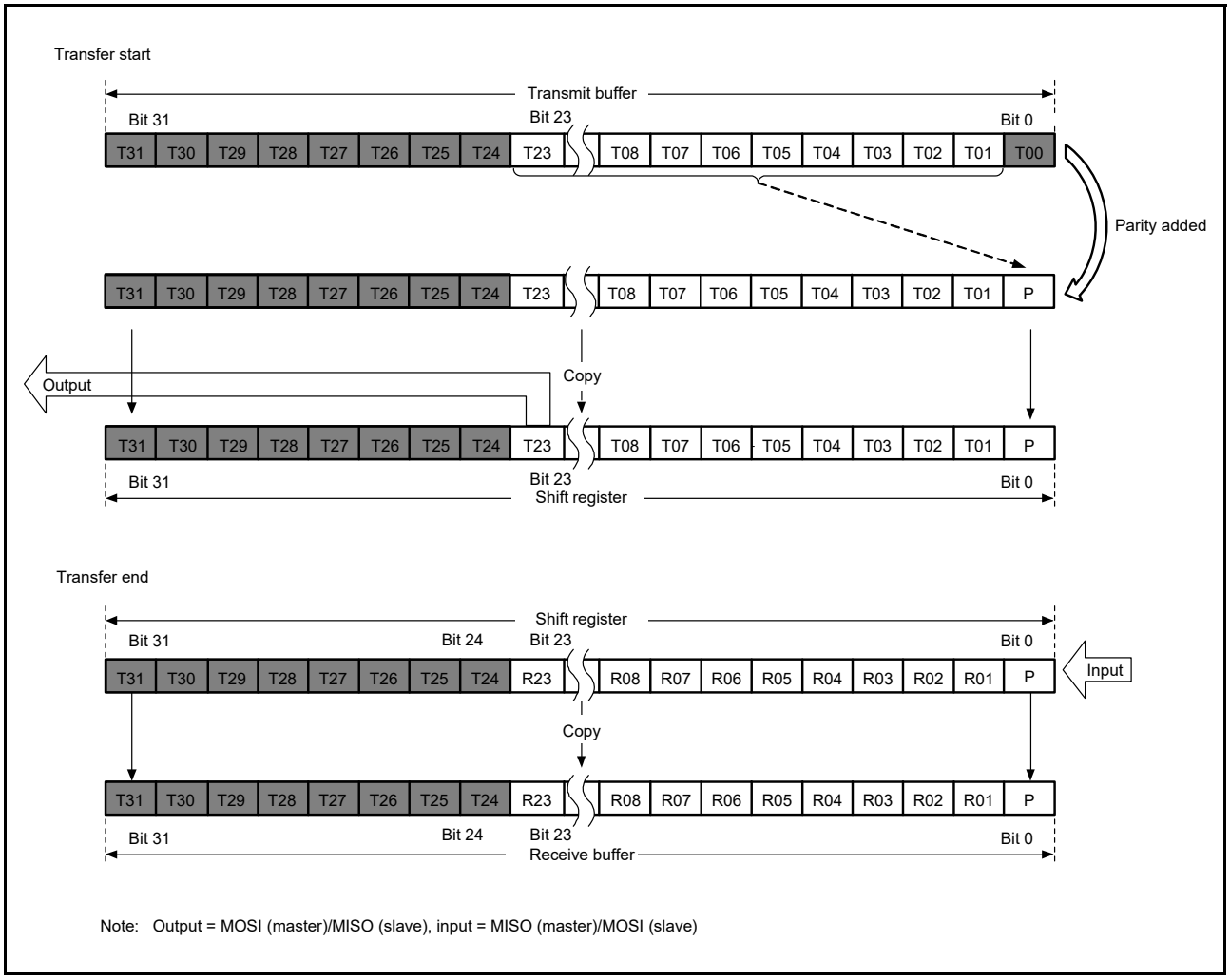


Figure 33.19 MSB-first transfer with 24-bit data, parity enabled

(3) LSB-First Transfer 32-Bit Data

Figure 33.20 shows operations by 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 are 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 bits R00 to P are collected after input of the required number of cycles of RSPCK, 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.

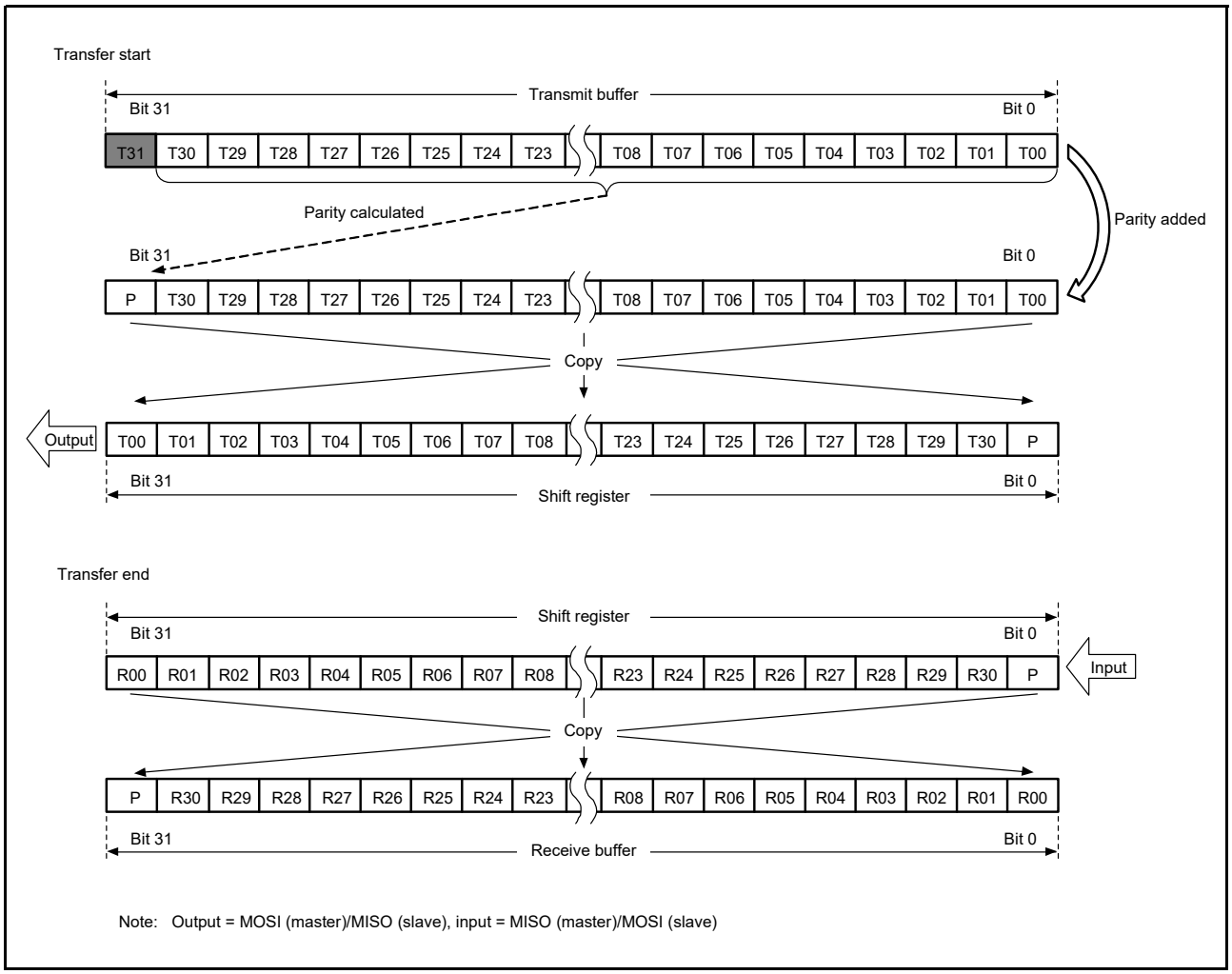


Figure 33.20 LSB-first transfer with 32-bit data, parity enabled

(4) LSB-First Transfer with 24-Bit Data

Figure 33.21 shows operations by the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, 24 bits as the SPI data length for an example that is not 32 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 bits R00 to P are collected after input of the required number of cycles of RSPCK, 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 are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the upper 8 bits of the receive buffer.

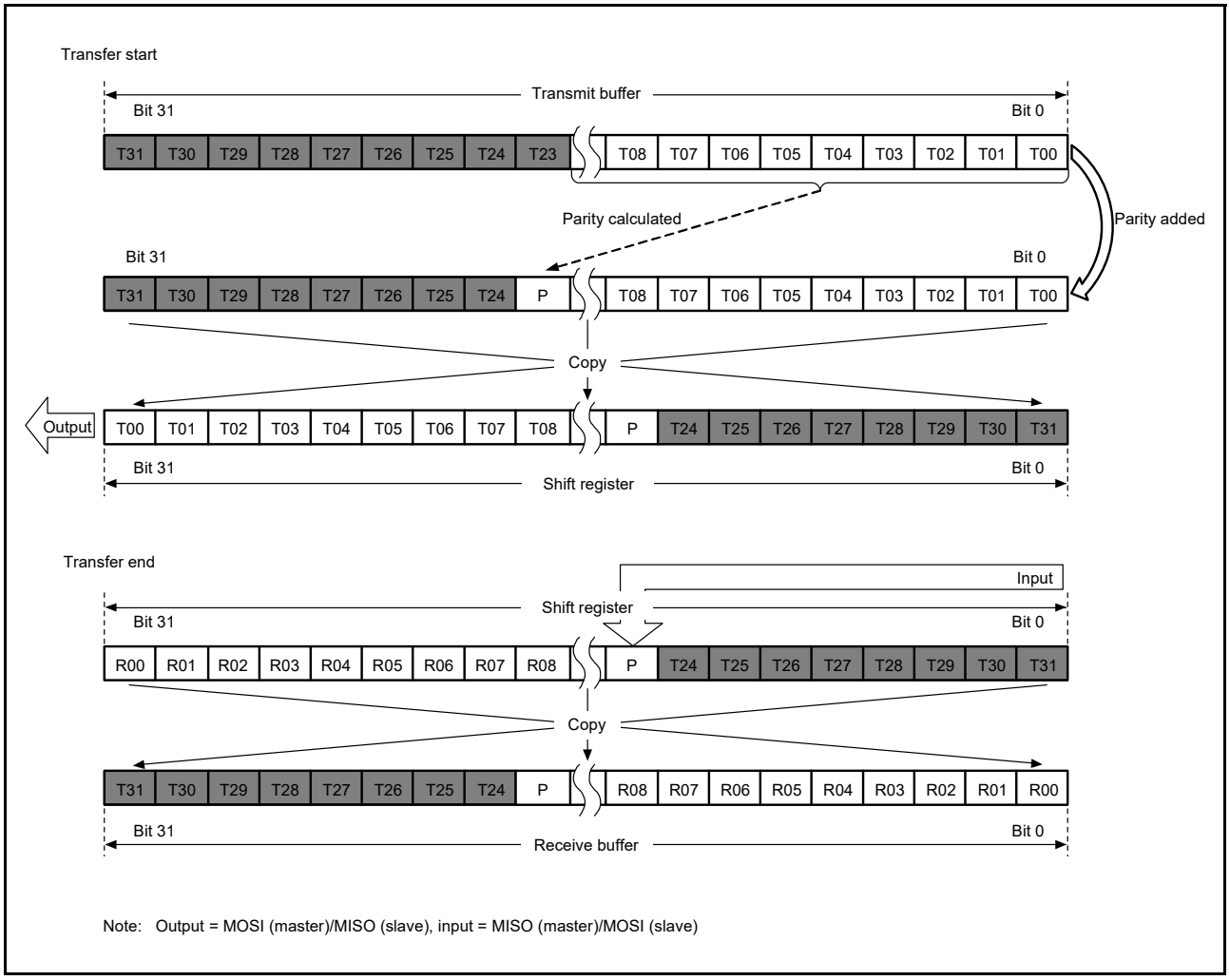


Figure 33.21 LSB-first transfer with 24-bit data and parity enabled

33.3.5 Transfer Formats

33.3.5.1 CPHA = 0

Figure 33.22 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.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 33.22, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0. 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 33.3.2, Controlling SPI Pins.

When the SPCMDm.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 RSPCK cycle. The change timing for the 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. 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 of the MCU is in master mode, see section 33.3.10.1, Master Mode Operation.

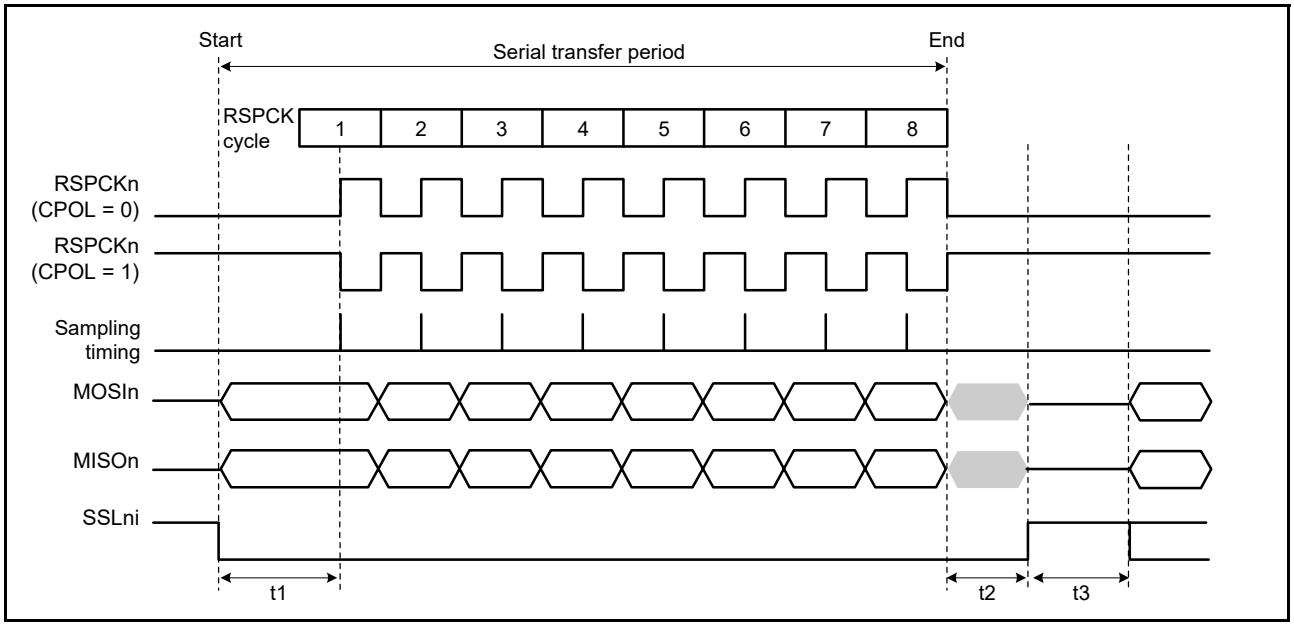


Figure 33.22 SPI transfer format with CPHA = 0

33.3.5.2 CPHA = 1

Figure 33.23 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.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 33.23, RSPCK (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0. RSPCK (CPOL = 0) 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). For details, see section 33.3.2, Controlling SPI Pins.

When the SPCMDm.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 SPCMDm.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 33.3.10.1, Master Mode Operation.

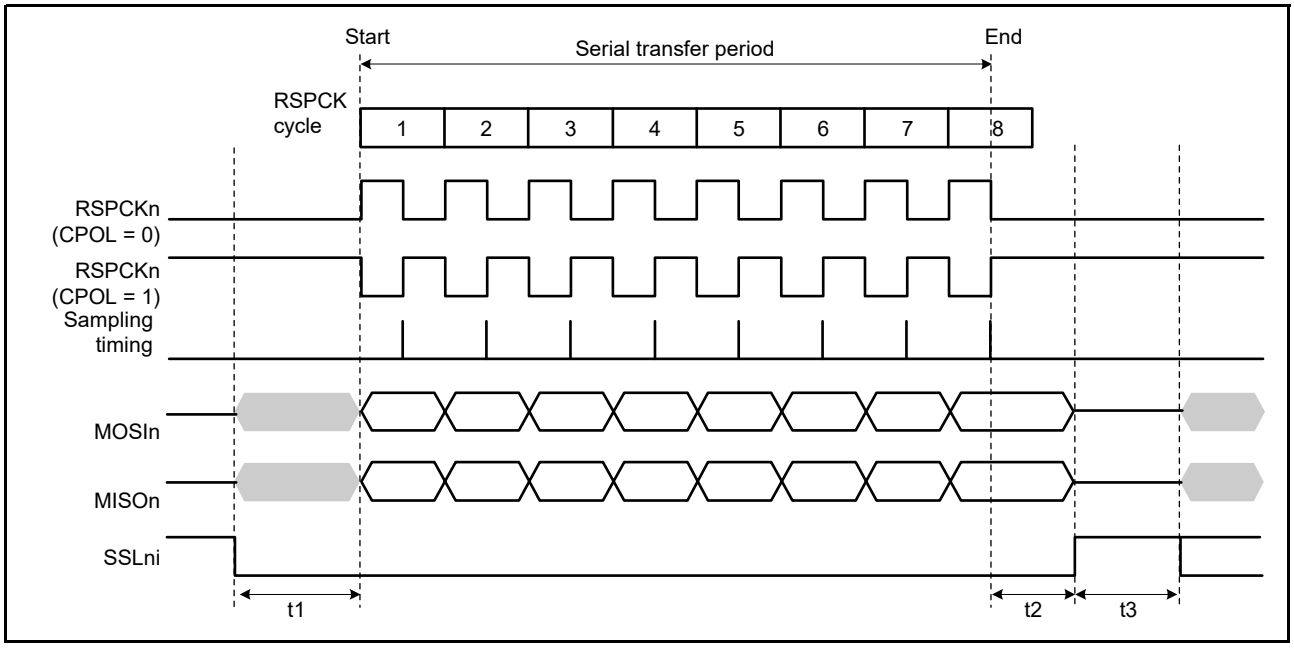


Figure 33.23 SPI transfer format with CPHA = 1

33.3.6 Data Transfer Modes

Full-duplex synchronous serial communications or transmit operations can only be selected by the communications operating mode select bit (SPCR.TXMD). The SPDR/SPDR_HA access shown in Figure 33.24 and Figure 33.25 indicate the condition of access to the SPDR/SPDR_HA register, where W denotes a write cycle.

33.3.6.1 Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0)

Figure 33.24 shows an example of operation where the communications operating mode select bit (SPCR.TXMD) is set to 0. In the example, the SPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, meaning the number of transferred bits.

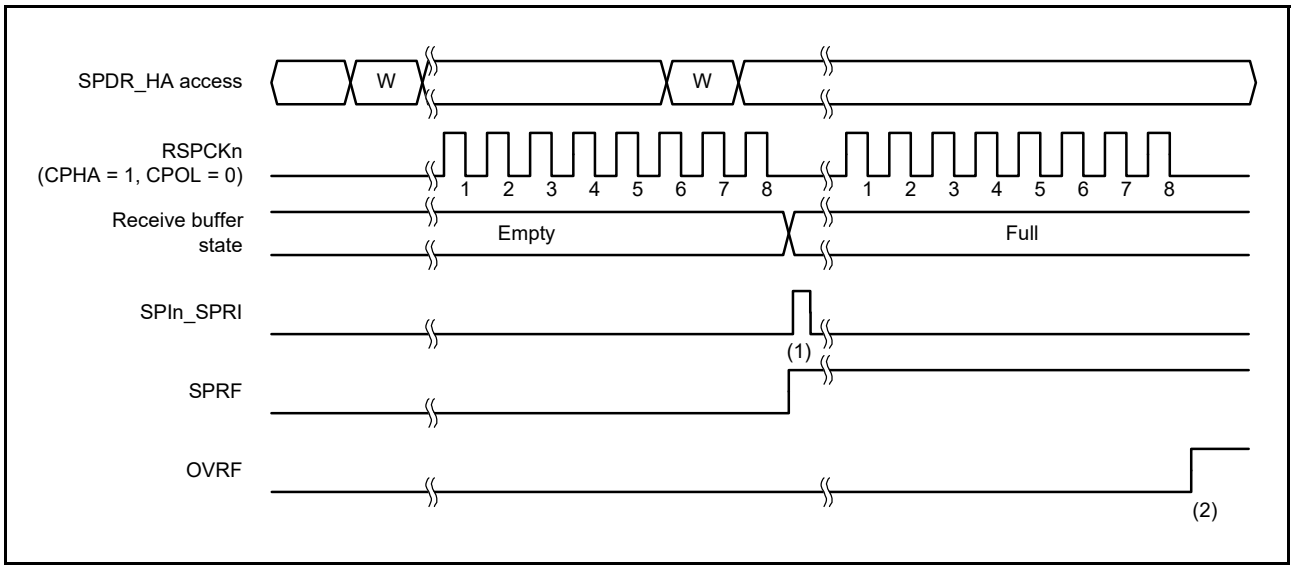


Figure 33.24 Operation example when SPCR.TXMD = 0

The operation of the flags at timings shown in (1) and (2) in the figure is as follows:

(1) When a serial transfer ends with the SPDR_HA receive buffer empty, the SPI generates a receive buffer full interrupt request (SPIn_SPRI), the SPSR.SPRF flag is set to 1, and the received data is copied from 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.

33.3.6.2 Transmit-Only Operations (SPCR.TXMD = 1)

Figure 33.25 shows an example operation where the communications operating mode select bit (SPCR.TXMD) is set to 1. In the example, the SPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.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.

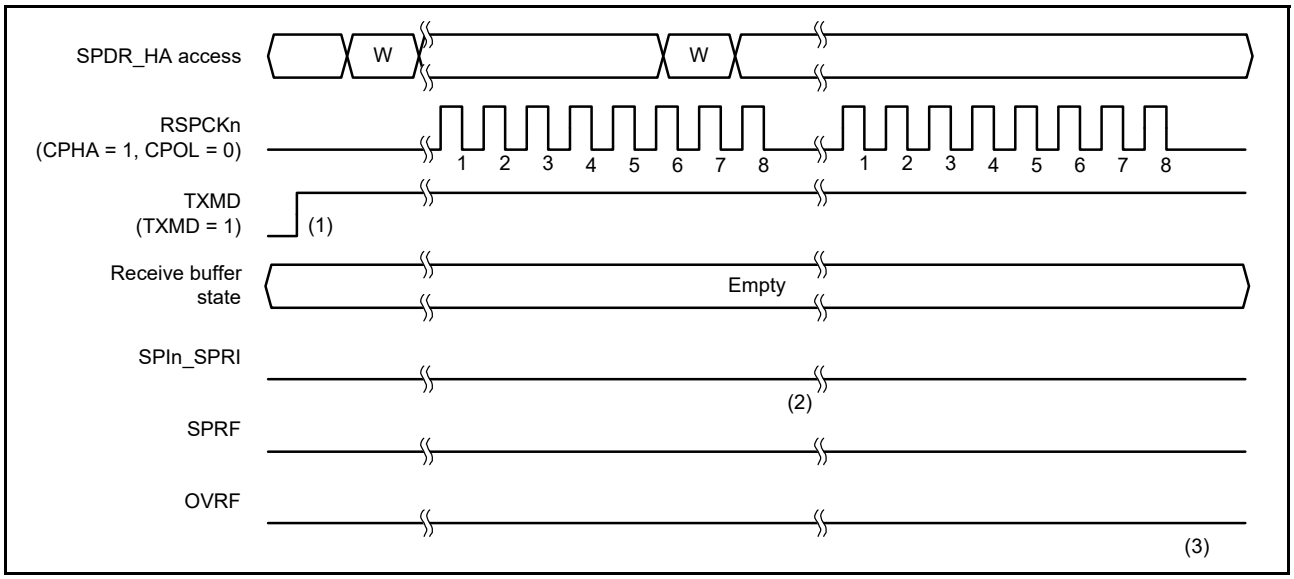


Figure 33.25 Operation example with SPCR.TXMD = 1

The operation of the flags at timings shown in (1) to (3) in the figure is as follows:

(1) Make sure there is no data left in the receive buffer (SPSR.SPRF flag is 0) and the SPSR.OVRF flag is 0 before entering the mode of transmit-only operations (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.

When performing transmit-only operations (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).

33.3.7 Transmit Buffer Empty and Receive Buffer Full Interrupts

Figure 33.26 and Figure 33.27 show an example operation of the transmit buffer empty interrupt (SPIn_SPTI) and the receive buffer full interrupt (SPIn_SPRI). The SPDR_HA register access shown in Figure 33.26 and Figure 33.27 indicates the condition of access to the SPDR_HA register, where W denotes a write cycle, and R a read cycle. In Figure 33.27, the SPI performs an 8-bit serial transfer in which the SPCR.TXMD bit is 0, the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. In Figure 33.26, the SPI performs an 8-bit serial transfer in which the SPCR.TXMD bit is 0, the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 0, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, meaning the number of transferred bits.

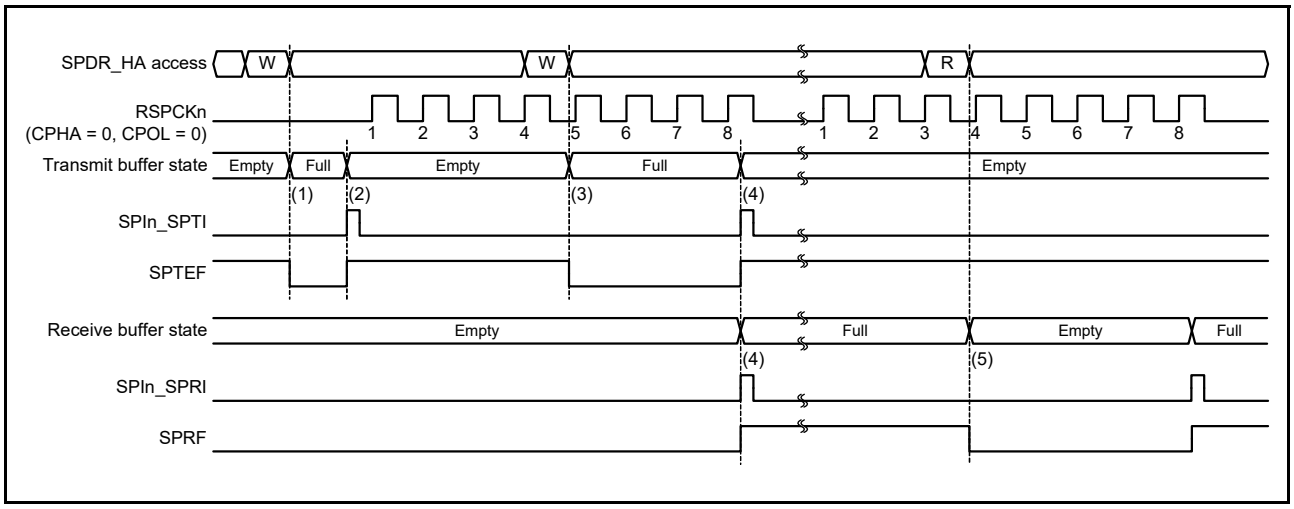


Figure 33.26 Operation example of SPIn_SPTI and SPIn_SPRI interrupts with CPHA = 0 and CPOL = 0

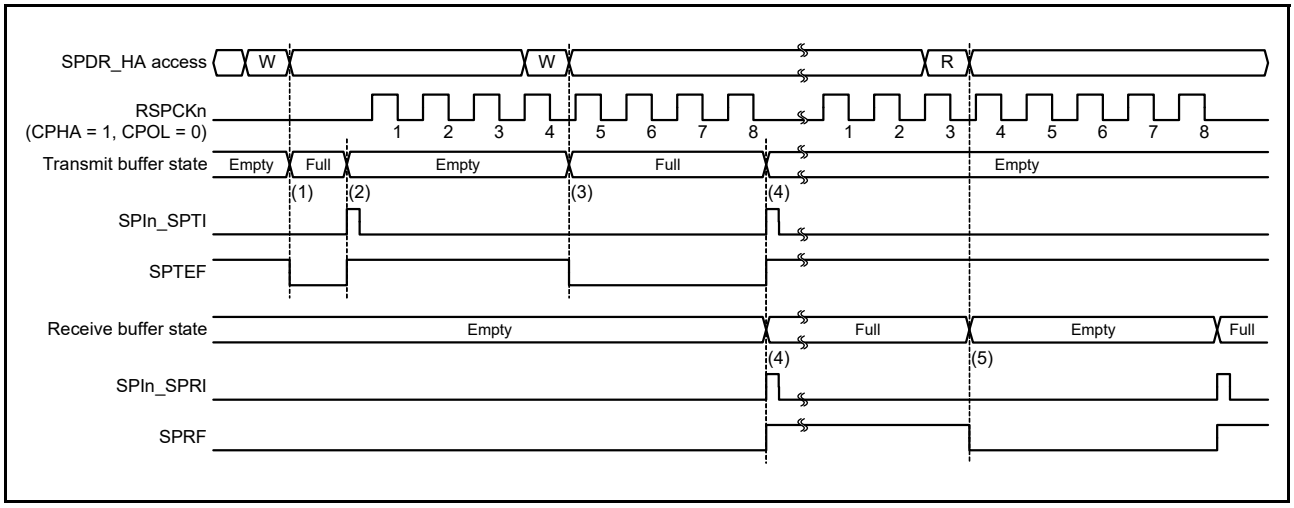


Figure 33.27 Operation example of SPIn_SPTI and SPIn_SPRI interrupts with CPHA = 1 and CPOL = 0

The operation of the SPI shown in (1) to (5) in the figure is as follows:

(1) When transmit data is written to SPDR_HA when the transmit buffer of SPDR_HA is empty (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 and generates a transmit buffer empty interrupt request (SPIn_SPTI) and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the mode of the SPI. For details, see section 33.3.10, SPI Operation, and section 33.3.11, 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 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 and generates a receive buffer full interrupt request (SPIn_SPRI), and sets the SPRF flag to 1. Because the shift register is empty on completion of serial transfer, if the transmit buffer was full before the serial transfer ended, the SPI sets the SPTEF flag to 1 and copies the 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, and 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 using the SPRF flag, the receive data can be read.

If SPDR_HA is written to when the transmit buffer holds untransmitted data (the SPTEF flag is 0), the SPI does not update the data in the transmit buffer. When writing to SPDR_HA, make sure either to use a transmit buffer empty interrupt request or to process a 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 with the receive buffer 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 33.3.8, 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 in the ICU, where n is the interrupt vector number), can be used to confirm the states of the transmission and reception buffers. Similarly, the SPTEF and SPRF flags can be used to confirm the states of the transmission and reception buffers. See section 14, Interrupt Controller Unit (ICU), for the interrupt vector numbers.

33.3.8 Error Detection

In the normal SPI serial transfer, the data written to the SPDR/SPDR_HA transmit buffer is transmitted, and the received data can be read from the SPDR/SPDR_HA receive buffer. 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 a non-normal transfer occurs, the SPI detects the event as an underrun error, overrun error, parity error, or mode fault error. Table 33.8 lists the relationship between non-normal transfer operations and the SPI error detection function.

Table 33.8 Relationship between non-normal transfer operations and SPI error detection function

	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 MISOA output signal is stopped • SPI function is disabled.	Underrun error
4	Serial transfer terminates when the receive buffer is full	• The contents of the receive buffer are kept • Missing receive data.	Overrun error
5	An incorrect parity bit is received during full-duplex synchronous serial communications with the parity function enabled	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 to SSLn3 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 to SSLn3 output signals is stopped • SPI function is disabled.	Mode fault error
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 33.8, the SPI does not detect an error. To prevent data omission during writes to SPDR/SPDR_HA, the writes must be executed using a transmit buffer empty interrupt request (when 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 reads must be executed using an SPI receive buffer full interrupt request (when SPSR.SPRF flag is 1).

For information on:

- Underrun errors, indicated in operation 3, see section 33.3.8.4, Underrun Errors.
- Overrun errors, indicated in operation 4, see section 33.3.8.1, Overrun Errors.
- Parity errors, indicated in operation 5, see section 33.3.8.2, Parity Errors.
- Mode fault errors, indicated in 6 to 8, see section 33.3.8.3, Mode Fault Errors.
- Transmit and receive interrupts, see section 33.3.7, Transmit Buffer Empty and Receive Buffer Full Interrupts.

33.3.8.1 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 occurrence of the error is saved in the receive buffer. To set the OVRF flag to 0, write 0 to the OVRF flag after the CPU has read SPSR with the OVRF flag set to 1.

Figure 33.28 shows an example operation of the OVRF and SPRF flags. The SPSR and SPDR_HA accesses shown in Figure 33.28 indicates the condition of accesses to SPSR and SPDR_HA, respectively, where W denotes a write cycle, and R a read cycle. In the example, the SPI performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, meaning the number of transferred bits.

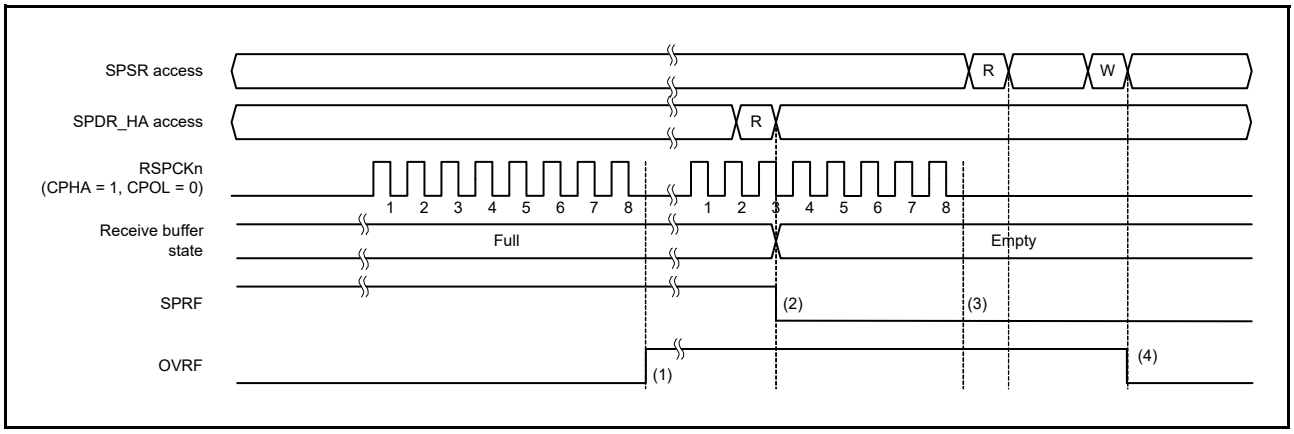


Figure 33.28 Operation example of OVRF flag and SPRF flag

The operation of the flags at the timing shown in (1) to (4) in Figure 33.28 is as follows:

(1) If a serial transfer terminates with the SPRF flag being 1 (the 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 if the SPPE bit is 1, parity errors are not detected. In master mode, the SPI copies the value of the pointer to the SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) When SPDR_HA is read, the SPI outputs the data in the receive buffer. The SPRF flag is then set to 0. The emptying of the receive buffer does not set the OVRF flag to 0.

(3) If the serial transfer ends with the OVRF flag being 1 (an overrun error occurs), the SPI does not copy the data in the shift register to the receive buffer (the SPRF flag is not set to 1). A receive buffer full interrupt is not generated. Even if the SPPE bit is 1, parity errors are not detected. When in master mode, the SPI does not update the SPSSR.SPECM[2:0] bits. When in an overrun error state and 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 is set to 0.

The application can check for an overrun either by reading SPSR or by using an SPI error interrupt and reading SPSR. When executing a serial transfer, make sure that overrun errors are detected early, for instance by reading SPSR immediately after SPDR_HA is read. When the SPI is used in master mode, the value of the pointer to the SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

If an overrun error occurs and the OVRF flag is set to 1, normal reception operations cannot be performed until the OVRF flag is set to 0.

When the RSPCK auto-stop function is enabled in master mode, an overrun error does not occur. Figure 33.29 and Figure 33.30 show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.

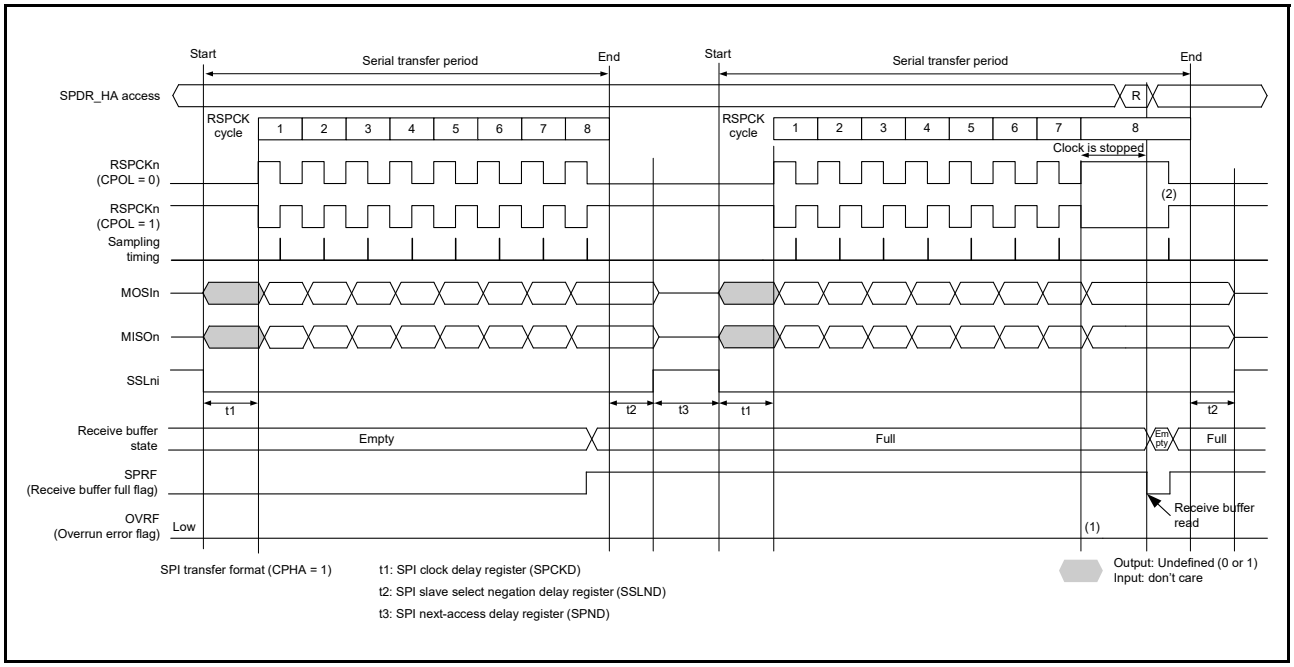


Figure 33.29 Clock stop waveform when serial transfer continues while receive buffer is full in master mode (CPHA = 1)

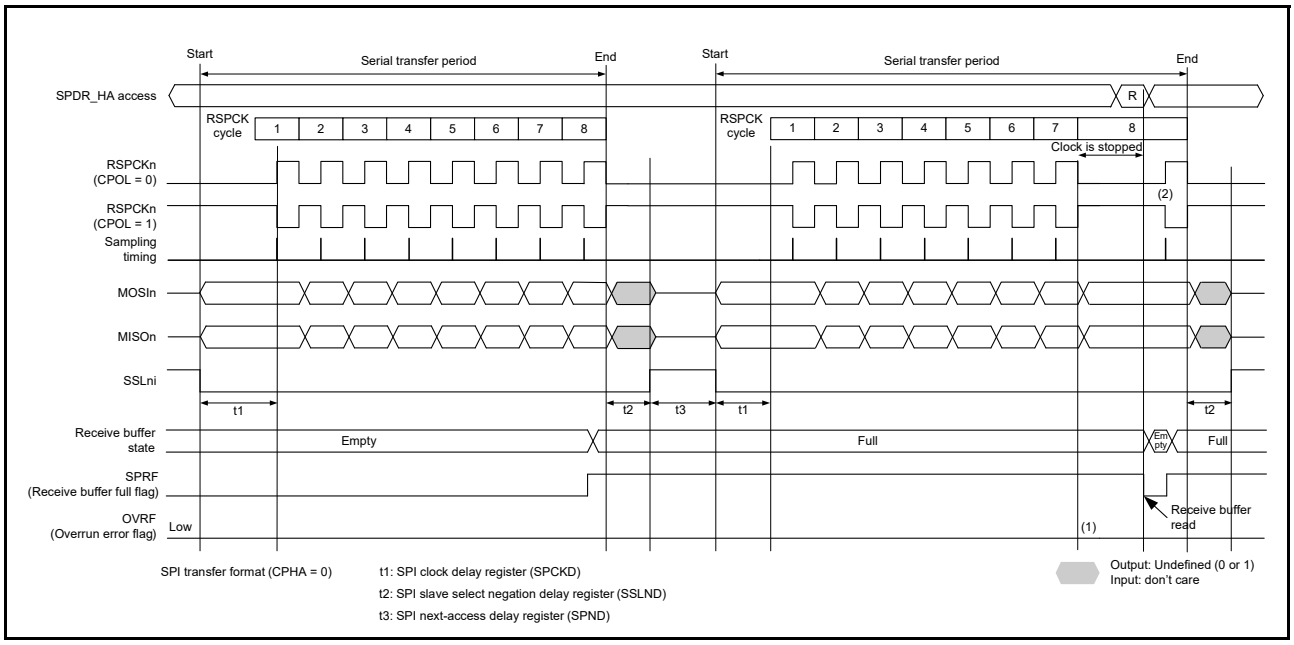


Figure 33.30 Clock stop waveform when serial transfer continues while receive buffer is full in master mode (CPHA = 0)

The operation of the flags at the timings shown in (1) and (2) in Figure 33.29 and Figure 33.30 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_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 SPSR.SPRF flag is set to 0).

33.3.8.2 Parity Errors

If 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 SPSR register is read with the PERF flag set to 1.

Figure 33.31 shows an example operation of the OVRF and PERF flags. The SPSR access shown in Figure 33.31 indicates the condition of access to the SPSR register, where W denotes a write cycle, and R a read cycle. In the example of Figure 33.31, full-duplex synchronous serial communications is performed while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1. The SPI performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, meaning the number of transferred bits.

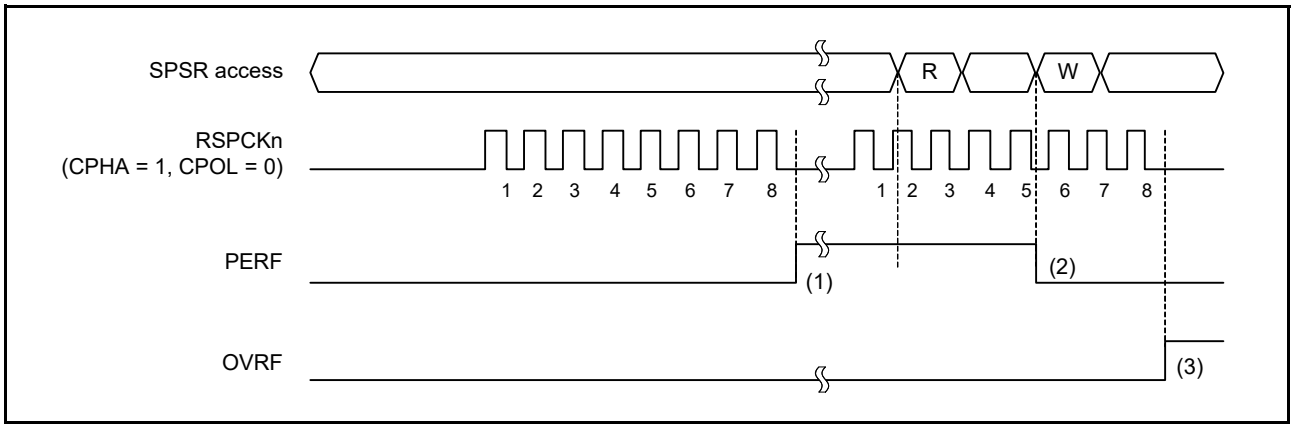


Figure 33.31 Operation example of PERF flag

The operation of the flags at the timing shown in (1) to (3) in Figure 33.31 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. In master mode, the SPI copies the value of the pointer to the SPCMDm register to the SPSSR.SPECM[2:0] bits.

(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.

The application can check for a parity error either by reading the SPSR register or by using an SPI error interrupt and reading the SPSR register. When executing a serial transfer, make sure that parity errors are detected early, for instance by reading SPSR. When the SPI is used in master mode, the pointer value to the SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

33.3.8.3 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 it sets the SPSR.MODF flag to 1. On detecting the mode fault error, the SPI copies the value of the pointer to SPCMDm to the SPSSR.SPECM[2:0] bits. 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 driving of the output signals and clears the SPCR.SPE bit to 0 (see section 33.3.9, Initializing the SPI). For multi-master configuration, detection of a mode fault error is used to stop the driven 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 using the SPI in master mode, the value of the pointer to the SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

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.

33.3.8.4 Underrun Errors

When a serial transfer begins with the SPCR.MSTR bit set to 0 (slave mode), the SPCR.SPE bit set to 1, and the transmission data not prepared, the SPI detects an underrun error. The SPI then sets the SPSR.MODF flag and the SPSR.UDRF flag to 1.

On detecting an underrun error, the SPI stops driving the output signals and clears the SPCR.SPE bit to 0 (see section 33.3.9, Initializing the SPI).

Underrun errors can be checked for either by reading the SPSR register or by using an SPI error interrupt and reading the SPSR register. 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, set the MODF flag to 0.

33.3.9 Initializing the SPI

If 0 is written to the SPCR.SPE bit or 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. The following describes initialization by clearing the SPCR.SPE bit and initialization by a system reset.

33.3.9.1 Initialization by Clearing the 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 (SPSR.SPTEF flag is set to 1).

Initialization by the 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, if the SPE bit is set to 1 again.

The SPSR.SPRF, SPSR.OVRF, SPSR.MODF, SPSR.PERF, SPSR.UDRF flags, and the value of the SPI sequence status register (SPSSR) 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 (SPSR.SPTEF flag is set to 1). Therefore, if the SPCR.SPTIE bit is set to 1 after SPI initialization, a transmit buffer empty interrupt is generated. When the SPI is initialized, to disable any transmit buffer empty interrupts, write 0 to the SPTIE bit simultaneously while writing 0 to the SPE bit.

33.3.9.2 Initialization by System Reset

A system reset completely initializes the SPI by initializing all bits that control the SPI, the status bits, and the data registers, in addition to the requirements described in section 33.3.9.1, Initialization by Clearing the SPE Bit.

33.3.10 SPI Operation

33.3.10.1 Master Mode Operation

The only difference between single-master and multi-master mode operation is the use of mode fault error detection (see section 33.3.8, Error Detection). In single-master mode, the SPI does not detect mode fault errors, while in multi-master mode, it does. This section explains operations that are common to both modes.

(1) Starting 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 being empty (data for the next transfer is not set) (SPSR.SPTEF flag is 1). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR/SPDR_HA, 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, 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.

For details on the SPI transfer format, see section 33.3.5, Transfer Formats. The polarity of the SSLni output pins depends on the SSLP register settings.

(2) Terminating Serial Transfer

Regardless of the SPCMDm.CPHA bit setting, the SPI terminates a serial transfer after transmitting an RSPCKn edge associated with the final sampling timing. If free space is available in the receive buffer (SPRX) (SPSR.SPRF flag is 0), on termination of serial transfer, the SPI copies data from the shift register to the receive buffer of the SPDR/SPDR_HA register.

Note: The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMDm.SPB[3:0] bit setting. The polarity of the SSLni output pin depends on the SSLP register settings. For details on the SPI transfer format, see section 33.3.5, Transfer Formats.

(3) Sequence Control

The transfer format in master mode is determined by the SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers.

The SPSCR register determines the sequence configuration for serial transfers that the SPI executes in master mode. The following items are set in the SPCMDm register:

- SSLni pin output signal value
- MSB-first/LSB-first
- Data length
- Some of the bit rate settings
- RSPCK polarity/phase
- Whether SPCKD is to be referenced
- Whether SSLND is to be referenced
- Whether SPND is to be referenced.

The SPBR register holds some of the bit rate settings such as SPCKD, an SPI clock delay value, SSLND, an SSL negation delay, and SPND, a next-access delay value.

According to the sequence length that is assigned to SPSCR, the SPI makes up a sequence comprised of a part or all of SPCMDm register. The SPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the SPI function is enabled, the SPI loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The SPI increments the pointer each time the next-access delay period for a data transfer ends. On completion of the serial transfer that corresponds to the final command containing the sequence, the SPI sets the pointer to SPCMD0 to execute the sequence repeatedly.

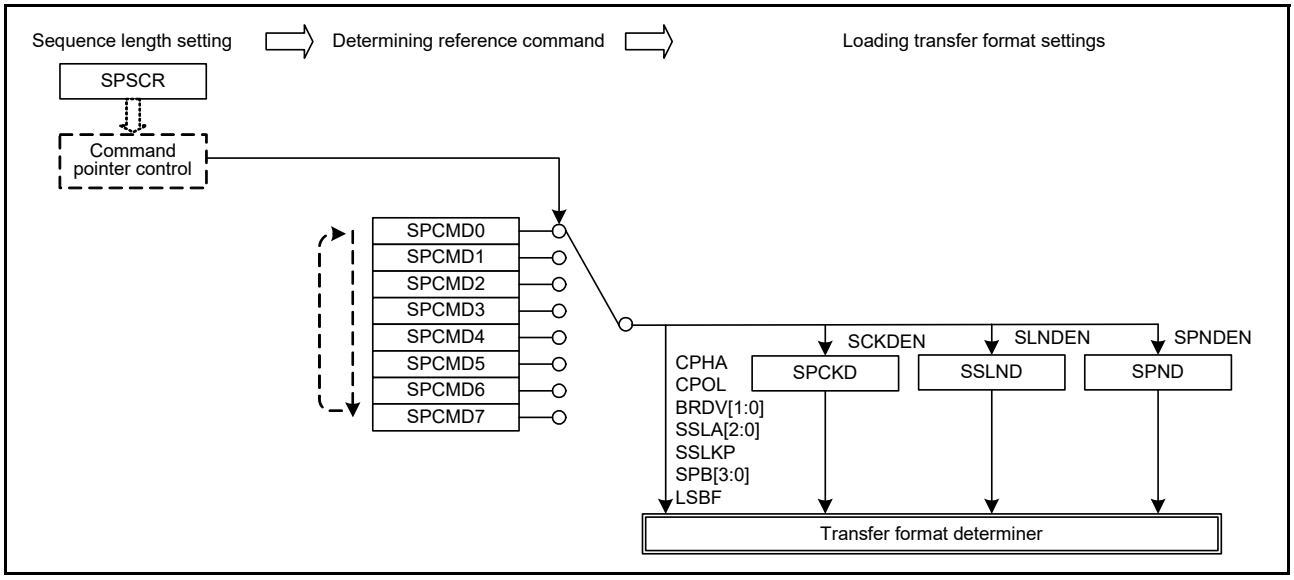


Figure 33.32 Procedure to determine the format of serial transfer in master mode

In this section, a frame is the combination of the SPDR/SPDR_HA data and the SPCMDm settings.

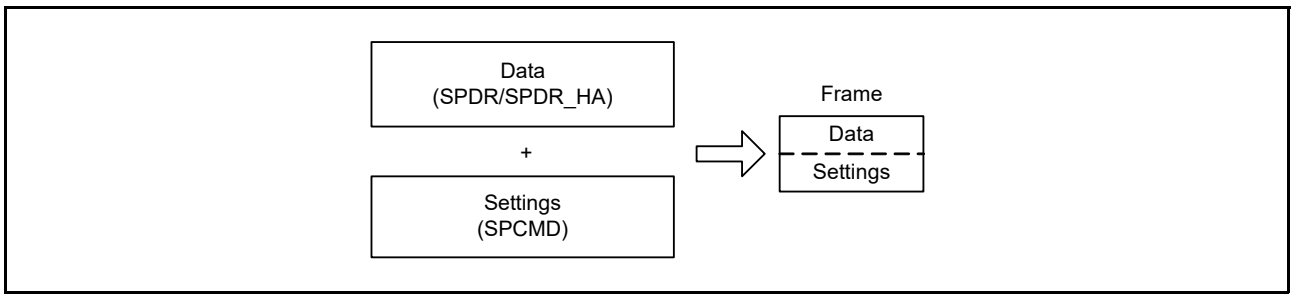


Figure 33.33 Conceptual diagram of frames

Figure 33.34 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 33.4.

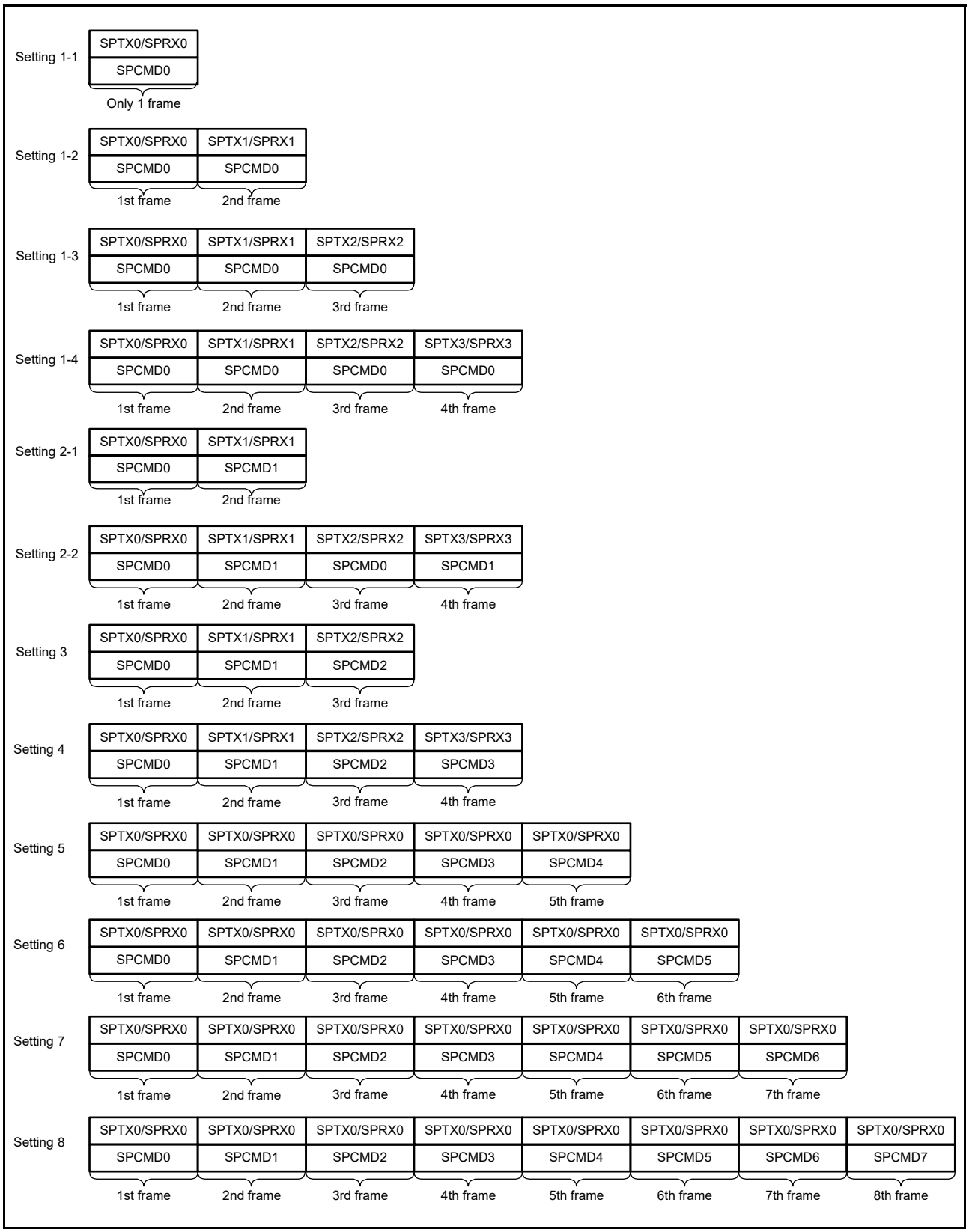


Figure 33.34 Relationship between the SPI Command Register and the transmit/receive buffers in sequence operations

(4) Burst Transfer

If the SPCMDm.SSLKP bit that the SPI references during the current serial transfer is 1, the SPI maintains the SSLni signal level during the serial transfer until the beginning of the SSLni signal assertion for the next serial transfer. If the SSLni signal level for the next serial transfer is the same as the SSLni signal level for the current serial transfer, the SPI can execute continuous serial transfers while keeping the SSLni signal assertion status (burst transfer).

Figure 33.35 shows an example of an SSLni signal operation for a burst transfer that is implemented using SPCMD0 and SPCMD1 register settings. The following section explains the SPI operations (1) to (7) as shown in Figure 33.35.

Note: The polarity of the SSLni output signal depends on the SSLP register settings.

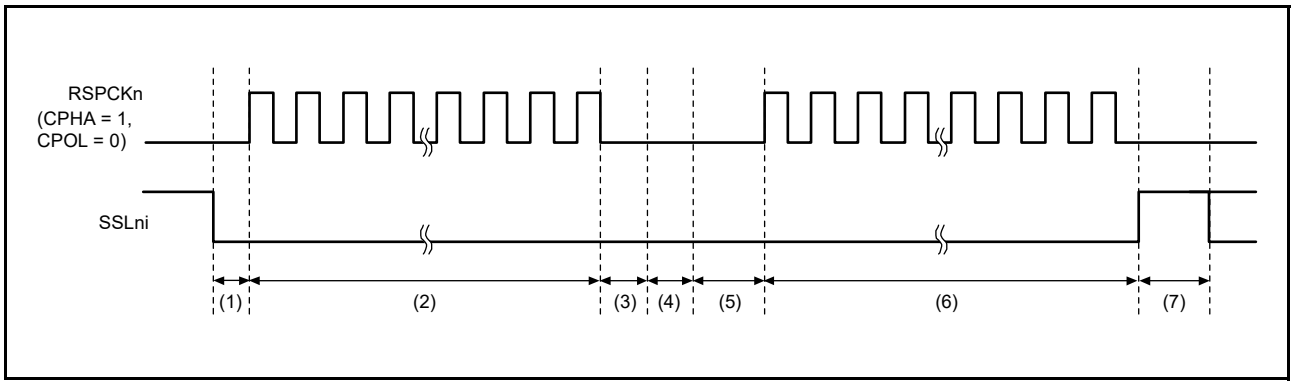


Figure 33.35 Example of burst transfer operation using the SSLKP bit

1. Based on SPCMD0, the SPI asserts the SSLni signal and inserts RSPCK delays.
2. The SPI executes serial transfers according to SPCMD0.
3. The SPI inserts SSL negation delays.
4. Because the SPCMD0.SSLKP bit is 1, the SPI keeps the SSLni signal value on SPCMD0. This period is sustained at a minimum for a period equals to the next-access delay of SPCMD0. If the shift register is empty after the minimum period has passed, this period is sustained until the transmit data is stored in the shift register for the next transfer.
5. Based on SPCMD1, the SPI asserts the SSLni signal and inserts RSPCK delays.
6. The SPI executes serial transfers according to SPCMD1.
7. Because the SPCMD1.SSLKP bit is 0, the SPI negates the SSLni signal. In addition, a next-access delay is inserted according to SPCMD1.

If the SSLni signal output settings in the SPCMDm register where 1 is assigned to the SSLKP bit are different from the SSLni signal output settings in the SPCMDm register to be used in the next transfer, the SPI switches the SSLni signal status to SSLni signal assertion as shown in (5) in Figure 33.35. This corresponds to the command for the next transfer.

Note: If such an SSLni signal switching occurs, the slaves that drive the MISOn signal compete, and collision of signal levels might occur.

The SPI in master mode references the SSLni signal operation within the module when the SSLKP bit is not used. Even when the SPCMDm.CPHA bit is 0, the SPI can accurately start serial transfers using the SSLni signal assertion for the next transfer that is detected internally.

(5) RSPCK Delay (t1)

The RSPCK delay in master mode depends on the SPCMDm.SCKDEN bit setting and the SPCKD register setting. The SPI determines the SPCMDm register to be referenced during a serial transfer by pointer control, and determines an RSPCK delay using the SPCMDm.SCKDEN bit and SPCKD, as listed in Table 33.9. For a definition of RSPCK delay, see section 33.3.5, Transfer Formats.

Table 33.9 Relationship between the SCKDEN bit, SPCKD, and RSPCK delays

SPCMDm.SCKDEN bit	SPCKD.SCKDL[2:0] bits	RSPCK delays
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

(6) SSL Negation Delay (t2)

The SSL negation delay in master mode depends on the SPCMDm.SLNDEN bit setting and the SSLND register setting. The SPI determines the SPCMDm register to be referenced by pointer control during serial transfer, and determines the SSL negation delay using the SPCMDm.SLNDEN bit and SSLND, as listed in Table 33.10. For a definition of SSL negation delay, see section 33.3.5, Transfer Formats.

Table 33.10 Relationship between SLNDEN bit, SSLND, and SSL negation delay

SPCMDm.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

(7) Next-Access Delay (t3)

The next-access delay in master mode depends on the SPCMDm.SPNDEN bit setting and the SPND register setting. The SPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines the next-access delay during serial transfer using the SPCMDm.SPNDEN bit and SPND, as listed in Table 33.11. For a definition of next-access delay, see section 33.3.5, Transfer Formats.

Table 33.11 Relationship between SPNDEN bit, SPND, and next-access delays

SPCMDm.SPNDEN bit	SPND.SPNDL[2:0] bits	Next-access delays
0	000b to 111b	1 RSPCK + 2 PCLK
1	000b	1 RSPCK + 2 PCLK
	001b	2 RSPCK + 2 PCLK
	010b	3 RSPCK + 2 PCLK
	011b	4 RSPCK + 2 PCLK
	100b	5 RSPCK + 2 PCLK
	101b	6 RSPCK + 2 PCLK
	110b	7 RSPCK + 2 PCLK
	111b	8 RSPCK + 2 PCLK

(8) Initialization Flow

Figure 33.36 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, DMAC, and I/O ports, see the individual block descriptions.

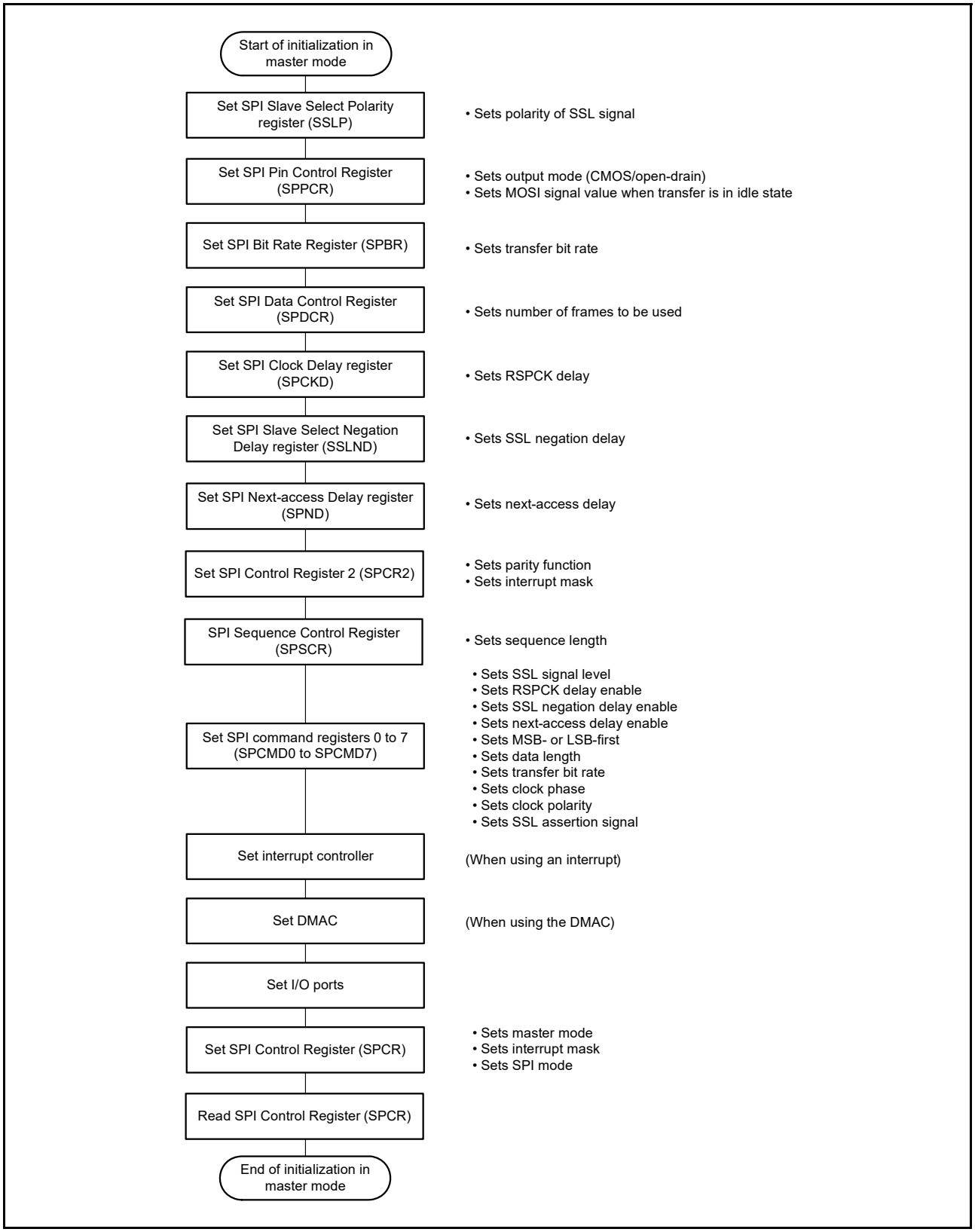


Figure 33.36 Example of initialization flow in master mode for SPI operation

(9) Software Processing Flow

Figure 33.37 to Figure 33.39 show example flows of software processing.

(a) Transmit Processing Flow

When transmitting data and when the SPIn_SPII interrupt is enabled, the CPU is notified of the completion of data transmission after the last data write for transmission.

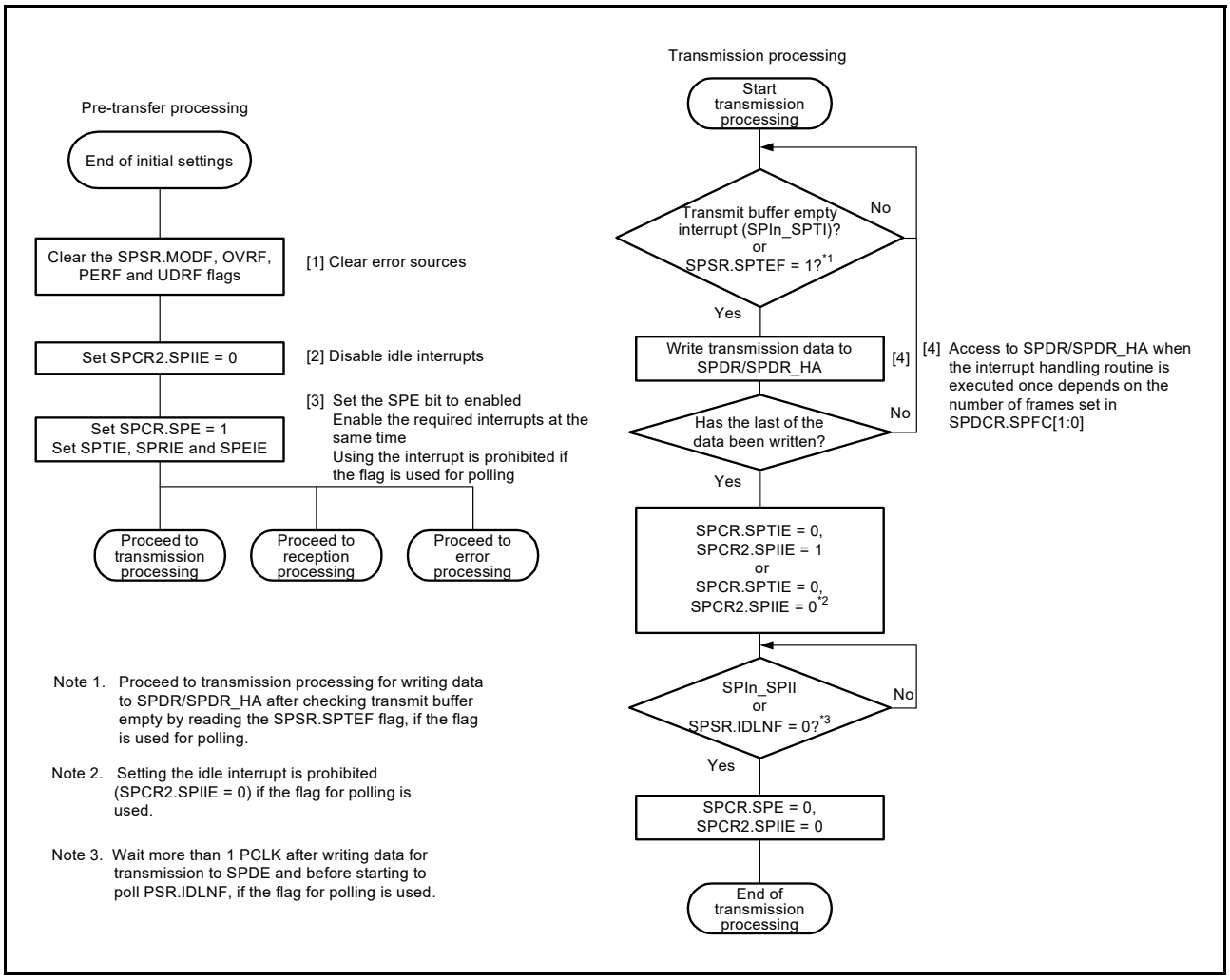


Figure 33.37 Transmission flow in master mode transmission

(b) Receive Processing Flow

The SPI does not handle receive-only operations, so processing for transmission is required.

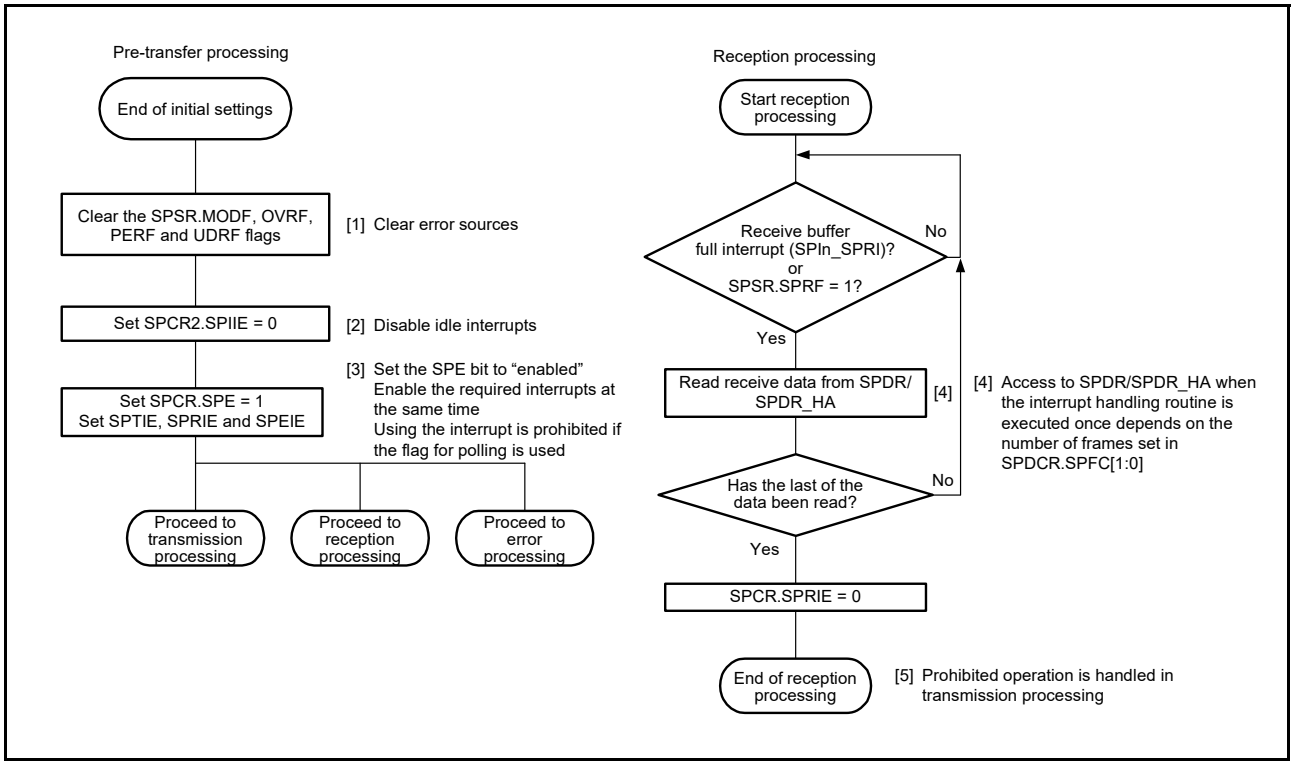


Figure 33.38 Reception flow in master mode

(c) Flow of Error Processing

The SPI detects mode fault errors, underrun errors, overrun errors, and parity errors. When a mode fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, the SPCR.SPE bit is not cleared and operations for transmission and reception continue. Renesas recommends clearing the SPCR.SPE bit to stop operations for errors other than mode-fault errors. Not doing so leads to updating of the SPSSR.SPECM[2:0] bits.

When an error occurs using 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 transmit buffer empty or the receive buffer full interrupt request. If the SPIn_SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

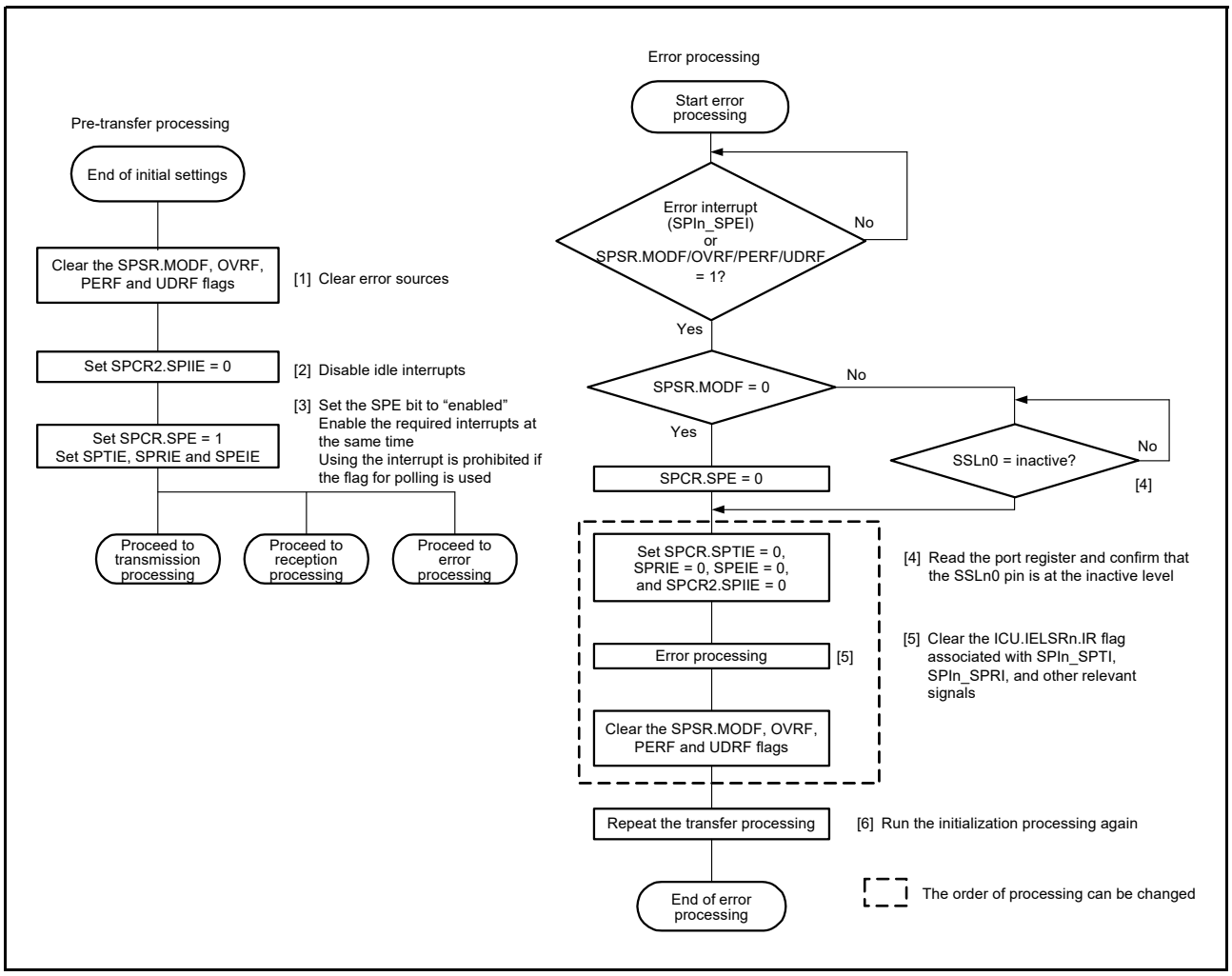


Figure 33.39 Flowchart for error processing in master mode

33.3.10.2 Slave Mode Operation

(1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSLn0 input signal assertion, the SPI 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.

If the CPHA bit is 1, when detecting the first RSPCKn edge in an SSLn0 signal asserted condition, the SPI must drive valid data to the MISOn output signal. 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 33.3.5, Transfer Formats. The polarity of the SSLn0 input signal depends on the setting of the SSLP.SSL0P bit.

(2) Terminating Serial Transfer

Regardless of the SPCMD0.CPHA bit, the SPI terminates the serial transfer after detecting an RSPCKn edge associated with the final sampling timing. When free space is available in the receive buffer (SPSR.SPRF flag is 0), on termination of a 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 33.3.8, Error Detection).

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] bit setting. The polarity of the SSLn0 input signal depends on the SSLP.SSL0P bit setting.

For details on the SPI transfer format, see section 33.3.5, 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 type of configuration shown in Figure 33.7 as an example, if the SPI is used in single-slave mode, the SSLn0 signal is fixed at the active state. Therefore, when the CPHA bit is set to 0, the SPI cannot correctly start a serial transfer. To correctly execute transmit and receive operations by the SPI in slave mode in a configuration in which the SSLn0 input signal is fixed at the active state, the CPHA bit must be set to 1. If the application requires setting the CPHA bit to 0, the SSLn0 input signal must not be fixed.

(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. If the CPHA bit is 1, the period from the first RSPCKn edge to the sampling timing for the reception of the final bit in an SSLn0 signal active state corresponds to a serial transfer period. 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.

If the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

(5) Initialization Flow

Figure 33.40 shows an example of initialization flow for SPI operation where the SPI is in slave mode. For information on how to set up the Interrupt Controller Unit, DMAC, and I/O ports, see the individual block descriptions.

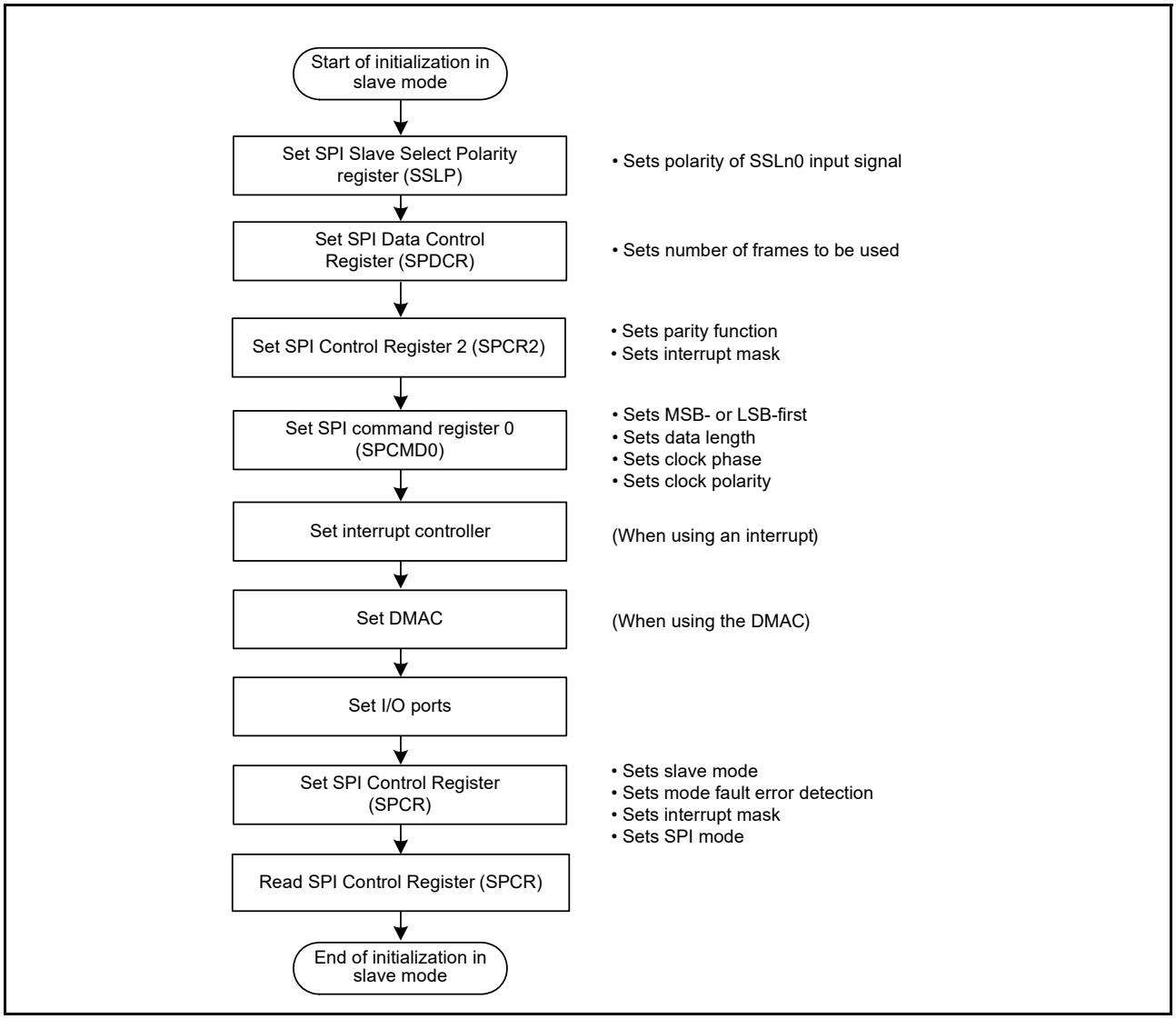


Figure 33.40 Example initialization flow in slave mode for SPI operation

(6) Software Processing Flow

Figure 33.41 to Figure 33.43 show example flows of software processing.

(a) Transmit Processing Flow

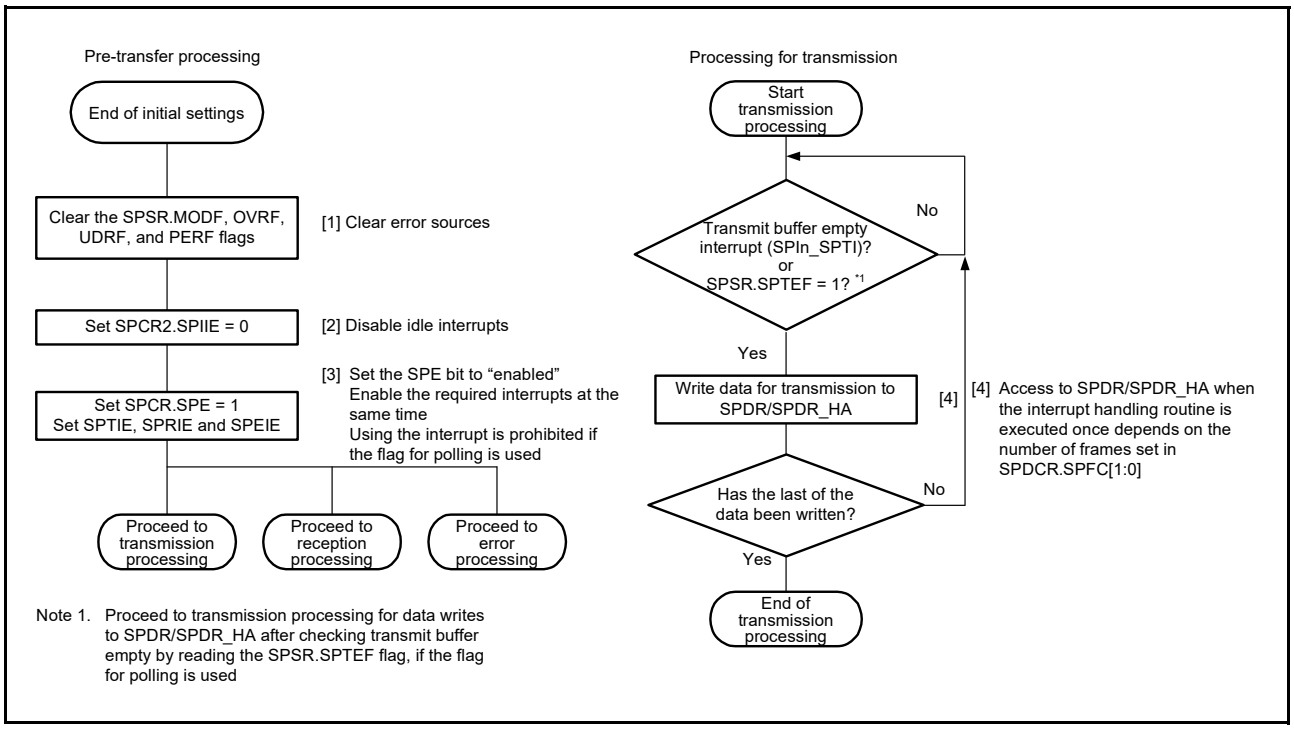


Figure 33.41 Transmission flow in slave mode

(b) Receive Processing Flow

The SPI does not handle receive-only operation, so processing for transmission is required.

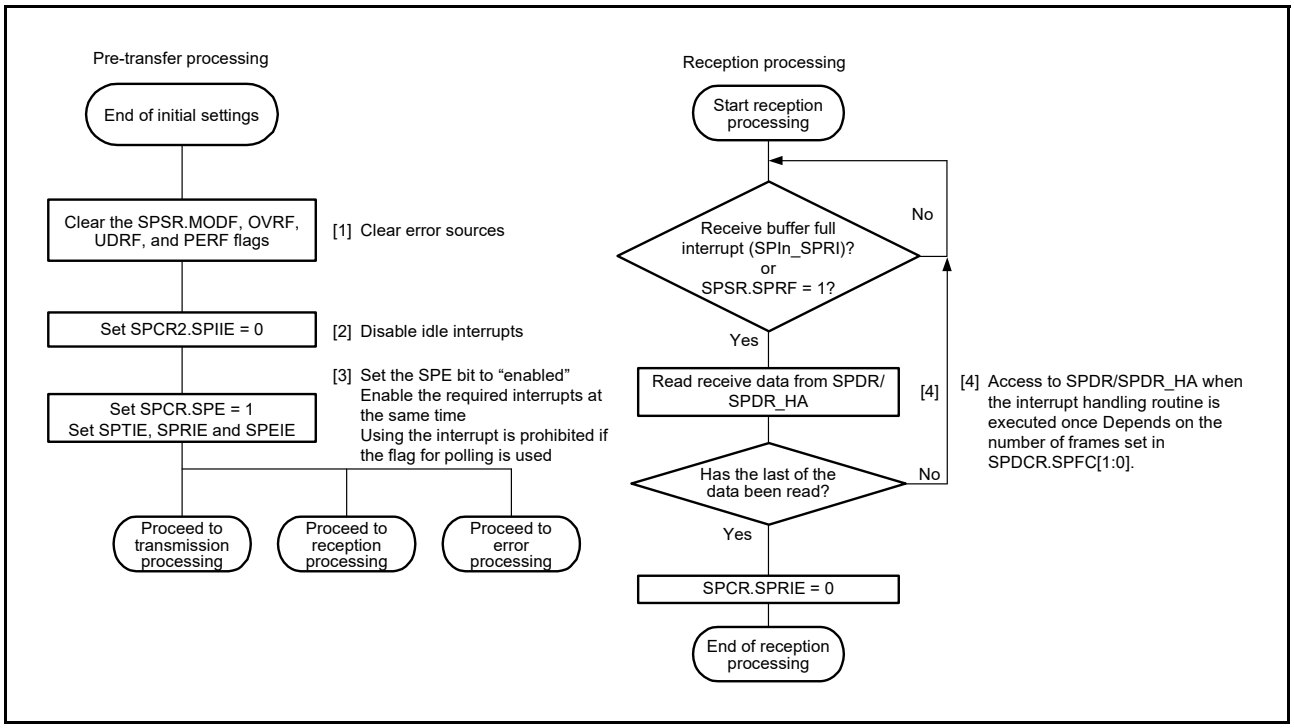


Figure 33.42 Reception flow in slave mode

(c) Flow of Error Processing

In slave 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 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 transmit buffer empty or the receive buffer full interrupt request. If the receive buffer full request is indicated, read the receive buffer and initialize the sequencer in the SPI.

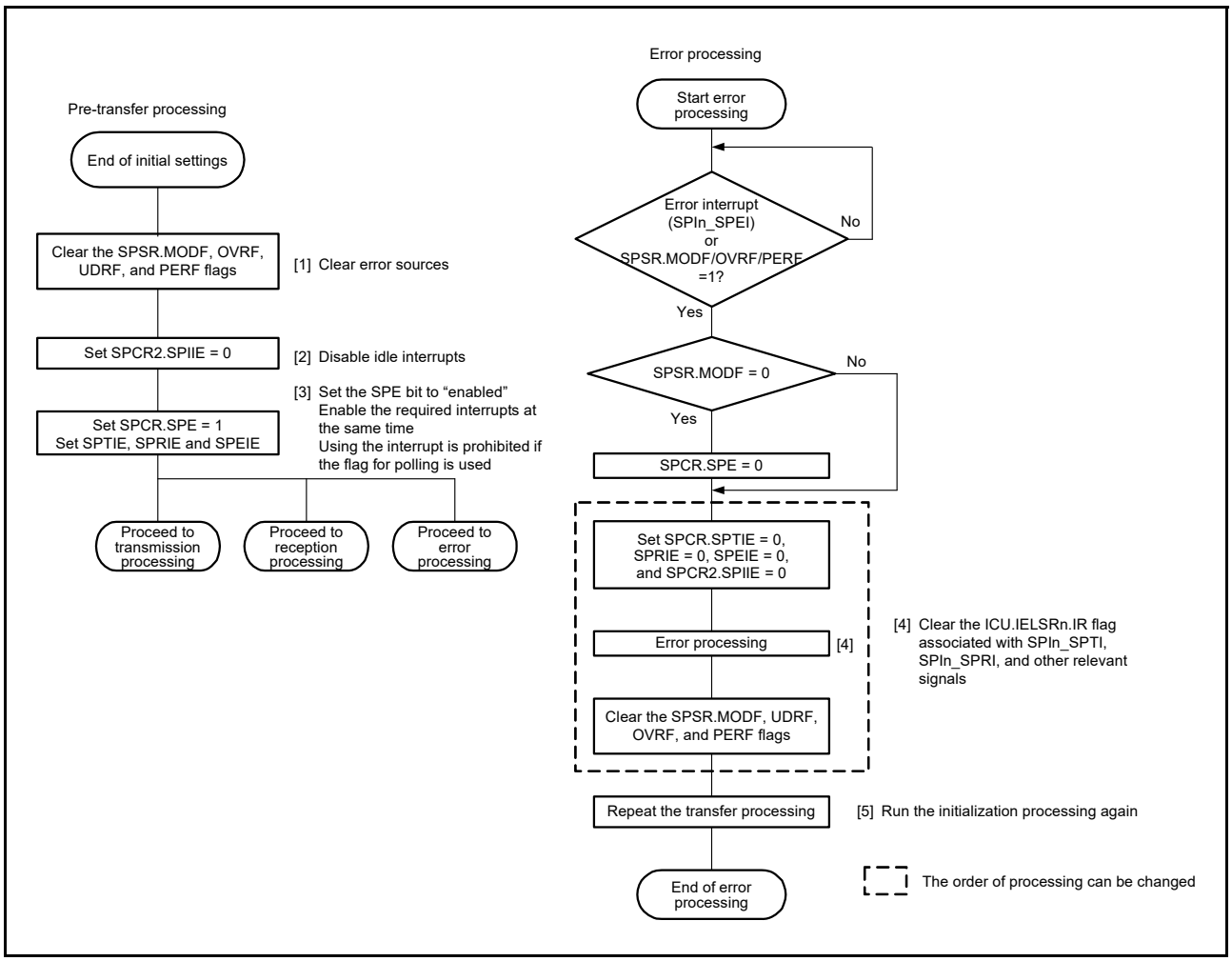


Figure 33.43 Error processing flow for slave mode

33.3.11 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 three pins, RSPCKn, MOSIn, and MISOn, handle communications. Each SSLni pin is available as an I/O port pin.

Although clock synchronous operation does not require the use of the SSLni pin, operation of the module is the same as in SPI operation. That is, in both master and slave operations, communications can be performed with the same flow, except that mode fault errors are not detected because the SSLni pin is not used.

Additionally, do not perform operation if clock synchronous operation proceeds when the SPCMDm.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

33.3.11.1 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 being empty, that is, data for the next transfer is not set, and the SPSR.SPTEF flag is 1. When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR/SPDR_HA, the SPI copies data from the transmission 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.

For details on the SPI transfer format, see section 33.3.5, Transfer Formats. Transfer in clock synchronous operation is conducted without the SSLn0 output signal.

(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 (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).

Note: The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMDm.SPB[3:0] bit setting.

For details on the SPI transfer format, see section 33.3.5, Transfer Formats. Transfer in clock synchronous operation is conducted without the SSLn0 output signal.

(3) Sequence Control

The transfer format in master mode is determined by SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers. Although the SSLni signals are not output in clock synchronous operation, these settings are valid.

The SPSCR register determines the sequence configuration for serial transfers that are executed by the SPI in master mode. The following items are set in the SPCMDm register:

- SSLni output signal value
- MSB-first or LSB-first
- Data length
- Some of the bit rate settings
- RSPCKn polarity/phase
- Whether SPCKD is to be referenced
- Whether SSLND is to be referenced
- Whether SPND is to be referenced.

SPBR holds some of the bit rate settings such as SPCKD, an SPI clock delay value, SSLND, an SSL negation delay, and SPND, a next-access delay value.

According to the sequence length that is assigned to SPSCR, the SPI makes up a sequence comprised of a part or all of SPCMDm register. The SPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the SPI function is enabled, the SPI loads the pointer to the commands in the SPCMD0 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer. The SPI increments the pointer each time the next-access delay period for a data transfer ends. On completion of the serial transfer that corresponds to the final command of the sequence, the SPI sets the pointer to the SPCMD0 register, and the sequence is executed repeatedly.

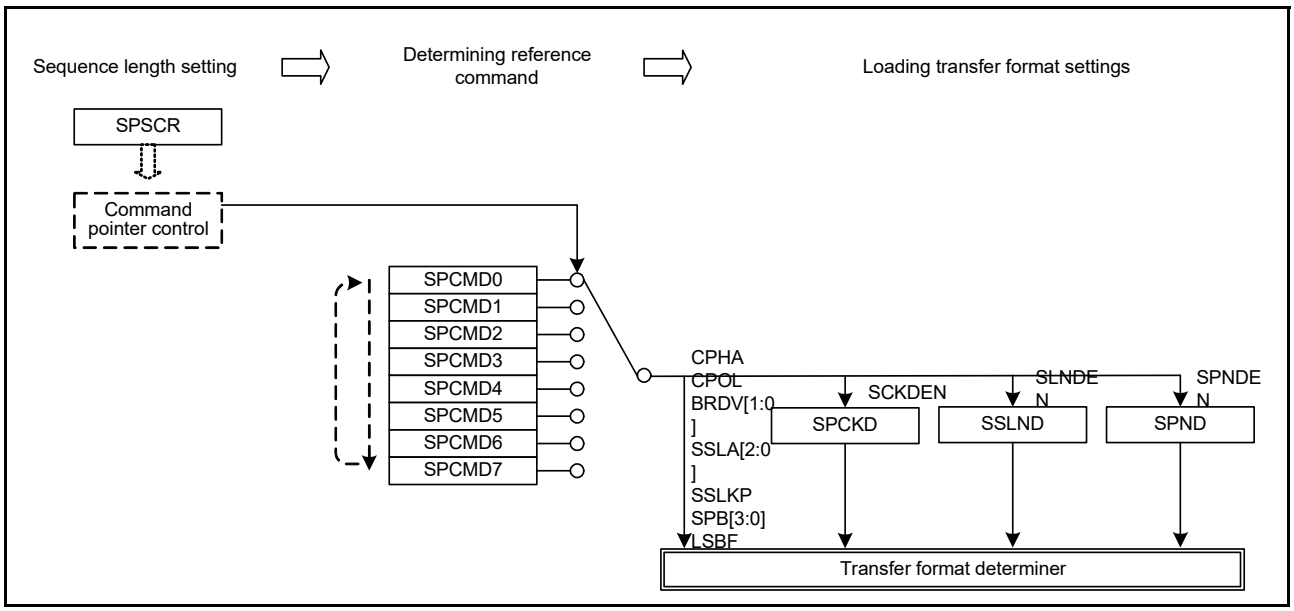


Figure 33.44 Procedure for determining the form of serial transmission in master mode

In this section, a frame is the combination of the data (SPDR/SPDR_HA) and the settings of SPCMDm.

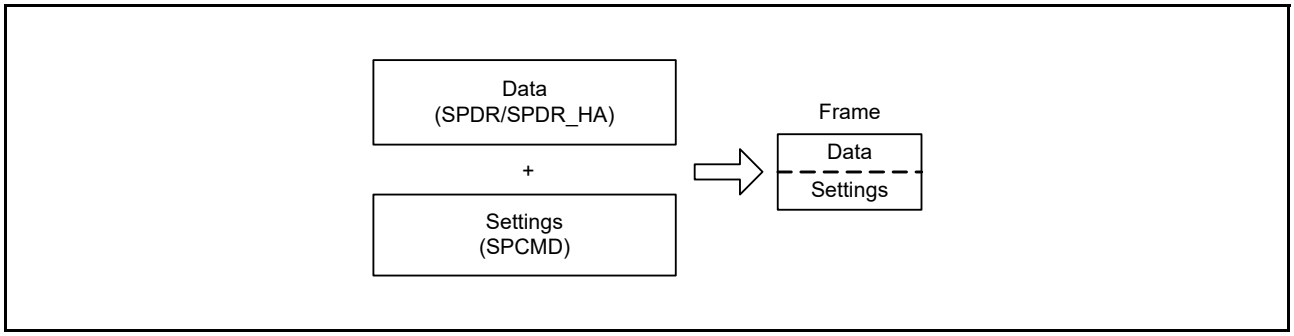


Figure 33.45 Conceptual diagram of frames

Figure 33.46 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 33.4.

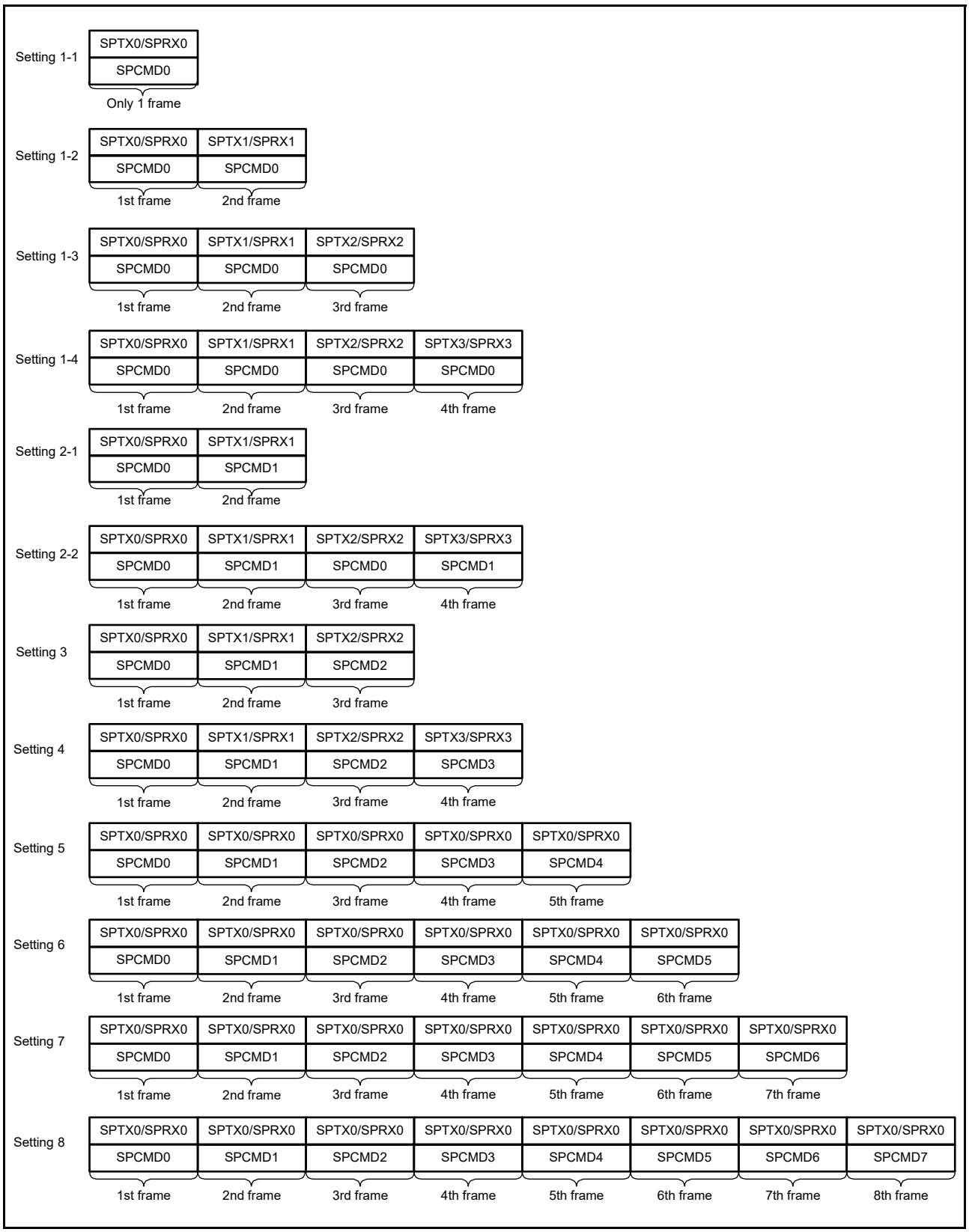


Figure 33.46 Association between SPI Command Register and transmit/receive buffers in sequence operations

(4) Initialization Flow

Figure 33.47 shows an example of initialization flow for clock synchronous operation when the SPI is in master mode. For a description of how to set up the Interrupt Controller Unit, DMAC, and I/O ports, see the individual block descriptions.

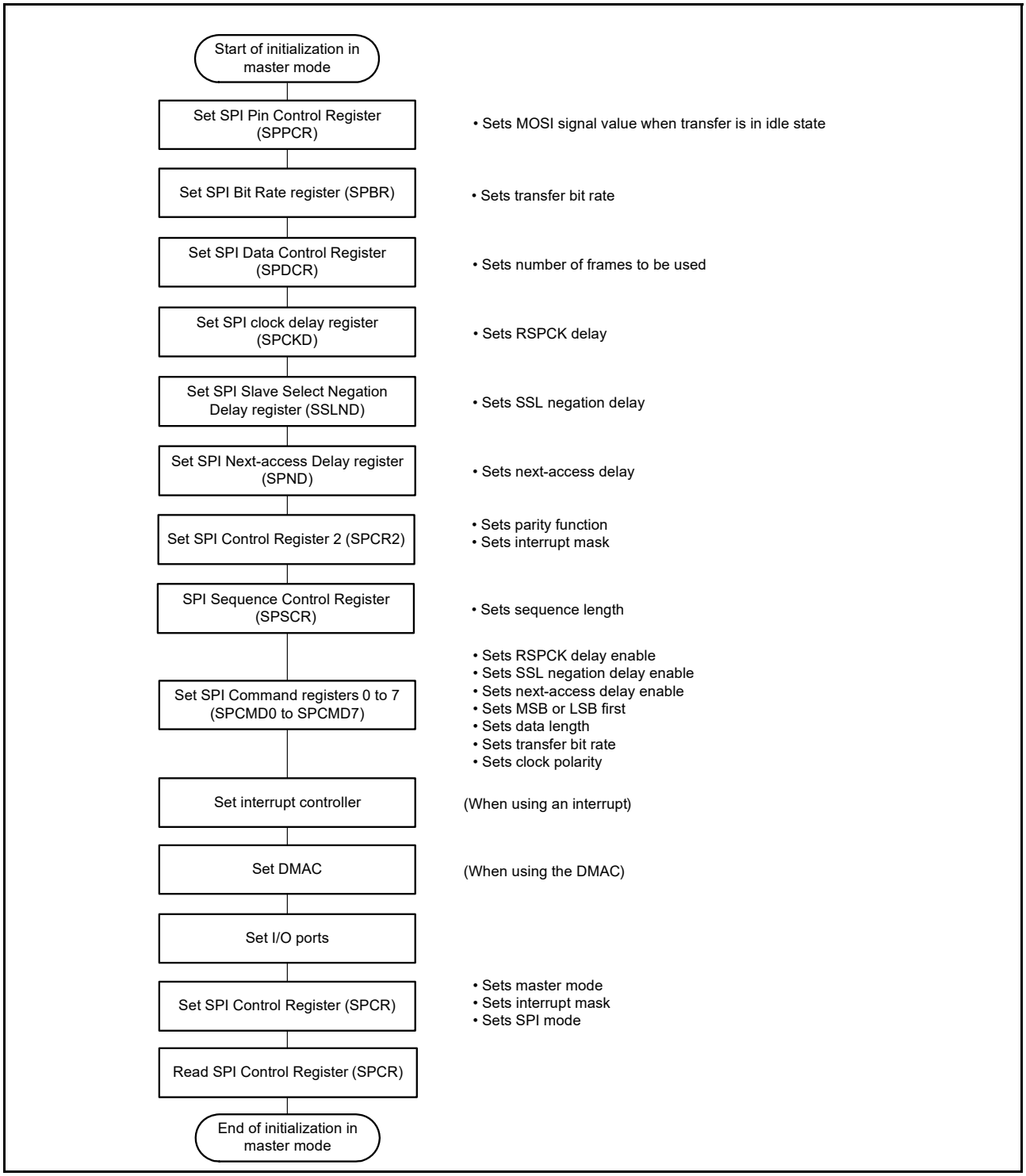


Figure 33.47 Example of initialization flow in master mode for clock synchronous operation

(5) Software Processing Flow

Software processing during clock-synchronous master operation is the same as that for SPI master operation. For details, see section 33.3.10.1, (9) Software Processing Flow.

Note: Mode-fault errors are not generated in this mode.

33.3.11.2 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.

For details on the SPI transfer format, see section 33.3.5, Transfer Formats. The SSLn0 input signal is not used in clock synchronous operation.

(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 (SPSR.SPRF flag is 0), on termination of a 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] bit setting.

For details on the SPI transfer format, see section 33.3.5, Transfer Formats.

(3) Initialization Flow

Figure 33.48 shows an example of initialization flow for clock synchronous operation when the SPI is in slave mode. For a description of how to set up the Interrupt Controller Unit, DMAC, and I/O ports, see the individual block descriptions.

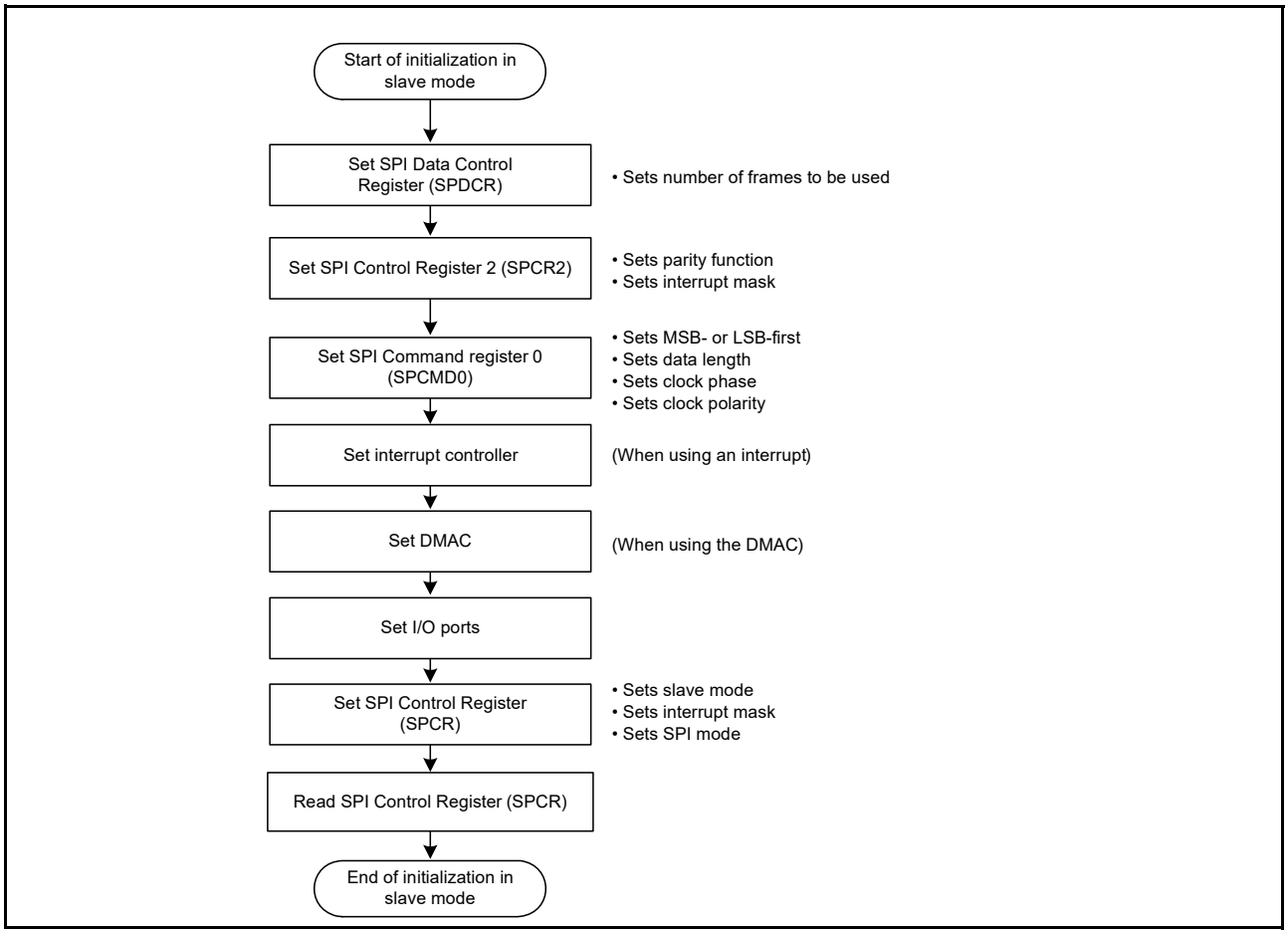


Figure 33.48 Example of initialization flow in slave mode for clock synchronous operation

(4) Software Processing Flow

Software processing during clock-synchronous slave operation is the same as that for SPI slave operation. For details, see section 33.3.10.2, (6) Software Processing Flow. Mode-fault errors do not occur under these conditions.

33.3.12 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 path and output path of the shift register. The SPI does not shut off the path between the MOSIn pin and the shift register if the SPCR.MSTR bit is 1, and 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 33.12 lists the relationship between the SPLP2 and SPLP bits and the received data. Figure 33.49 shows the configuration of the shift register I/O paths where the SPI in master mode is set in loopback mode (SPPCR.SPLP2 = 1, SPPCR.SPLP = 0 or 1).

Table 33.12 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

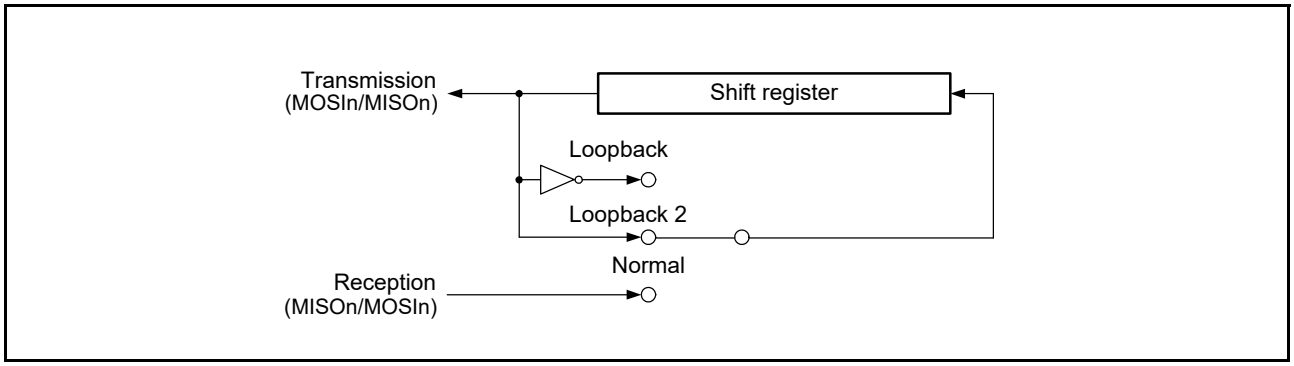


Figure 33.49 Configuration of shift register I/O paths in loopback mode for master mode

33.3.13 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 these units, the parity circuit performs self-diagnosis as shown in Figure 33.50.

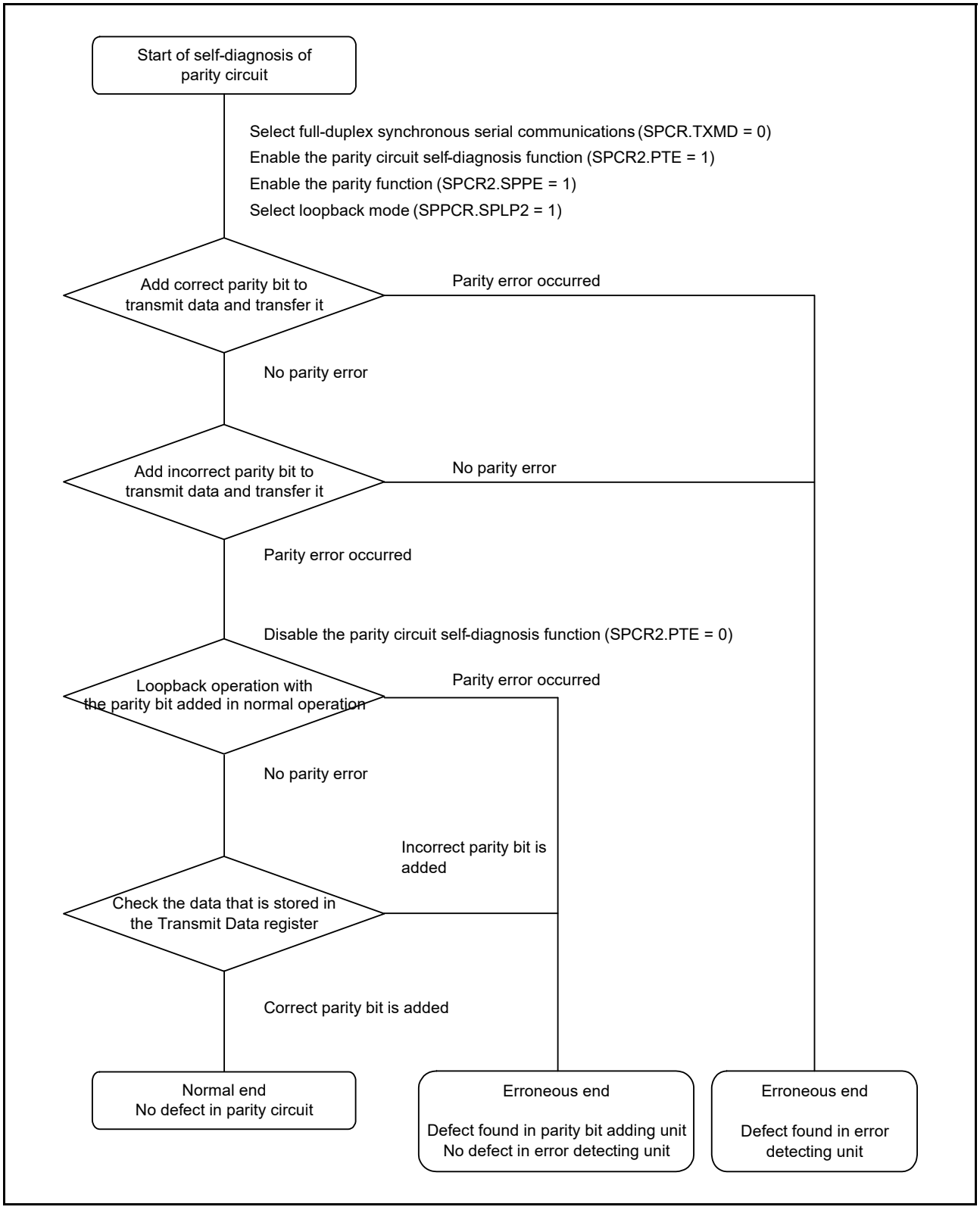


Figure 33.50 Self-diagnosis flow for parity circuit

33.3.14 Interrupt Sources

The SPI has eight interrupt sources:

- Receive buffer full
- Transmit buffer empty
- Transmission complete
- Mode fault
- Underrun
- Overrun
- Parity error
- SPI idle.

In addition, the DTC or DMAC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Because the vector address for SPIn_SPEI is allocated to interrupt requests triggered by mode-fault, underrun, overrun, and parity errors, the actual interrupt source must be determined from the flags. Table 33.13 lists the flags associated with the interrupt sources for the SPI. An interrupt is generated on satisfaction of an interrupt condition in Table 33.13. Clear the receive buffer full and transmit buffer empty sources through data transfer.

When using the DTC or DMAC to perform data transmission or reception, you must first set up the DTC or DMAC to be in transfer-enabled status before setting the SPI. For information on setting the DTC or DMAC, see section 17, DMA Controller (DMAC), or section 18, Data Transfer Controller (DTC).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt are generated while the ICU.IELSRn.IR flag is 1, the interrupt is not output as a request for the ICU but is saved internally (the capacity for retention is one request per source). A saved interrupt request is output when the ICU.IELSRn.IR flag becomes 0. A saved interrupt request is automatically discarded when it is output as an actual interrupt request. The interrupt enable bit (SPCR.SPTIE or SPCR.SPRIE bit) for an internally saved interrupt request can also be set to 0.

Table 33.13 SPI interrupt sources

Interrupt source	Symbol	Interrupt condition	DMAC/DTC activation
Receive buffer full	SPIn_SPRI	The receive buffer becomes full (SPSR.SPRF flag is 1) while the SPCR.SPRIE bit is 1	Possible
Transmit buffer empty	SPIn_SPTI	The transmit buffer becomes empty (SPSR.SPTEF flag is 1) while the SPCR.SPTIE bit is 1	Possible
SPI errors (mode fault, underrun, overrun and parity error)	SPIn_SPEI	The SPSR.MODF, OVRF, PERF or UDRF flag is set to 1 while the SPCR.SPEIE bit is 1	Impossible
SPI idle	SPIn_SPII	The SPSR.IDLNF flag is set to 0 while the SPCR2.SPIIE bit is 1	Impossible
Transmission-complete	SPIn_SPTEND	In master mode, an interrupt is generated when the IDLNF flag (SPI idle flag) changes from 1 to 0. In slave mode, an interrupt occurs with conditions shown in Table 33.15.	Impossible

33.4 Event Link Operation

The Event Link Controller (ELC) is capable of producing 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-complete event output.

The event link output signal is output regardless of the interrupt enable bit setting.

33.4.1 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.

33.4.2 Transmit Buffer Empty Event Output

This event signal is output when data for transmission is transferred from the transmission buffer to the shift register and when the value of the SPE bit changes from 0 to 1.

33.4.3 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 33.5.4, Constraint on Mode Fault, Underrun, Overrun or Parity Error Event Output if using this event signal.

(1) Mode Fault

Table 33.14 lists the conditions for occurrence of a mode fault event.

Table 33.14 Conditions for occurrence of mode fault

Conditions	SPCR.MODFEN bit	SSLn0 pin	Remarks
SPI operation (SPMS = 0) Slave (SPCR.MSTR bit = 0)	1	Not active	Event is output only when the 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 SPCR.MSTR bit is 0 and 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 reception 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.

33.4.4 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).

33.4.5 Transmission-Complete 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 33.15 Conditions for generation of transmission-complete event in slave mode

Conditions	Transmit buffer state	Shift register state	Others
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 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 or the underrun error.

33.5 Usage Notes

33.5.1 Settings for the Module-Stop State

The Module Stop Control Register B (MSTPCRB) can enable or disable SPI operation. The SPI is initially stopped after a reset. The registers become accessible on release from the module-stop state. For details on the Module Stop Control Register B, see section 11, Low Power Modes.

33.5.2 Constraint on Low Power Function

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.

33.5.3 Constraint on Starting Transfer

If the ICU.IELSRn.IR flag is 1 at the time transfer is to start, an interrupt request is internally saved after transfer starts, and this 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 (SPCR.SPE is 0).
2. Set the relevant interrupt enable bit (SPCR.SPTIE or SPCR.SPRIE) to 0.
3. Read the relevant interrupt enable bit (SPCR.SPTIE or SPCR.SPRIE) and confirm that its value is 0.
4. Set the ICU.IELSRn.IR flag to 0.

33.5.4 Constraint 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 multi-master mode (the SPCR.SPMS bit is 0, the SPCR.MSTR bit is 1, and the SPCR.MODFEN bit is 1).

33.5.5 Constraint on the SPRF and SPTEF Flags

Using an interrupt is prohibited (set the SPCR.SPRIE and SPCR.SPTIE bits to 0.) if your application uses the flag for polling. Either the interrupt or the flag can be used, not both.

34. Quad Serial Peripheral Interface (QSPI)

34.1 Overview

The Quad Serial Peripheral Interface module (QSPI) is a memory controller for connecting a serial ROM (nonvolatile memory such as a serial flash memory, serial EEPROM, or serial FeRAM) that has an SPI-compatible interface.

Table 34.1 lists the QSPI specifications, Figure 34.1 shows the block diagram, and Table 34.2 lists the I/O pins.

Table 34.1 QSPI specifications

Parameter	Specification
Number of channels	1 channel
SPI	• Support for extended SPI, dual SPI, and quad SPI protocols • Configurable to SPI mode 0 and SPI mode 3 • Address width selectable from 8, 16, 24, or 32 bits.
Timing adjustment function	Configurable to support a wide range of serial flash
Flash read function	• Support for the Read, Fast Read, Fast Read Dual Output, Fast Read Dual I/O, Fast Read Quad Output, and Fast Read Quad I/O instructions • Substitutable Instruction code • Adjustable number of dummy cycles • A prefetch function • Polling processing • SPI bus cycle extension function.
Direct communication function	Flexible support for a wide variety of serial flash instructions and functions through software control, including erase, write, ID read, and power-down control
Interrupt source	Error interrupt
Module stop function	Module-stop state can be set

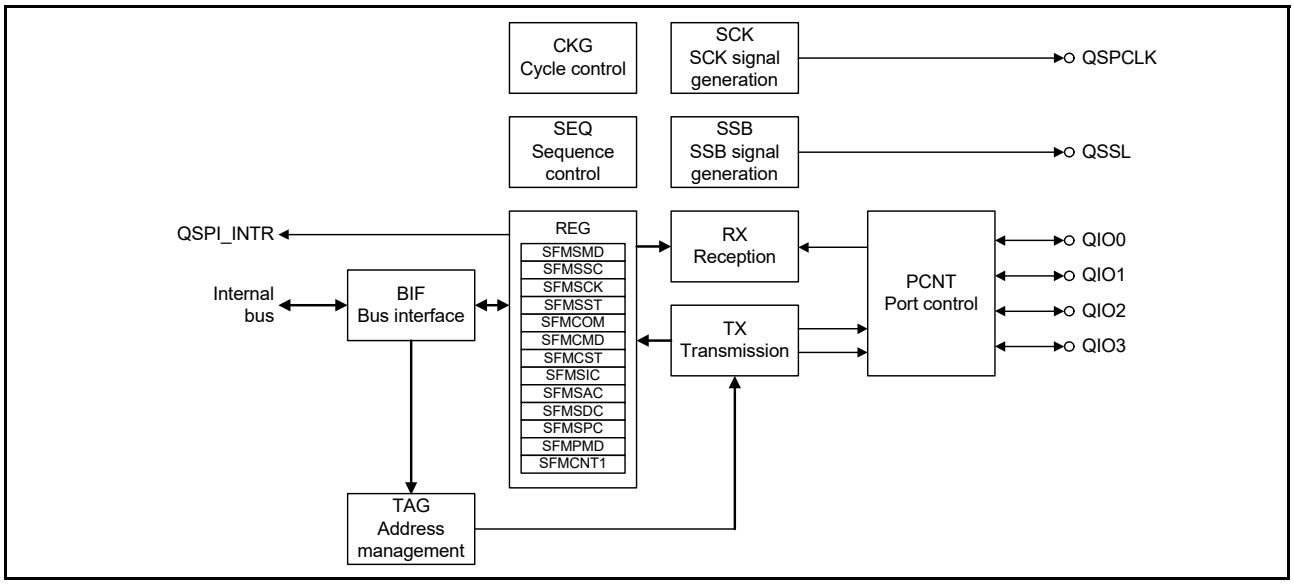


Figure 34.1 QSPI block diagram

Table 34.2 QSPI I/O pins

Pin name	I/O	Function
QSPCLK	Output	QSPI clock output pin
QSSL	Output	QSPI slave select pin
QIO0	I/O	Data 0 I/O
QIO1	I/O	Data 1 I/O
QIO2	I/O	Data 2 I/O
QIO3	I/O	Data 3 I/O

34.2 Register Descriptions

34.2.1 Transfer Mode Control Register (SFMSMD)

Address(es): QSPI.SFMSMD 6400 0000h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SFMCCE	—	—	—	SFMOSW	SFMOHW	SFMOEX	SFMMD3	SFMPAE	SFMPFE	SFMSE[1:0]		—	SFMRM[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Symbol	Bit name	Description	R/W
b2 to b0	SFMRM[2:0]	Serial interface read mode select	b2 b0 0 0 0: Standard Read 0 0 1: Fast Read 0 1 0: Fast Read Dual Output 0 1 1: Fast Read Dual I/O 1 0 0: Fast Read Quad Output 1 0 1: Fast Read Quad I/O 1 1 0: Setting prohibited (an unpredictable operation can result) 1 1 1: Setting prohibited (an unpredictable operation can result).	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	SFMSE[1:0]	QSSL extension function select after SPI bus access	b5 b4 0 0: Do not extend QSSL 0 1: Extend QSSL by 33 QSPCLK 1 0: Extend QSSL by 129 QSPCLK 1 1: Extend QSSL infinitely.	R/W
b6	SFMPFE	Prefetch function select	0: Disable prefetch 1: Enable prefetch.	R/W
b7	SFMPAE	Function select for stopping prefetch at locations other than on byte boundaries	0: Disable Function 1: Enable Function.	R/W
b8	SFMMD3	SPI mode select. An initial value is determined by input to CFGMD3	0: SPI mode 0 1: SPI mode 3.	R/W
b9	SFMOEX	Extension select for the I/O buffer output enable signal for the serial interface	0: Do not extend 1: Extend by 1 QSPCLK.	R/W
b10	SFMOHW	Hold time adjustment for serial transmission	0: Do not extend high-level width of QSPCLK during transmission 1: Extend high-level width of QSPCLK by 1 PCLKA during transmission.	R/W
b11	SFMOSW	Setup time adjustment for serial transmission	0: Do not extend low-level width of QSPCLK during transmission. 1: Extend low-level width of QSPCLK by 1 PCLKA during transmission.	R/W
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	SFMCCE	Read instruction code select	0: Default instruction code set for each instruction 1: Instruction code written in the SFMSIC register.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

34.2.2 Chip Selection Control Register (SFMSSC)

Address(es): QSPI.SFMSSC 6400 0004h

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

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

Bits	Symbol	Bit name	Description	R/W
b3 to b0	SFMSW	Minimum high-level width select for QSSL signal	b3 b2 b1 b0 0 0 0 0: 1 QSPCLK 0 0 0 1: 2 QSPCLK 0 0 1 0: 3 QSPCLK 0 0 1 1: 4 QSPCLK 0 1 0 0: 5 QSPCLK 0 1 0 1: 6 QSPCLK 0 1 1 0: 7 QSPCLK 0 1 1 1: 8 QSPCLK 1 0 0 0: 9 QSPCLK 1 0 0 1: 10 QSPCLK 1 0 1 0: 11 QSPCLK 1 0 1 1: 12 QSPCLK 1 1 0 0: 13 QSPCLK 1 1 0 1: 14 QSPCLK 1 1 1 0: 15 QSPCLK 1 1 1 1: 16 QSPCLK.	R/W
b4	SFMSHD	QSSL signal release timing select	0: Release QSSL 0.5 QSPCLK after the last rising edge of QSPCLK 1: Release QSSL 1.5 QSPCLK after the last rising edge of QSPCLK.	R/W
b5	SFMSLD	QSSL signal output timing select	0: Output QSSL 0.5 QSPCLK before the first rising edge of QSPCLK 1: Output QSSL 1.5 QSPCLK before the first rising edge of QSPCLK.	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

34.2.3 Clock Control Register (SFMSKC)

Address(es): QSPI.SFMSKC 6400 0008h

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

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

Bits	Symbol	Bit name	Description	R/W
b4 to b0	SFMDV	Serial interface reference cycle select (pay attention to the irregularity)	b4 b3 b2 b1 b0 0 0 0 0 0: 2 PCLKA 0 0 0 0 1: 3 PCLKA (multiplied by an odd number)*1 0 0 0 1 0: 4 PCLKA 0 0 0 1 1: 5 PCLKA (multiplied by an odd number)*1 0 0 1 0 0: 6 PCLKA 0 0 1 0 1: 7 PCLKA (multiplied by an odd number)*1 0 0 1 1 0: 8 PCLKA 0 0 1 1 1: 9 PCLKA (multiplied by an odd number)*1 0 1 0 0 0: 10 PCLKA 0 1 0 0 1: 11 PCLKA (multiplied by an odd number)*1 0 1 0 1 0: 12 PCLKA 0 1 0 1 1: 13 PCLKA (multiplied by an odd number)*1 0 1 1 0 0: 14 PCLKA 0 1 1 0 1: 15 PCLKA (multiplied by an odd number)*1 0 1 1 1 0: 16 PCLKA 0 1 1 1 1: 17 PCLKA (multiplied by an odd number)*1 1 0 0 0 0: 18 PCLKA 1 0 0 0 1: 20 PCLKA 1 0 0 1 0: 22 PCLKA 1 0 0 1 1: 24 PCLKA 1 0 1 0 0: 26 PCLKA 1 0 1 0 1: 28 PCLKA 1 0 1 1 0: 30 PCLKA 1 0 1 1 1: 32 PCLKA 1 1 0 0 0: 34 PCLKA 1 1 0 0 1: 36 PCLKA 1 1 0 1 0: 38 PCLKA 1 1 0 1 1: 40 PCLKA 1 1 1 0 0: 42 PCLKA 1 1 1 0 1: 44 PCLKA 1 1 1 1 0: 46 PCLKA 1 1 1 1 1: 48 PCLKA.	R/W
b5	SFMDTY	Duty ratio correction function select for the QSPCLK signal	0: Make no correction 1: Delay the rising of the QSPCLK signal by 0.5 PCLKA (valid with PCLKA multiplied by an odd number).	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When PCLKA multiplied by an odd number is selected, the high-level width of the QSPCLK signal is longer than the low-level width by 1 PCLKA before duty ratio correction.

34.2.4 Status Register (SFMSST)

Address(es): QSPI.SFMSST 6400 000Ch

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

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

Bits	Symbol	Bit name	Description	R/W
b4 to b0	PFCNT	Number of bytes of prefetched data	b4 b0 0 0 0 0 0: 0 bytes 0 0 0 0 1: 1 byte 0 0 0 1 0: 2 bytes 0 0 0 1 1: 3 bytes 0 0 1 0 0: 4 bytes 0 0 1 0 1: 5 bytes 0 0 1 1 0: 6 bytes 0 0 1 1 1: 7 bytes 0 1 0 0 0: 8 bytes 0 1 0 0 1: 9 bytes 0 1 0 1 0: 10 bytes 0 1 0 1 1: 11 bytes 0 1 1 0 0: 12 bytes 0 1 1 0 1: 13 bytes 0 1 1 1 0: 14 bytes 0 1 1 1 1: 15 bytes 1 0 0 0 0: 16 bytes 1 0 0 0 1: 17 bytes 1 0 0 1 0: 18 bytes Other settings are reserved.	R
b5	—	Reserved	This bit is read as 0.	R
b6	PFFUL	Prefetch buffer state	0: Prefetch buffer has free space 1: Prefetch buffer is full.	R
b7	PFOFF	Prefetch function operation state	0: Prefetch function operating 1: Prefetch function not enabled or not operating.	R
b31 to b8	—	Reserved	These bits are read as 0.	R

34.2.5 Communication Port Register (SFMCOM)

Address(es): QSPI.SFMCOM 6400 0010h

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

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

x: Undefined

Bits	Symbol	Bit name	Description	R/W
b7 to b0	SFMD	Port for direct communication with the SPI bus	Input to and output from this port is converted to an SPI bus cycle. This port is accessible in the direct communication mode, when DCOM = 1. Access to this port is ignored in the ROM access mode.	R/W
b 31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

34.2.6 Communication Mode Control Register (SFMCMD)

Address(es): QSPI.SFMCMD 6400 0014h

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

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

Bits	Symbol	Bit name	Description	R/W
b0	DCOM	Mode select for communication with the SPI bus	0: ROM access mode 1: Direct communication mode.	R/W
b 31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

34.2.7 Communication Status Register (SFMCST)

Address(es): QSPI.SFMCST 6400 0018h

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

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

Bits	Symbol	Bit name	Description	R/W
b0	COMBSY	SPI bus cycle completion state in direct communication	0: No serial transfer being processed 1: Serial transfer being processed.	R
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	EROMR	ROM access detection status in direct communication mode	0: ROM access not detected 1: ROM access detected.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

34.2.8 Instruction Code Register (SFMSIC)

Address(es): QSPI.SFMSIC 6400 0020h

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

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

Bits	Symbol	Bit name	Description	R/W
b7 to b0	SFMCIC	Serial flash instruction code to substitute		R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

34.2.9 Address Mode Control Register (SFMSAC)

Address(es): QSPI.SFMSAC 6400 0024h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	SFM4BC	—	—	SFMAS	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bits	Symbol	Bit name	Description	R/W
b1, b0	SFMAS	Number of address bytes select for the serial interface	b1 b0 0 0: 1 byte 0 1: 2 bytes 1 0: 3 bytes 1 1: 4 bytes.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SFM4BC	Default instruction code select, when serial interface address width is 4 bytes	0: Do not use 4-byte address read instruction code 1: Use 4-byte address read instruction code.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

34.2.10 Dummy Cycle Control Register (SFMSDC)

Address(es): QSPI.SFMSDC 6400 0028h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SFMXD								SFMXEN	SFMXST	—	—	SFMDN[3:0]			
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bits	Symbol	Bit name	Description	R/W
b3 to b0	SFMDN[3:0]	Number of dummy cycles select for Fast Read instructions	b3 b2 b1 b0 0 0 0 0:Default dummy cycles of each instruction: - Fast Read Quad I/O:6 QSPCLK - Fast Read Quad Output:8 QSPCLK - Fast Read Dual I/O:4 QSPCLK - Fast Read Dual Output:8 QSPCLK - Fast Read:8 QSPCLK. 0 0 0 1:3 QSPCLK*1 0 0 1 0:4 QSPCLK 0 0 1 1:5 QSPCLK 0 1 0 0:6 QSPCLK 0 1 0 1:7 QSPCLK 0 1 1 0:8 QSPCLK 0 1 1 1:9 QSPCLK 1 0 0 0:10 QSPCLK 1 0 0 1:11 QSPCLK 1 0 1 0:12 QSPCLK 1 0 1 1:13 QSPCLK 1 1 0 0:14 QSPCLK 1 1 0 1:15 QSPCLK 1 1 1 0:16 QSPCLK 1 1 1 1:17 QSPCLK.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SFMXST	XIP mode status	0: Normal (non-XIP) mode 1: XIP mode.	R
b7	SFMXEN	XIP mode permission	0: XIP mode prohibited 1: XIP mode permitted.	R/W
b15 to b8	SFMXD	Mode data for serial flash (control XIP mode)		R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To avoid a conflict with the input/output switch of the serial flash pin connected to QIO0 pin, select more than 4 QSPCLK dummy cycles when the output enable signal is extended by setting the SFMOEX bit of the SFMSMD register to 1.

34.2.11 SPI Protocol Control Register (SFMSPC)

Address(es): QSPI.SFMSPC 6400 0030h

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

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

Bits	Symbol	Bit name	Description	R/W
b1, b0	SFMSPI	SPI protocol select	b1 b0 0 0: Extended SPI protocol 0 1: Dual SPI protocol 1 0: Quad SPI protocol 1 1: Setting prohibited.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SFMSDE	Minimum time select for input output switch, when dual SPI protocol or quad SPI protocol is selected and in standard read mode	0: Do not allocate minimum switch time 1: Allocate minimum switch time equivalent to 1 QSPCLK.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

34.2.12 Port Control Register (SFMPMD)

Address(es): QSPI.SFMPMD 6400 0034h

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

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

Bits	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	SFMWPL	WP pin specification	0: Low level 1: High level.	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

34.2.13 External QSPI Address Register (SFMCNT1)

Address(es): QSPI.SFMCNT1 6400 0804h

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

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

Bits	Symbol	Bit name	Description	R/W
b25 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b26	QSPI_EXT[5:0]	Bank switching address When accessing from 6000 0000h to 63FF FFFFh, address bus is set from QSPI_EXT[5:0] to upper 6 bits of internal bus address.		R/W

34.3 Memory Map

34.3.1 Internal Bus Space

The locations of a serial flash and control register on the AHB space are determined by the address range of the area set in the configuration.

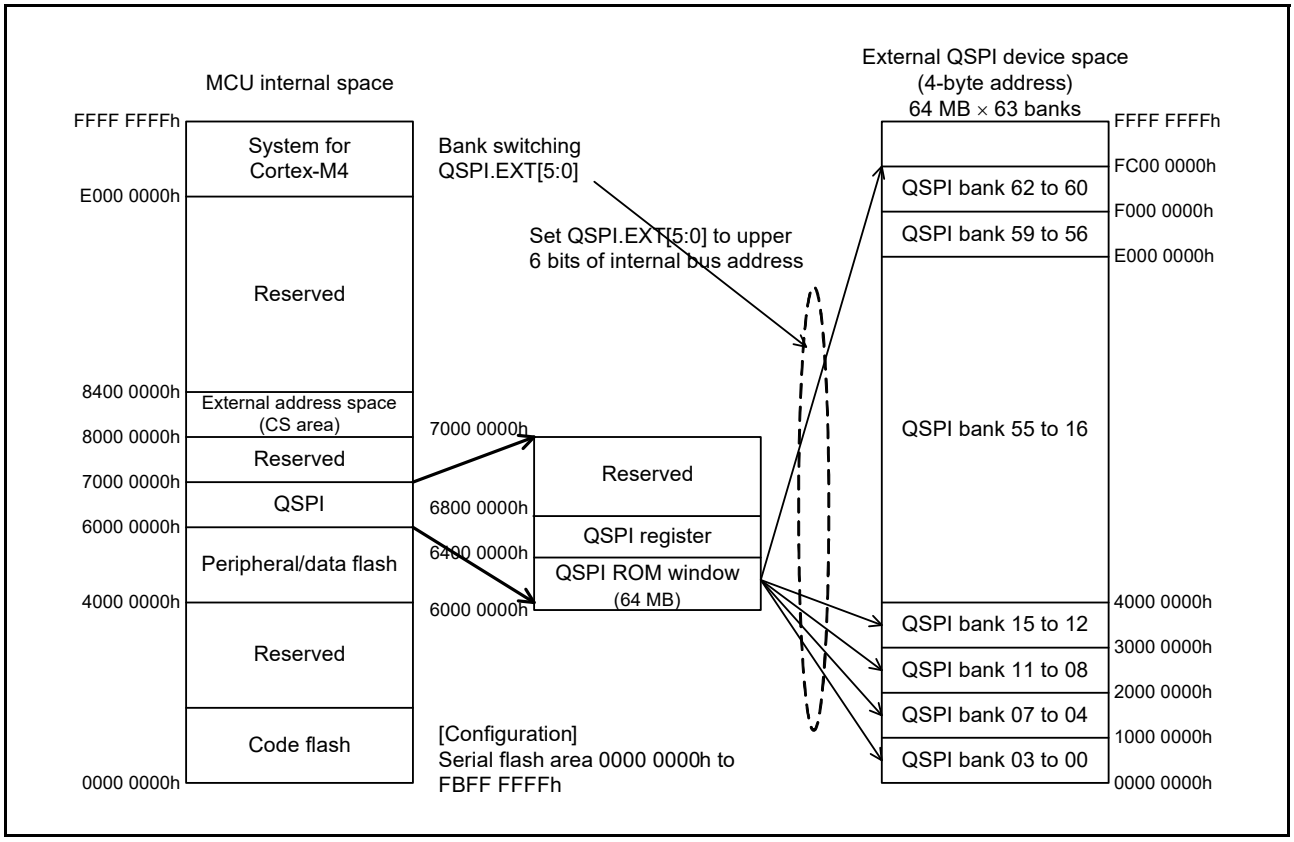


Figure 34.2 Default area setting and AHB space memory map

34.3.2 Address Width of the SPI Space and SPI Bus

The SPI space has a 32-bit address width for referencing the serial flash. When the SPI space is accessed for a read, an SPI bus cycle automatically starts, and data read from the serial flash is returned.

The address width of the SPI space is fixed at 32 bits. However, the address width of the SPI bus is selectable from 8, 16, 24, or 32 bits in the SFMAS[1:0] bits of the SFMSAC register.

If 8, 16, or 24 bits is selected as the address width of the SPI bus, only the lower part of the address used to access the SPI space is posted to the serial flash through the SPI bus. As a result, the mirror image of the serial flash associated with the address width of the SPI bus repeatedly appears in the SPI space.

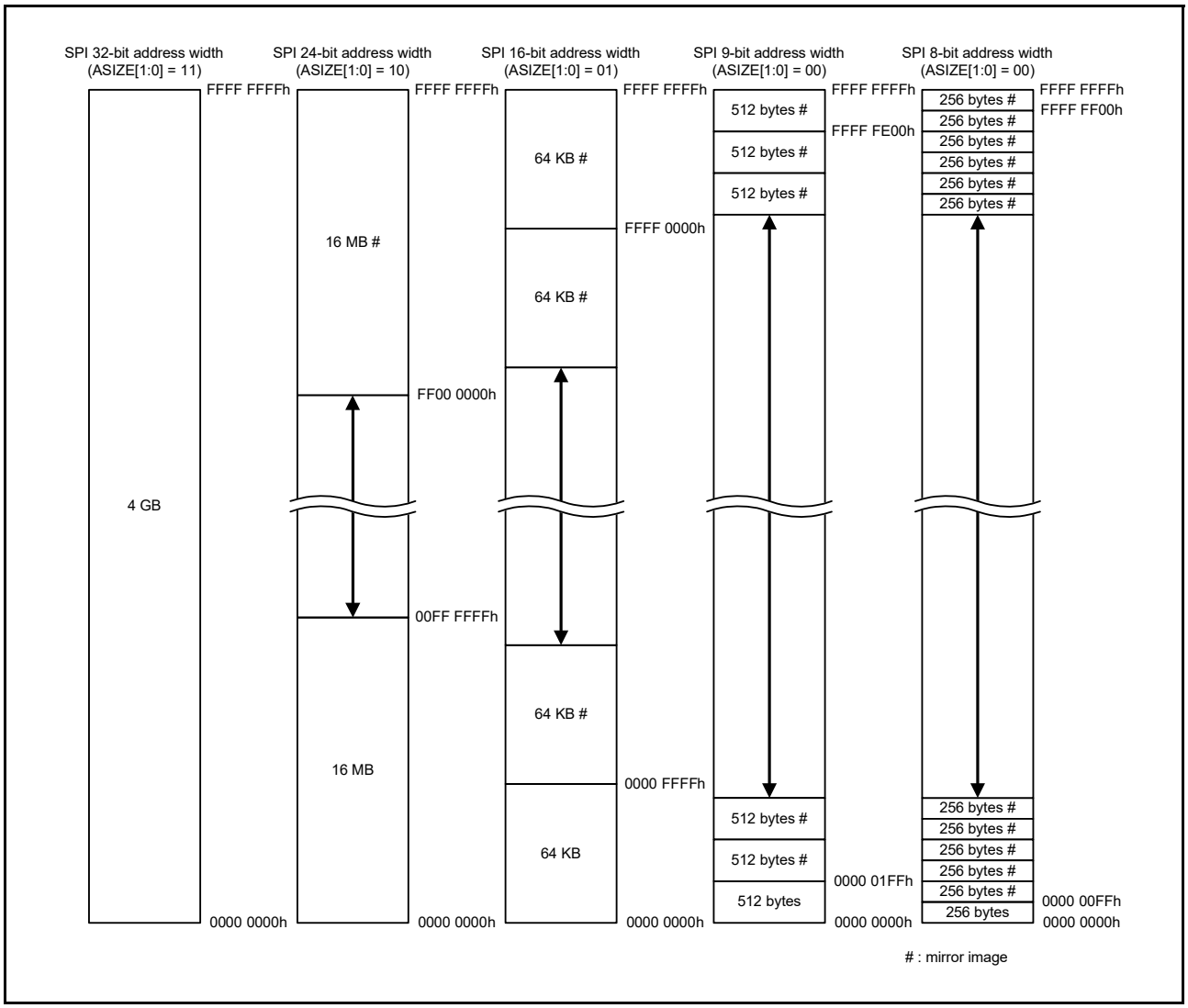


Figure 34.3 Memory map of SPI space

Note: The SPI bus address width is selected to 32 bits, 24 bits, 16 bits, and 8 bits with the SFMAS[1:0] bits of the SFMSAC register. When an 8-bit address width is selected, the address information of the 9th bit can be embedded in the Read instruction code. The address map in the figure is for the SPI 9-bit address width. For details of the Read instruction, see section 34.6.2, Standard Read Instruction.

34.4 SPI Bus

34.4.1 SPI Protocol

Extended SPI, dual SPI, and quad SPI are supported in addition to the SPI protocol used for serial flash connection.

The initial state of the SPI protocol is extended SPI and can be changed in the SFMSPI bit in the SFMSPC register.

The Extended SPI protocol always outputs instruction codes from a single QIO0 pin. It performs subsequent address and data I/O operation using one to four pins, depending on the instruction code format.

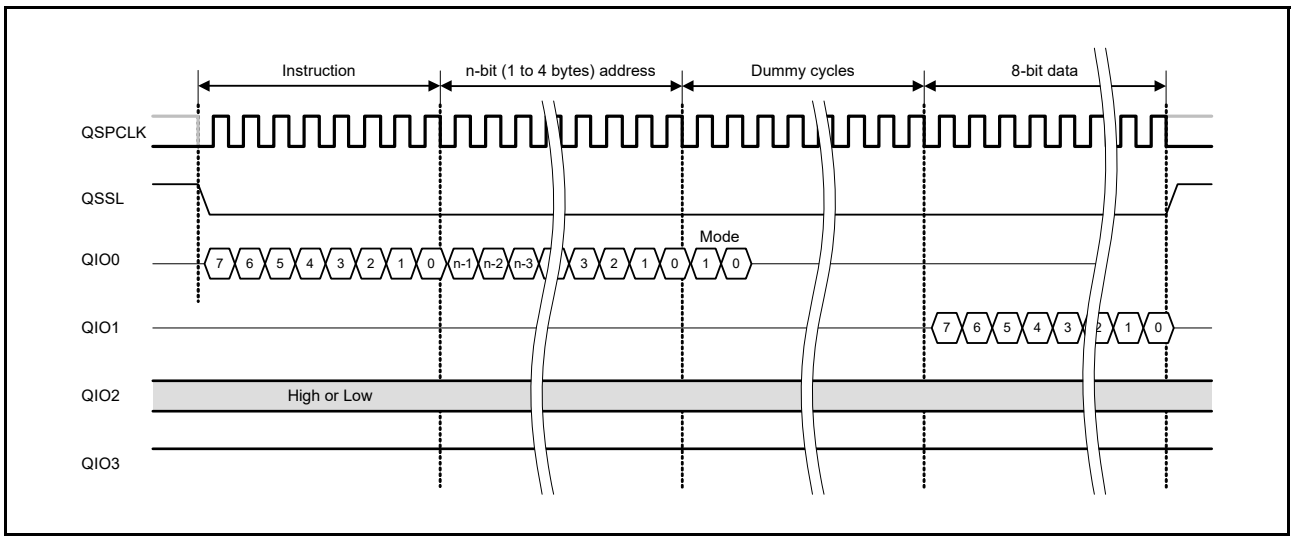


Figure 34.4 Extended SPI protocol example 1 for Fast Read

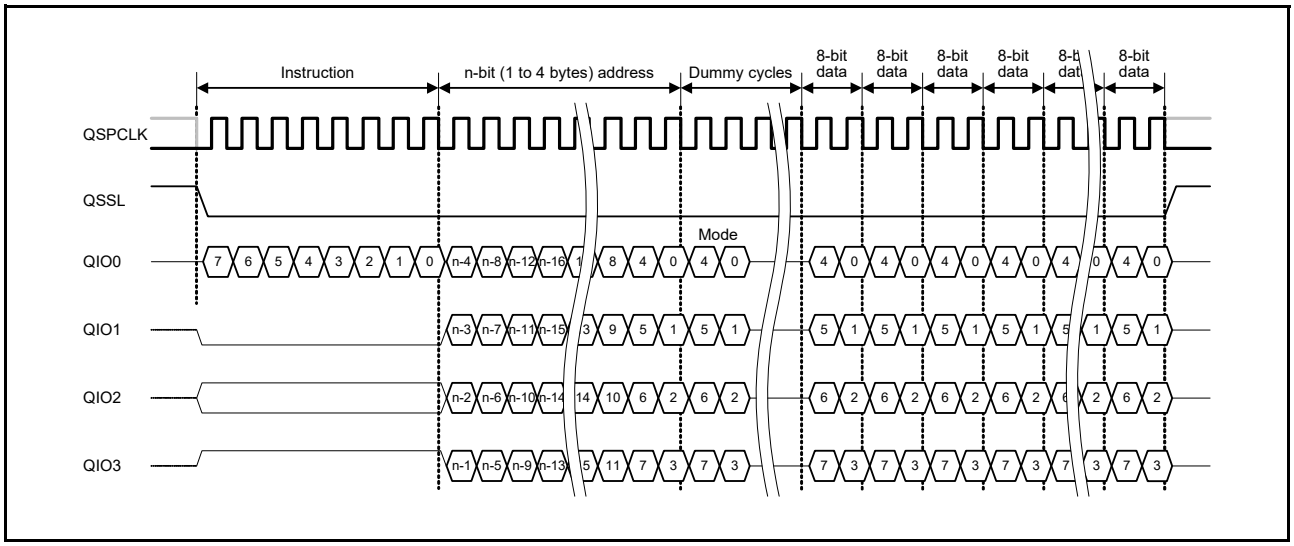


Figure 34.5 Extended SPI protocol example 2 for Fast Read Quad I/O

The Dual SPI protocol performs I/O operation of all signals such as instruction codes, addresses, and data using two pins, QIO0 and QIO1.

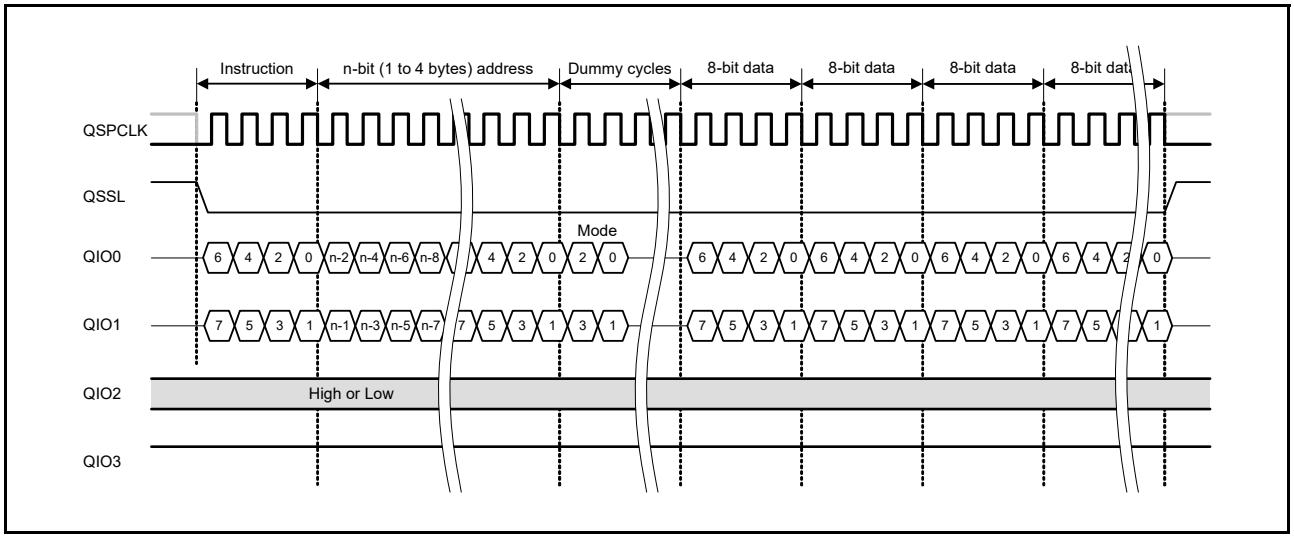


Figure 34.6 Dual SPI protocol example for Fast Read

The Quad SPI protocol performs I/O operation of all signals such as instruction codes, addresses, and data using four pins, QIO0, QIO1, QIO2, and QIO3.

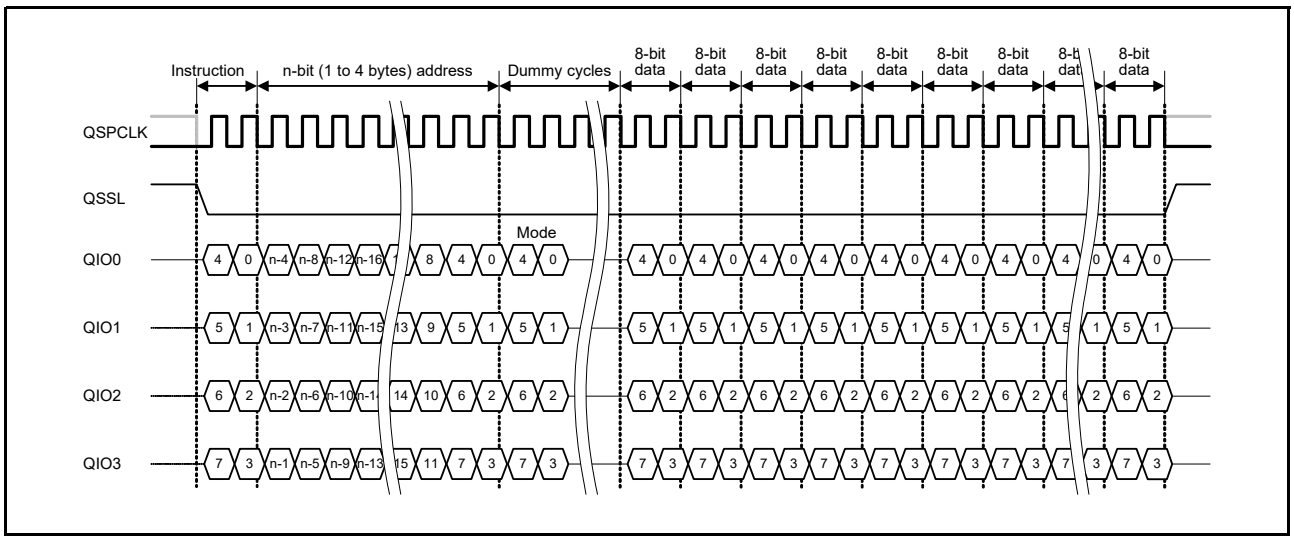


Figure 34.7 Quad SPI protocol example for Fast Read

34.4.2 SPI Mode

The initial SPI mode is set to SPI mode 0 or 3 by the CFGMD3 pin. This can be switched by changing the register setting during operation. The difference between SPI mode 0 and 3 is the standby level of the QSPCLK signal. The standby level of the QSPCLK signal in SPI mode 0 is low, and high in SPI mode 3.

Serial data is output from the QSPI on a falling edge of the serial clock and is read into the external flash on a rising edge of the serial clock. Serial data is output from the external flash on a falling edge of the serial clock and is read into the QSPI on the next falling edge of the serial clock.

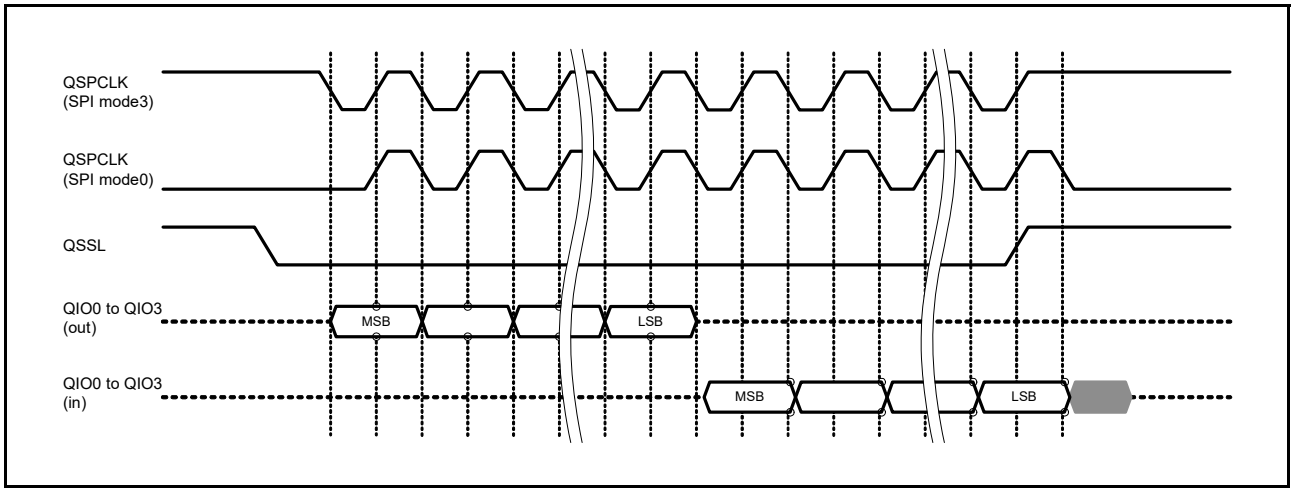


Figure 34.8 Basic timing of serial interface

34.5 SPI Bus Timing Adjustment

The timing of the SPI bus signal can be adjusted in the registers. The configured timing is applied to all SPI bus accesses, for both ROM access and direct communication.

34.5.1 SPI Bus Reference Cycles

The SPI bus operates on reference cycles obtained by multiplying PCLKA by an integer. The reference cycles are selectable within the range from PCLKA multiplied by 2 to PCLKA multiplied by 48, by setting the SFMDV[4:0] bits in the SFMSKC register.

Table 34.3 Relationship between SFMDV[4:0] bits, cycle multiplier, and serial clock frequencies

SFMDV[4:0]	Cycle multiplier	PCLKA frequency (MHz)
		48
11111	48	1.00
11110	46	1.04
11101	44	1.09
11100	42	1.14
11011	40	1.20
11010	38	1.26
11001	36	1.33
11000	34	1.41
10111	32	1.50
10110	30	1.60
10101	28	1.71
10100	26	1.85
10011	24	2.00
10010	22	2.18
10001	20	2.40
10000	18	2.67
01111	17	2.82
01110	16	3.00
01101	15	3.20
01100	14	3.43
01011	13	3.69
01010	12	4.00
01001	11	4.36
01000	10	4.80
00111	9	5.33
00110	8	6.00
00101	7	6.86
00100	6	8.00
00011	5	9.60
00010	4	12.00
00001	3	16.00
00000	2	24.00

34.5.2 QSPCLK Signal Duty Ratio

When the reference clock is configured as PCLKA multiplied by an even number, the high- and low-level widths of the QSPCLK signal match each other. When PCLKA is multiplied by an odd number, the high-level width of the QSPCLK signal is longer than the low-level width by 1 PCLKA.

To make the duty ratio of the QSPCLK signal close to 50% when PCLKA multiplied by an odd number is the reference clock, set the SFMDTY bit in the SFMSKC register to 1. With this setting, the rising edge of the QSPCLK output signal is delayed by a half of one PCLKA cycle to perform an interface operation equivalent to a duty ratio of 50%.

When the reference clock is PCLKA multiplied by an even number, the SFMDTY setting in the SFMSKC register is ignored.

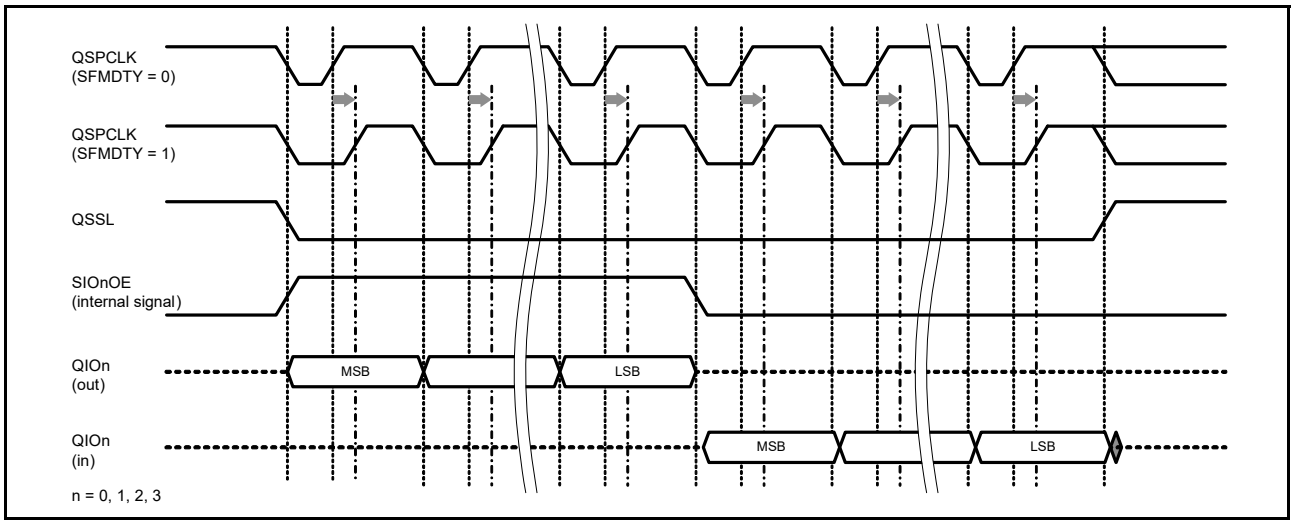


Figure 34.9 Example correction of the QSPCLK signal duty ratio using the SFMDTY bit when PCLKA is multiplied by 3

34.5.3 Minimum High-Level Width of QSSL Signal

Between adjacent SPI bus cycles, the QSSL signal must be held high (inactive) for a sufficient time to satisfy the deselect time required by the serial flash. The reference cycle multiplied by a number from 1 to 16 can be selected as the minimum high-level width of the QSSL output signal in the SFMSW[3:0] bits of the SFMSSC register.

34.5.4 QSSL Signal Setup Time

When the QSPCLK signal first rises after the QSSL signal is driven low, the QSSL signal setup time can be configured to satisfy the serial flash requirements. The setup time can be selected as 0.5 QSPCLK or 1.5 QSPCLK in the SFMSLD bit of the SFMSSC register.

The SFMSLD setting in the SFMSSC register is also applied to the allocation of the setup time from the output of the serial data output enable signal (QIO0OE/QIO1OE/QIO2OE/QIO3OE) until the first rising edge of the QSPCLK signal. Set a value that meets the most constrained timing condition for your application.

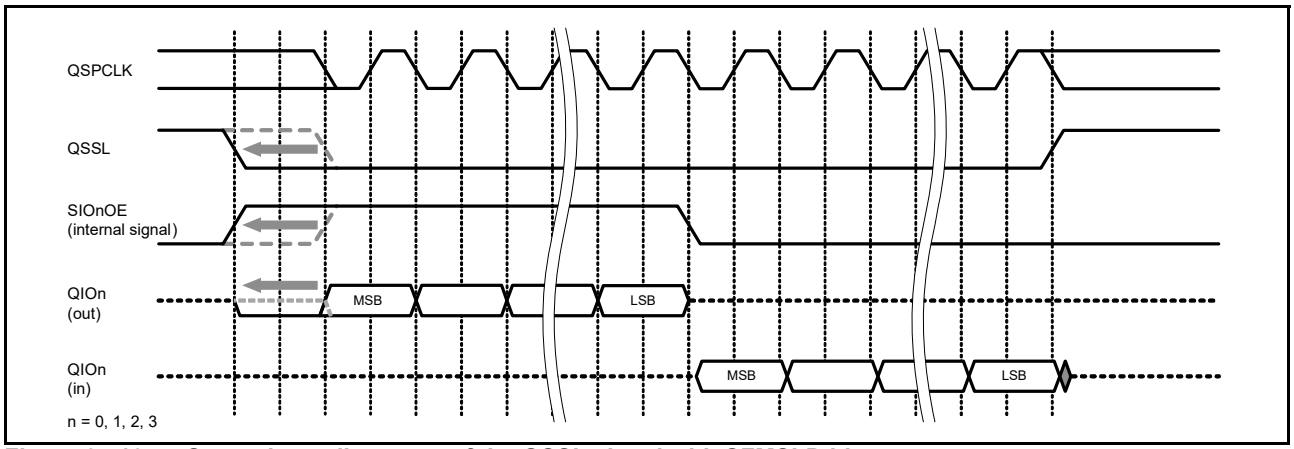


Figure 34.10 Setup time adjustment of the QSSL signal with SFMSLD bit

34.5.5 QSSL Signal Hold Time

When the QSSL signal is driven high after the last rising edge of the QSPCLK signal, the QSSL signal hold time can be configured to satisfy the device requirements. The hold time can be selected as 0.5 QSPCLK or 1.5 QSPCLK in the SFMSHD bit of the SFMSSC register.

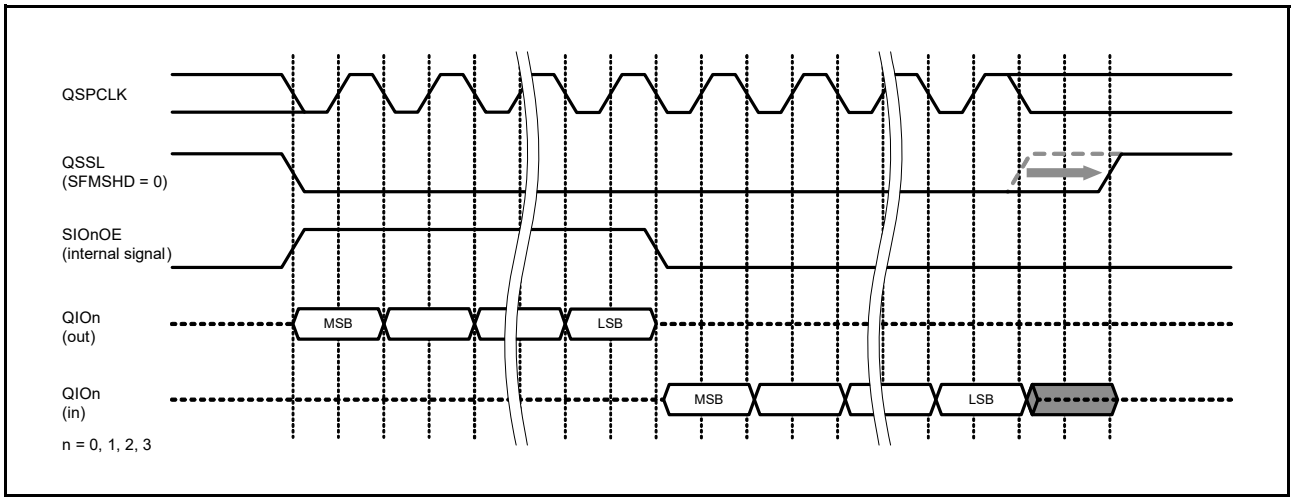


Figure 34.11 Hold time adjustment of the QSSL signal with SFMSHD bit

34.5.6 Hold Time of the Serial Data Output Enable

The buffer output enable of the QIO0, QIO1, QIO2, or QIO3 pin can be extended by 1 QSPCLK with the SFMOEX bit of the SFMSMD register. The extension target signals include only the output enable signals, namely, the QIO0E, QIO1OE, QIO2OE, and QIO3OE signals. The extension target signals do not include the output data signals QIO0O, QIO1O, QIO2O, and QIO3O.

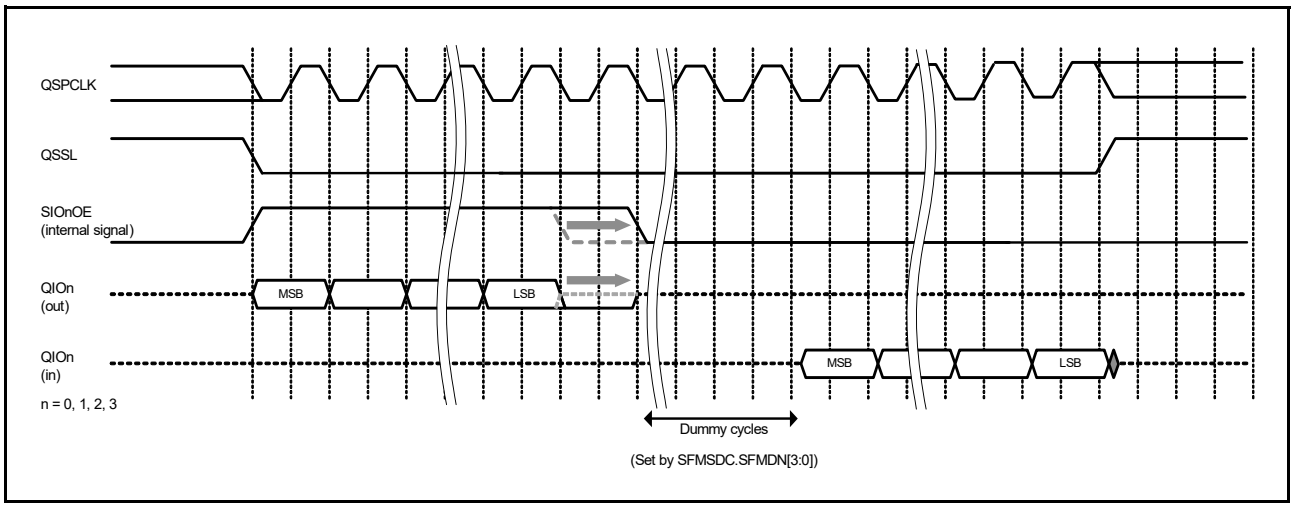


Figure 34.12 Hold time adjustment of output enable with SFMOEX bit

34.5.7 Setup Time of Serial Data Output

When a command or address is transmitted to the serial flash, the setup time begins on serial data output and ends when the QSPCLK signal rises. If this setup time is insufficient, it can be extended by 1 PCLKA with the SFMOSW bit of the SFMSMD register. If the SFMOSW bit is set to 1, the low-level width of QSPCLK during serial data transmission is extended by 1 PCLKA while data is being output from the QSPI. This function has no effect on serial data reception.

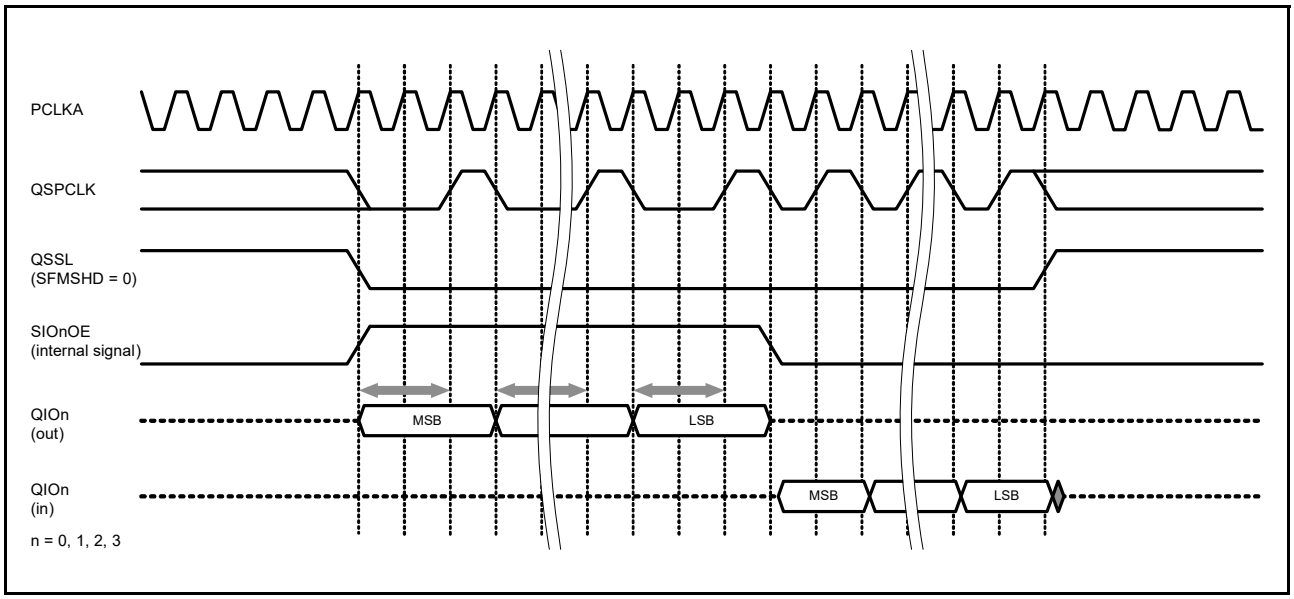


Figure 34.13 Setup time adjustment of serial data output with SFMOSW bit

34.5.8 Hold Time of Serial Data Output

When a command or address is transmitted to the serial flash, the hold time begins on the rising edge of QSPCLK and ends when the serial data makes another transmission.

If the hold time is insufficient, it can be extended by 1 PCLKA with the SFMOHW bit of the SFMSMD register. If the SFMOHW bit is set to 1, the high-level width of QSPCLK during serial data transmission is extended by 1 PCLKA while data is being output from the QSPI. This function has no effect on serial data reception.

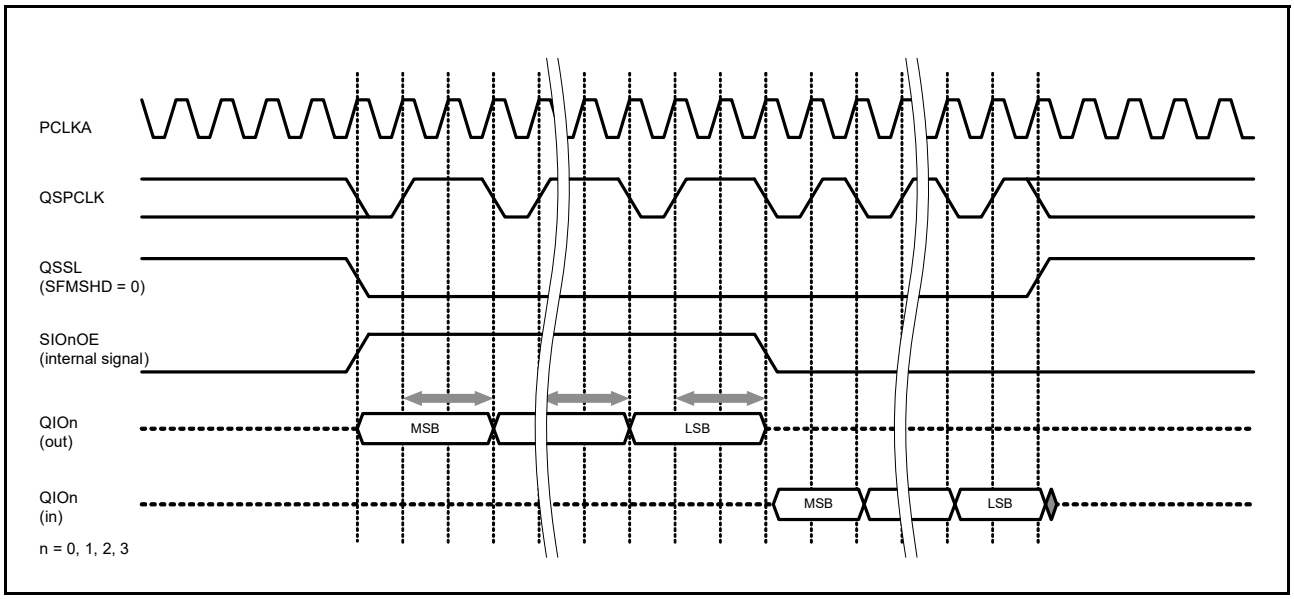


Figure 34.14 Setup time adjustment of serial data output with SFMOHW bit

34.5.9 Serial Data Receiving Latency

The serial flash outputs data in synchronization with the falling edge of the QSPCLK signal. The QSPI receives that data in synchronization with the falling edge of the subsequent QSPCLK signal. The delay from when the serial flash starts outputting data until the QSPI receives that data is called receiving latency. The QSPI adds a latency adjustment cycle immediately before the first data reception cycle in the SPI bus cycle. From the serial flash side, this is seen as an increase in the number of data reception cycles. This added latency adjustment cycle is not generated in the SPI bus cycle without accompanying data reception.

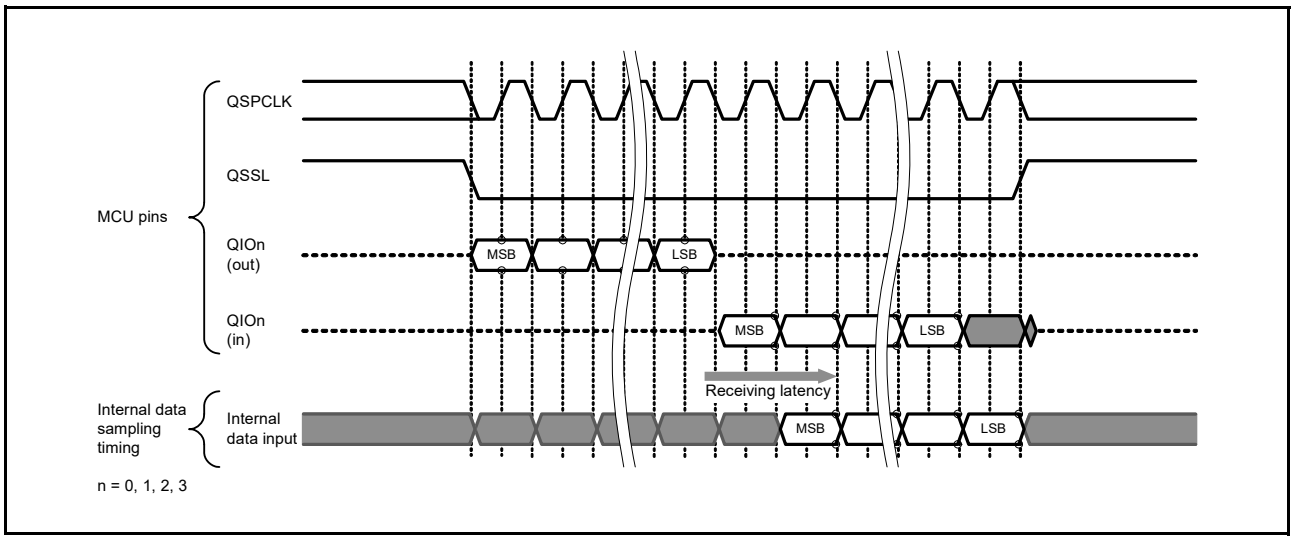


Figure 34.15 Receiving latency

34.6 SPI Instruction Set Used for Flash Access

34.6.1 Types of SPI Instructions Automatically Generated

When the serial flash is accessed, an SPI bus cycle using one of the instructions described in Table 34.4 to Table 34.8 is automatically generated based on the settings in the SFMAS[1:0] bits of the SFMSAC register and in the SFMSMD register.

Table 34.4 SPI instruction set automatically generated when SFMAS[1:0] = 00

Instruction format	Instruction code	Address bytes	Dummy cycles	Data bytes	Remark
Read	03h*1	1	-	1 to ∞	Required (SFMRM[2:0] = 000), (A8 = 0)
	0Bh*1	1	-	1 to ∞	Required(SFMRM[2:0] = 000), (A8 = 1)

Note 1. If the SFMSMD.SFMCCE bit is set to 1, the SFMSIC.SFMCIC[7:0] setting is used as an instruction code.
Note 2. Dummy cycle number can be set with the SFMDRC register.

Table 34.5 SPI instruction set automatically generated when SFMAS[1:0] = 01

Instruction format	Instruction code	Address bytes	Dummy cycles	Data bytes	Remark
Read	03h*1	2	-	1 to ∞	Required (SFMRM[2:0] = 000)

Note 1. If the SFMSMD.SFMCCE bit is set to 1, the SFMSIC.SFMCIC[7:0] setting is used as an instruction code.
Note 2. The number of dummy cycles can be set with the SFMDRC register.

Table 34.6 SPI instruction set automatically generated when SFMAS[1:0] = 10

Instruction format	Instruction code	Address bytes	Dummy cycles	Data bytes	Remark
Read	03h*1	3	-	1 to ∞	Required (SFMRM[2:0] = 000)
Fast Read	0Bh*1	3	8*2	1 to ∞	Selectable (SFMRM[2:0] = 001)
Fast Read Dual Output	3Bh*1	3	8*2	1 to ∞	Selectable (SFMRM[2:0] = 010)
Fast Read Dual I/O	BBh*1	3	4*2	1 to ∞	Selectable (SFMRM[2:0] = 011)
Fast Read Quad Output	6Bh*1	3	8*2	1 to ∞	Selectable (SFMRM[2:0] = 100)
Fast Read Quad I/O	EBh*1	3	6*2	1 to ∞	Selectable (SFMRM[2:0] = 101)
Write Enable	06h	-	-	-	Selectable (ENEX4B[1:0] = 10)
Exit 4-byte mode	E9h	-	-	-	Selectable (ENEX4B[1:0] = 01,10)

Note 1. If the SFMSMD.SFMCCE bit is set to 1, the SFMSIC.SFMCIC[7:0] setting is used as an instruction code.
Note 2. The number of dummy cycles can be set with the SFMDRC register.

Table 34.7 SPI instruction set automatically generated when SFMAS[1:0] = 11 and SFM4BC = 0

Instruction format	Instruction code	Address bytes	Dummy cycles	Data bytes	Remark
Read	03h*1	4	-	1 to ∞	Required (SFMRM[2:0] = 000)
Fast Read	0Bh*1	4	8*2	1 to ∞	Selectable (SFMRM[2:0] = 001)
Fast Read Dual Output	3Bh*1	4	8*2	1 to ∞	Selectable (SFMRM[2:0] = 010)
Fast Read Dual I/O	BBh*1	4	4*2	1 to ∞	Selectable (SFMRM[2:0] = 011)
Fast Read Quad Output	6Bh*1	4	8*2	1 to ∞	Selectable (SFMRM[2:0] = 100)
Fast Read Quad I/O	EBh*1	4	6*2	1 to ∞	Selectable (SFMRM[2:0] = 101)
Write Enable	06h	-	-	-	Selectable (ENEX4B[1:0] = 10)
Enter 4-byte mode	B7h	-	-	-	Selectable (ENEX4B[1:0] = 01,10)

Note 1. If the SFMSMD.SFMCCE bit is set to 1, the SFMSIC.SFMCIC[7:0] setting is used as an instruction code.
Note 2. The number of dummy cycles can be set with the SFMDRC register.

Table 34.8 SPI instruction set automatically generated when SFMAS[1:0] = 11 and SFM4BC = 1

Instruction format	Instruction code	Address bytes	Dummy cycles	Data bytes	Remark
Read	13h*1	4	-	1 to ∞	Required (SFMRM[2:0] = 000)
Fast Read	0Ch*1	4	8*2	1 to ∞	Selectable (SFMRM[2:0] = 001)
Fast Read Dual Output	3Ch*1	4	8*2	1 to ∞	Selectable (SFMRM[2:0] = 010)
Fast Read Dual I/O	BCh*1	4	4*2	1 to ∞	Selectable (SFMRM[2:0] = 011)
Fast Read Quad Output	6Ch*1	4	8*2	1 to ∞	Selectable (SFMRM[2:0] = 100)
Fast Read Quad I/O	ECh*1	4	6*2	1 to ∞	Selectable (SFMRM[2:0] = 101)
Write Enable	06h	-	-	-	Selectable (ENEX4B[1:0] = 10)
Enter 4-byte mode	B7h	-	-	-	Selectable (ENEX4B[1:0] = 01,10)

Note 1. If the SFMSMD.SFMCCE bit is set to 1, the SFMSIC.SFMCIC[7:0] setting is used as an instruction code.
Note 2. The number of dummy cycles can be set with the SFMDRC register.

34.6.2 Standard Read Instruction

The standard Read instruction is a common read instruction supported by most serial flash. When an SPI bus cycle starts, the serial flash selection signal is asserted, and the instruction code (03h/13h)*1 is output. Next, an address of the width (1 to 4 bytes) specified by the SFMAS[1:0] bits of the SFMSAC register is transmitted. Data is then received. This standard Read instruction is selected in the initial QSPI setting.

Note 1. Many 4 Kb serial flash devices have an address field not larger than 1 byte (A7 to A0) to minimize the overhead and to receive A8 information from bit 3 of the Read instruction code. To support these devices, the QSPI only outputs A8 (address bit 8) to bit 3 of the standard Read instruction code only when an address width of 1 byte is specified (SFMAS[1:0] = 00). This means that 0Bh might be output instead of 03h as the standard Read instruction code. This code duplicates the Fast Read instruction code. However, for most of the 2 Kb or smaller serial flash devices, with an address width of 1 byte, bit 3 of a command is designed to be excluded from decoding as a don't care bit, so such a Read instruction code is recognized correctly as the standard Read instruction code. In rare cases, some serial flash devices allow bit 3 to be decoded. When such a serial flash is connected, configure your application to avoid access resulting in A8 = 1.

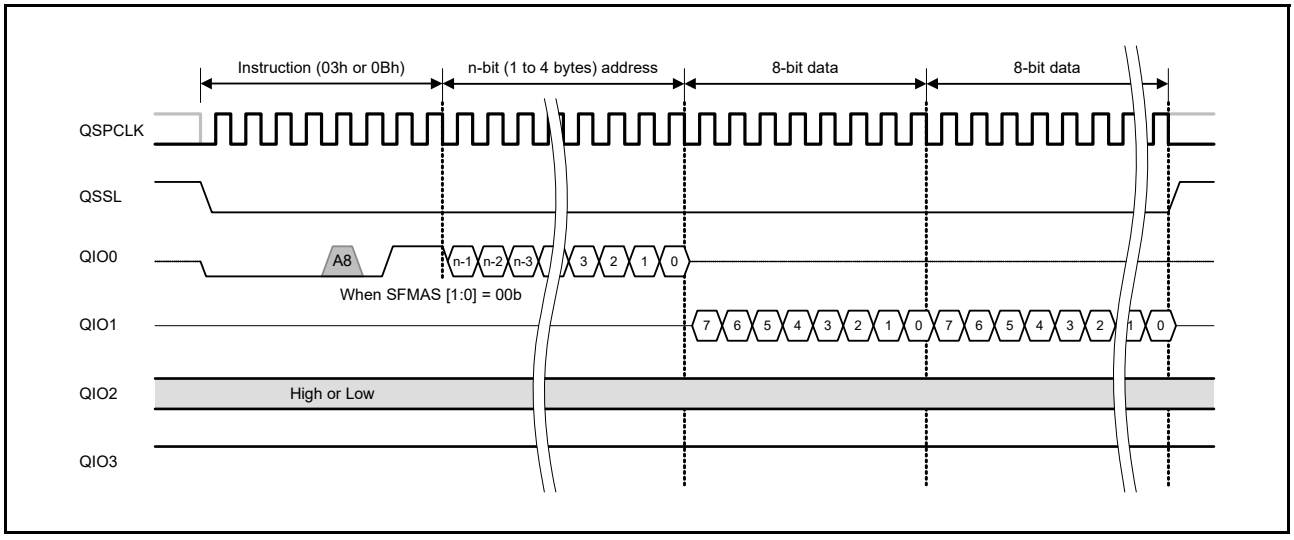


Figure 34.16 Standard read bus cycle

34.6.3 Fast Read Instruction

The Fast Read instruction is a read instruction that supports a higher communication clock speed than the standard Read instruction. When an SPI bus cycle starts, the serial flash selection signal is asserted, and the instruction code (0Bh/0Ch) is output. Next, an address with a width of 1 to 4 bytes, specified by SFMAS[1:0] bits of SFMSAC register, and a certain number of dummy cycles, specified in the SFMSDC register, are transmitted. Data is then received.

The first two dummy cycles are used to select or deselect the XIP mode. When the XIP mode is selected, the same instruction used this time is applied to the next SPI bus cycle, and instruction code transmission of the next SPI bus cycle is skipped. For details of the XIP mode, see section 34.8, XIP Control.

Switching to the Fast Read instruction is controlled in the SFMSMD register.

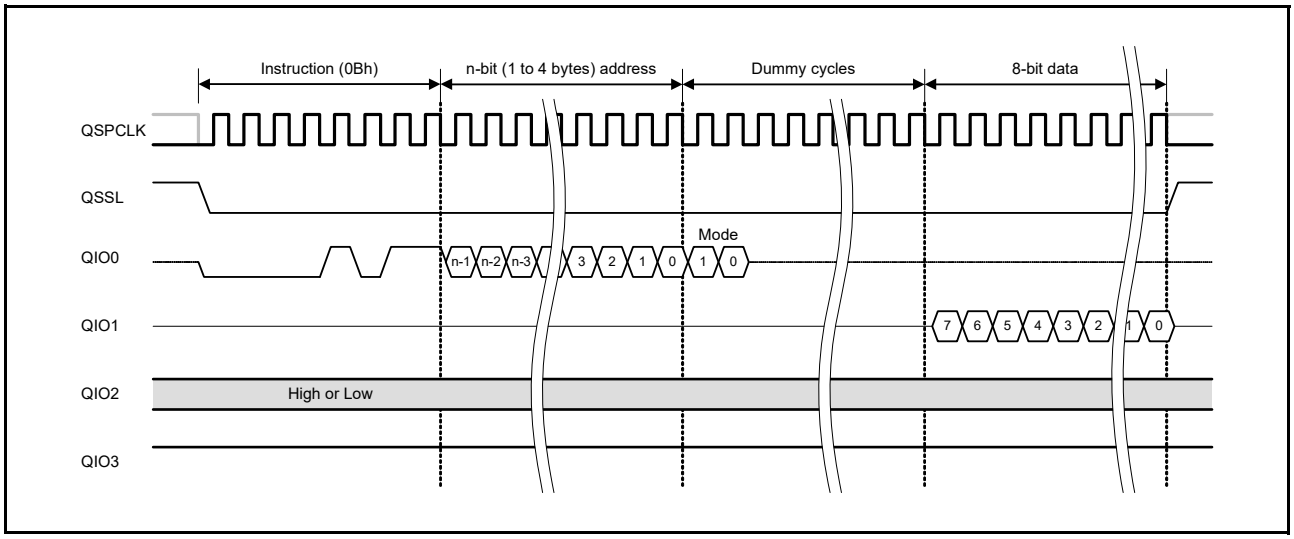


Figure 34.17 Fast Read bus cycle

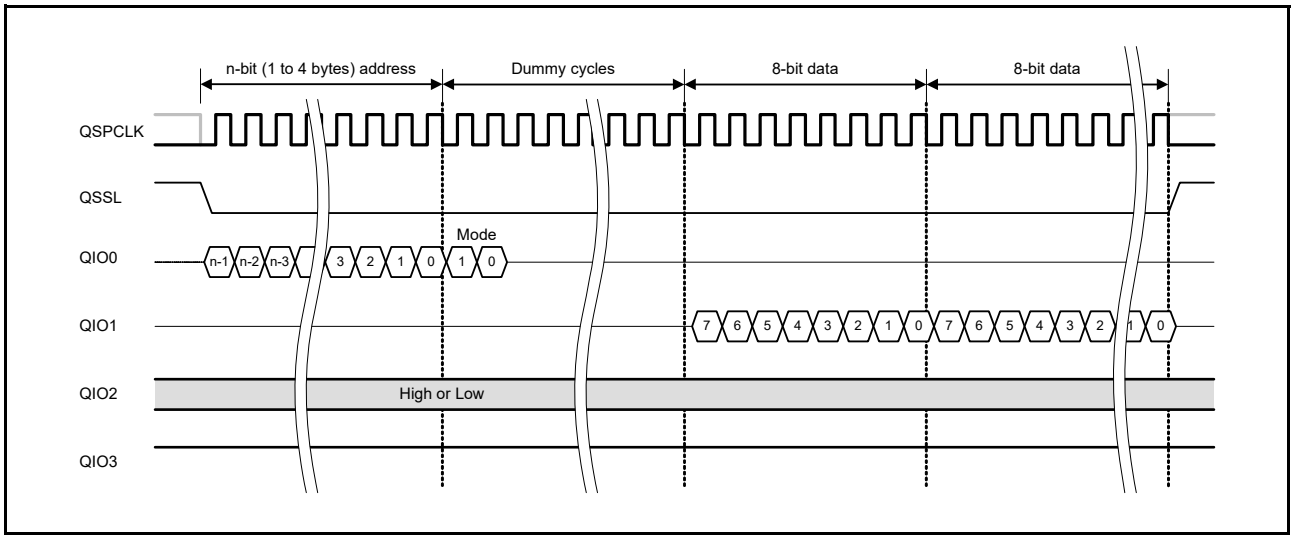


Figure 34.18 Fast Read bus cycle in XIP mode

Note: To use the Fast Read instruction, you must use a serial flash device that supports Fast Read transfers.

34.6.4 Fast Read Dual Output Instruction

The Fast Read Dual Output instruction is a read instruction that uses two signal lines to receive data. When the SPI bus cycle starts, the serial flash select signal is asserted. The instruction code (3Bh/3Ch) and an address with a width of 1 to 4 bytes, specified in the SFMAS[1:0] bits of the SFMSAC register, are transmitted from the QIO0 pin. Next, a certain number of dummy cycles, specified in the SFMSDC register, is generated. Data is then received through the QIO0 and QIO1 pins. Even bit data is received from the QIO0 pin and odd bit data is received from the QIO1 pin.

The first two dummy cycles are used to select XIP mode. When the XIP mode is selected, the same instruction used this time is applied to the next SPI bus cycle, and instruction code transmission of the next SPI bus cycle is skipped. For details on the XIP mode, see section 34.8, XIP Control.

Switching to Fast Read Dual Output is controlled in the SFMSMD register.

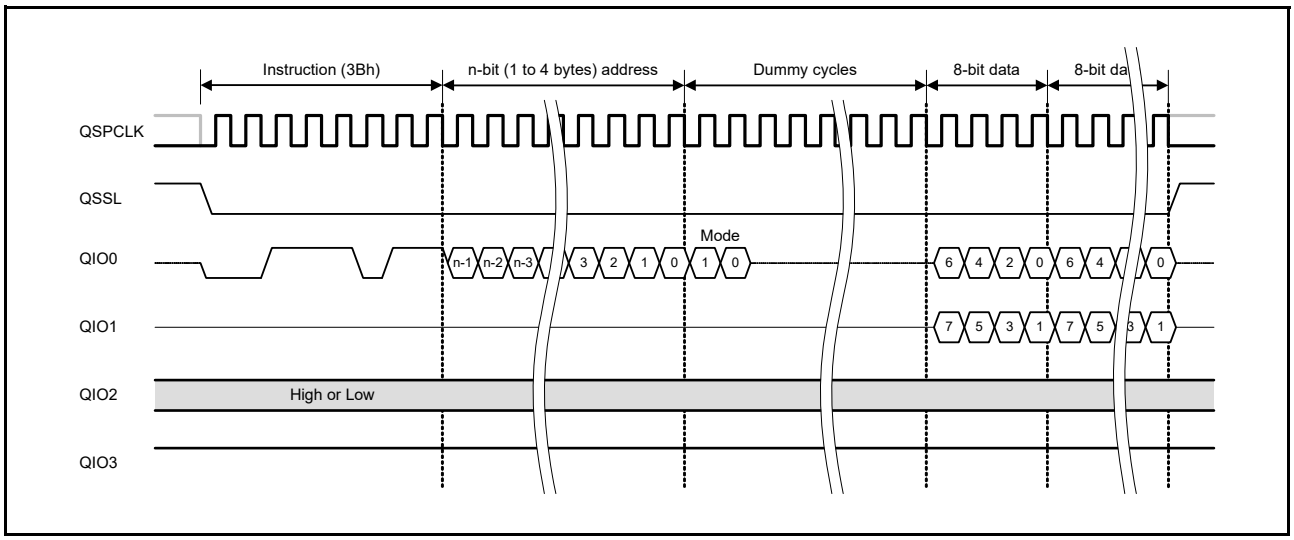


Figure 34.19 Fast read dual output bus cycle

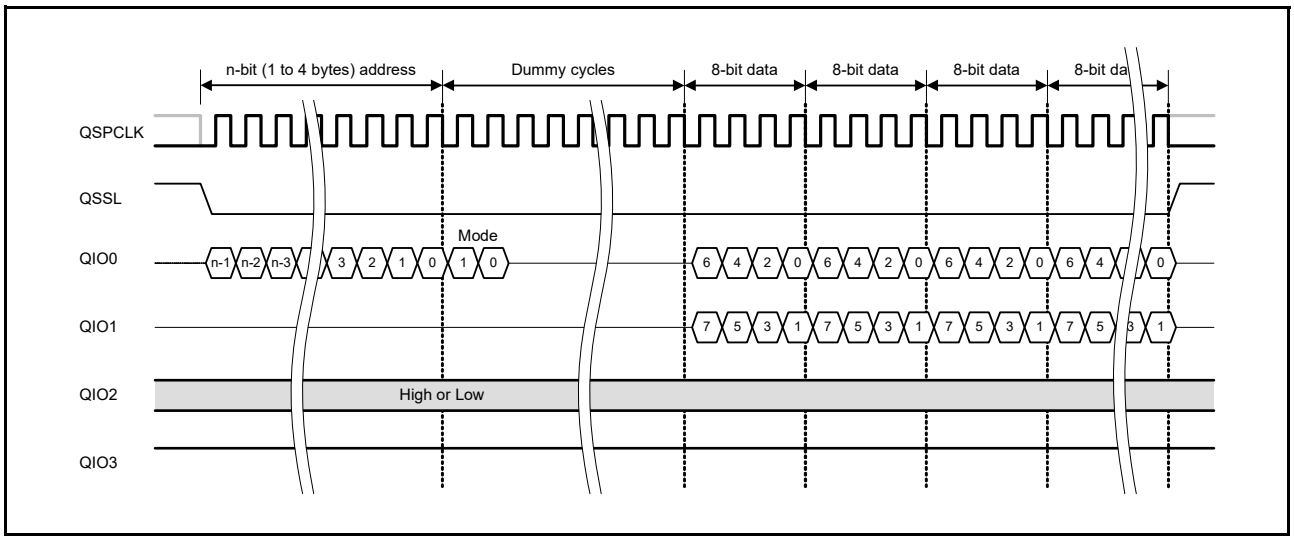


Figure 34.20 Fast read dual output bus cycle in XIP mode

Note: To use the Fast Read Dual Output instruction, you must use a serial flash that supports Fast Read Dual Output transfer.

34.6.5 Fast Read Dual I/O Instruction

The Fast Read Dual I/O instruction is a read instruction that uses two signal lines to transmit an address and receive data. When the SPI bus cycle starts, the serial flash select signal is asserted, and the instruction code (BBh/BCh) is output from the QIO0 pin. Next, an address with a width of 1 to 4 bytes, specified by the SFMAS[1:0] bits of the SFMSAC register is transmitted through the QIO0 and QIO1 pins, and a certain number of dummy cycles, specified in the SFMSDC register is generated. Data is then received through the QIO0 and QIO1 pins. Address and dummy cycle transmission and data reception are performed through the QIO0 pin for even bits and through the QIO1 pin for odd bits.

The first two dummy cycles are used to select XIP mode. When the XIP mode is selected, the same instruction used this time is applied to the next SPI bus cycle, and instruction code transmission of the next SPI bus cycle is skipped. For details of XIP mode, see section 34.8, XIP Control.

Switching to Fast Read Dual I/O is controlled in the SFMSMD register.

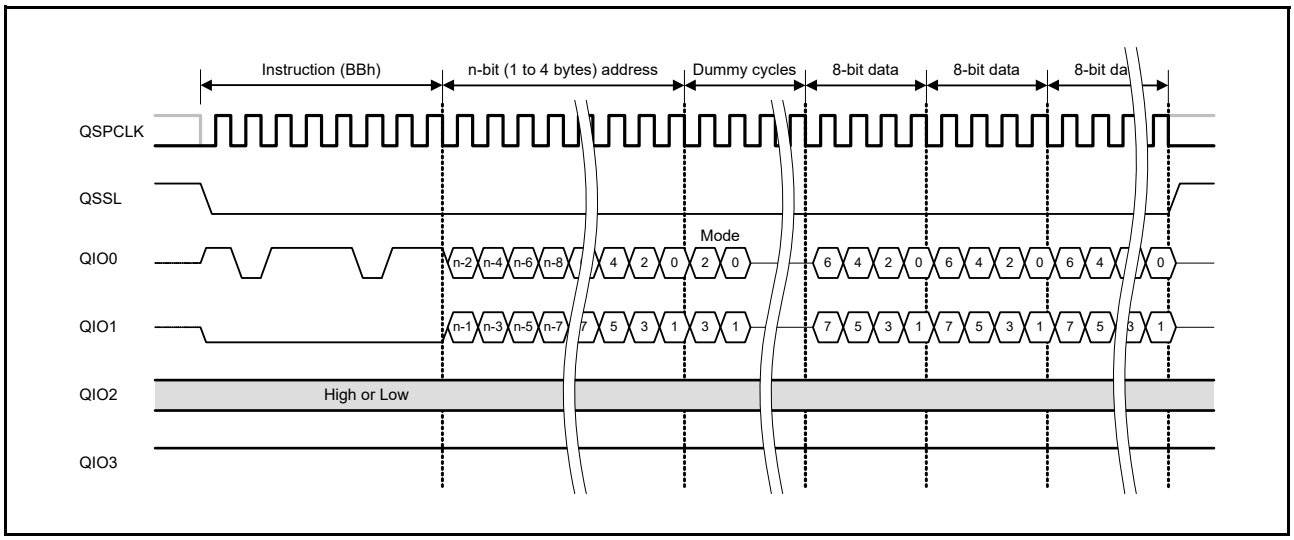


Figure 34.21 Fast read dual I/O bus cycle

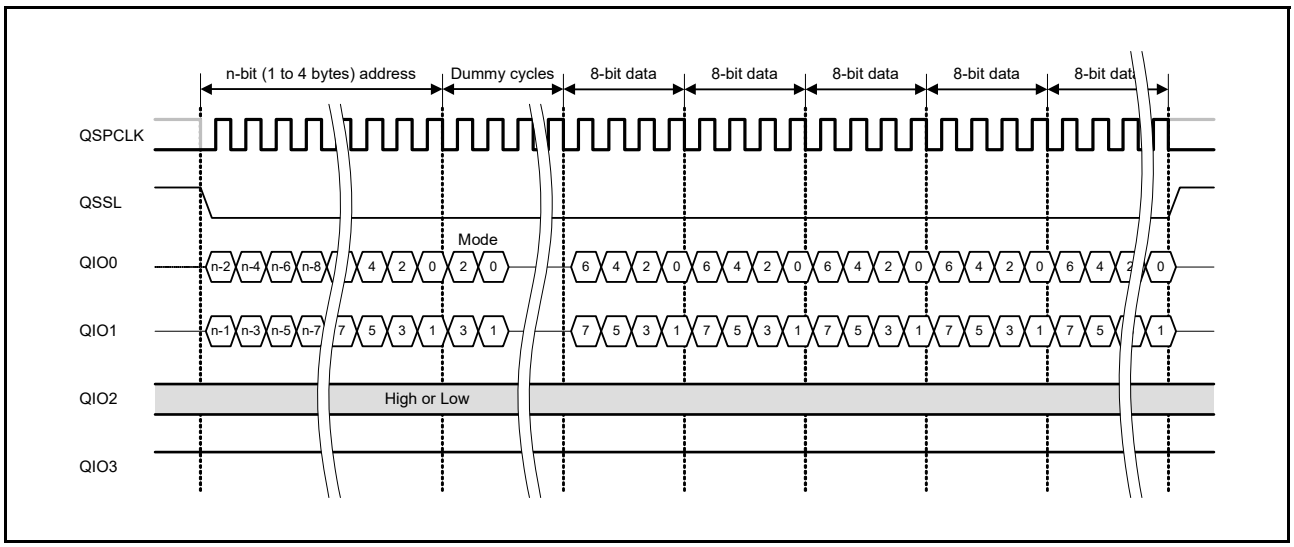


Figure 34.22 Fast read dual I/O bus cycle in XIP mode

Note: To use the Fast Read Dual I/O instruction, you must use a serial flash that supports Fast Read Dual I/O transfer.

34.6.6 Fast Read Quad Output Instruction

The Fast Read Quad Output instruction is a read instruction that uses four signal lines to receive data. When the SPI bus cycle starts, the serial flash select signal is asserted. The instruction code (6Bh/6Ch) and an address with a width of 1 to 4 bytes, specified in the SFMAS[1:0] bits of the SFMSAC register, are output from the QIO0 pin. Next, a certain number of dummy cycles, specified in the SFMDN[3:0] bits of the SFMSMD register is generated. Data is then received through the QIO0, QIO1, QIO2, and QIO3 pins.

The first two dummy cycles are used to select XIP mode. When the XIP mode is selected, the same instruction used this time is applied to the next SPI bus cycle, and instruction code transmission of the next SPI bus cycle is skipped. For details of XIP mode, see section 34.8, XIP Control.

Switching to Fast Read Quad Output is controlled in the SFMSMD register.

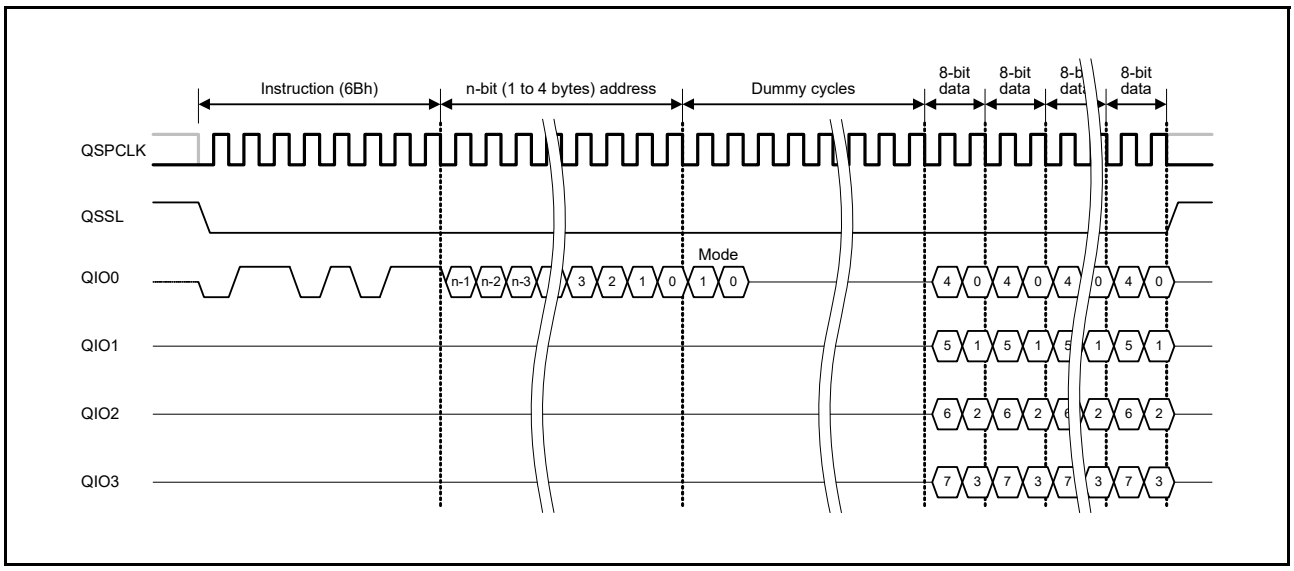


Figure 34.23 Fast read quad output bus cycle

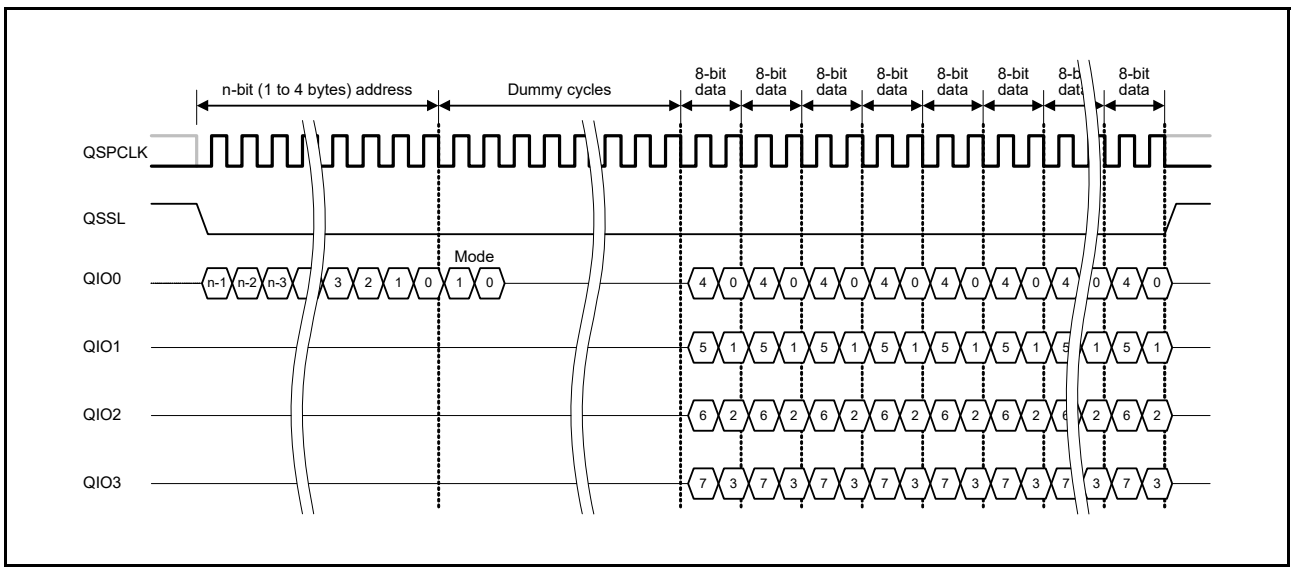


Figure 34.24 Fast read quad output bus cycle in XIP mode

Note: To use fast read quad output, you must use a serial flash that supports fast read quad output transfer.

34.6.7 Fast Read Quad I/O Instruction

The Fast Read Quad I/O instruction is a read instruction that uses four signal lines to transmit an address and receive data. When the SPI bus cycle starts, the serial flash select signal is asserted, and the instruction code (EBh/ECh) is output. Next, an address with a width of 1 to 4 bytes, specified in the SFMAS[1:0] bits of the SFMSAC register is transmitted through the QIO0, QIO1, QIO2, and QIO3 pins, and a certain number of dummy cycles, specified in the SFMDN[3:0] bits of the SFMSMD register is generated. Data is then received through the QIO0, QIO1, QIO2, and QIO3 pins.

The first two dummy cycles are used to select XIP mode. When the XIP mode is selected, the same instruction used this time is applied to the next SPI bus cycle, and instruction code transmission of the next SPI bus cycle is skipped. For details of XIP mode, see section 34.8, XIP Control.

Switching to fast read quad I/O is controlled the SFMSMD register.

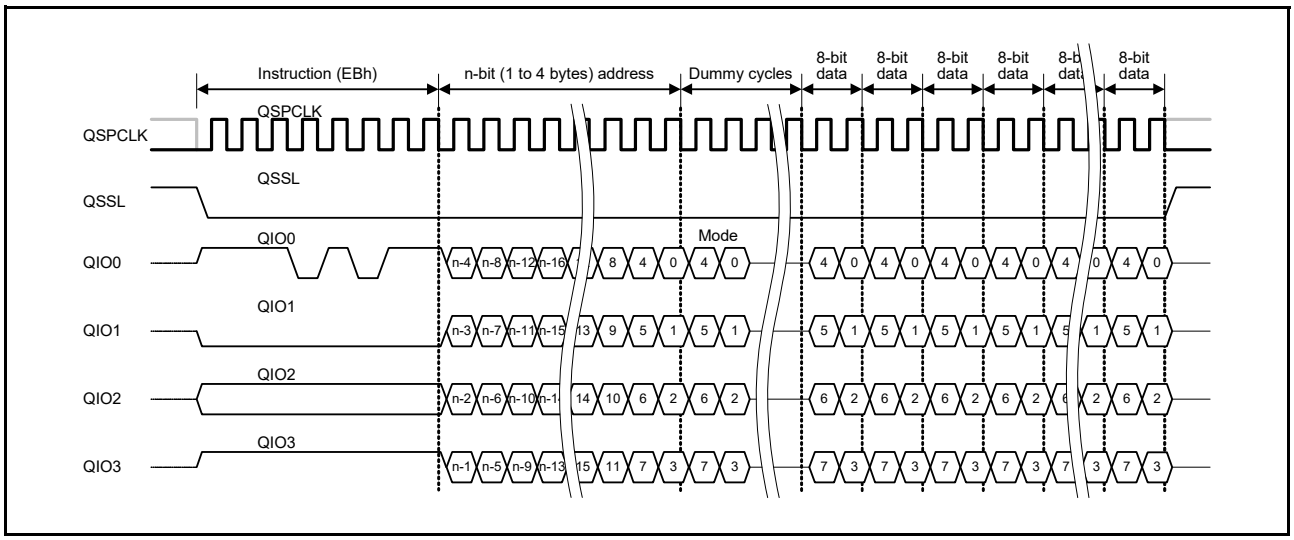


Figure 34.25 Fast read quad I/O bus cycle

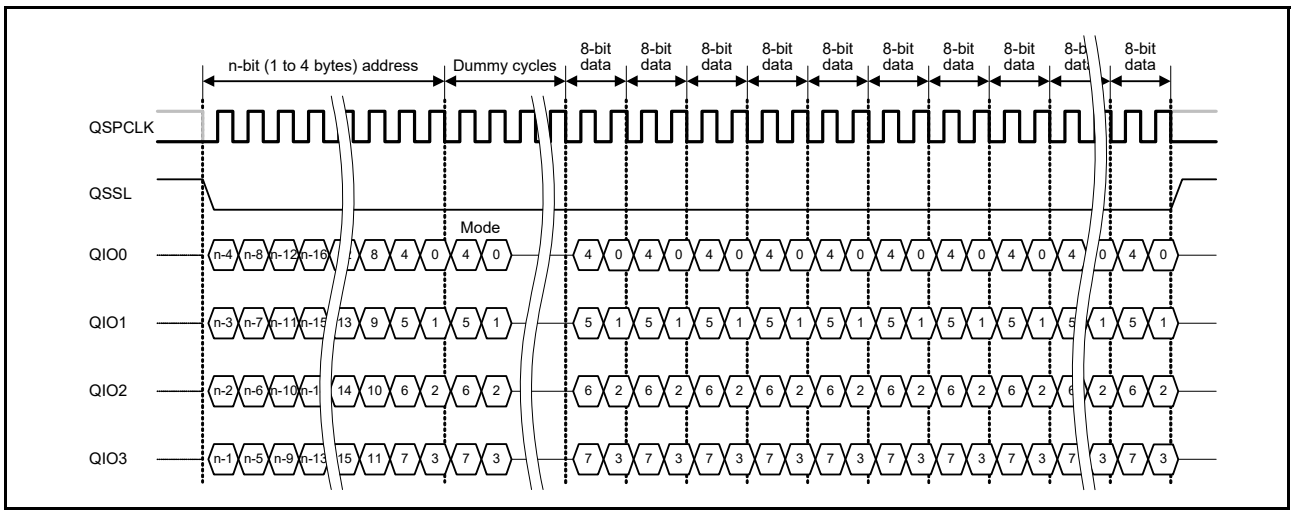


Figure 34.26 Fast read quad I/O bus cycle (in XIP mode)

Note: To use the Fast Read Quad I/O instruction, you must use a serial flash that supports Fast Read Quad I/O transfer.

34.6.8 Enter 4-Byte Mode Instruction

The enter 4-byte mode instruction sets the serial flash address width to four bytes. When the SPI bus cycle starts, the serial flash select signal is asserted, and the instruction code (B7h) is output.

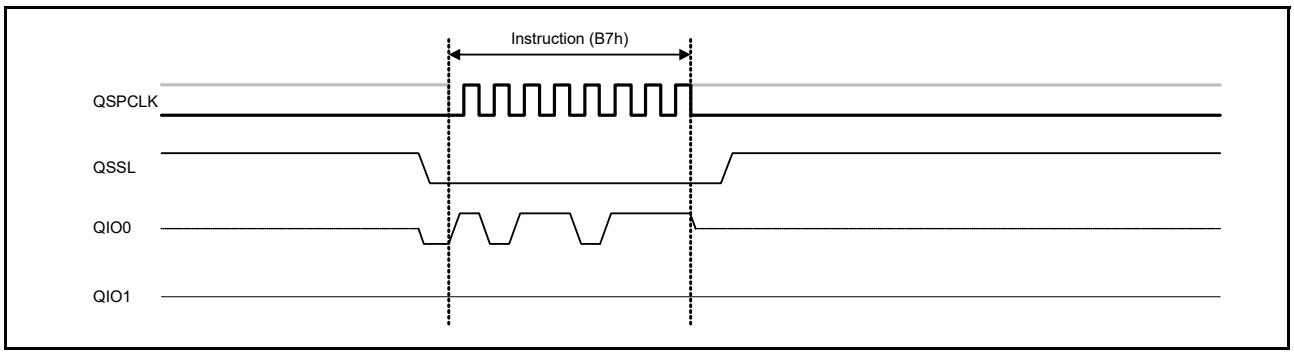


Figure 34.27 Bus cycle for enter 4-byte mode instruction

Note: The Enter 4-Byte Mode instruction is issued regardless of whether the serial flash is in 4- or 3-byte mode.

34.6.9 Exit 4-byte Mode Instruction

The exit 4-byte mode instruction sets the serial flash address width to three bytes. When the SPI bus cycle starts, the serial flash select signal is asserted, the instruction code (E9h) is output.

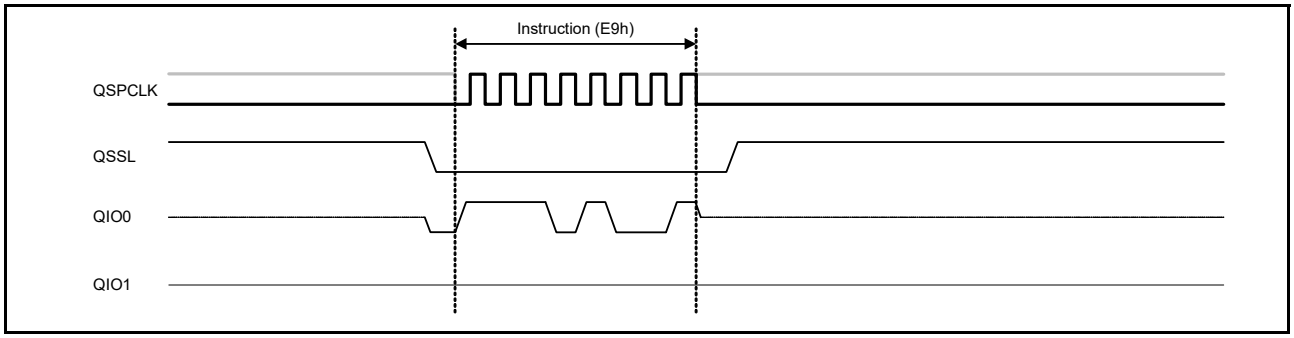


Figure 34.28 Bus cycle for exit 4-byte mode instruction

Note: The exit 4-byte mode instruction is issued regardless of whether the serial flash is in 4- or 3-byte mode.

34.6.10 Write Enable Instruction

The write enable instruction enables changing of the serial flash address width. When the SPI bus cycle starts, the serial flash select signal is asserted, and the instruction code (06h) is output.

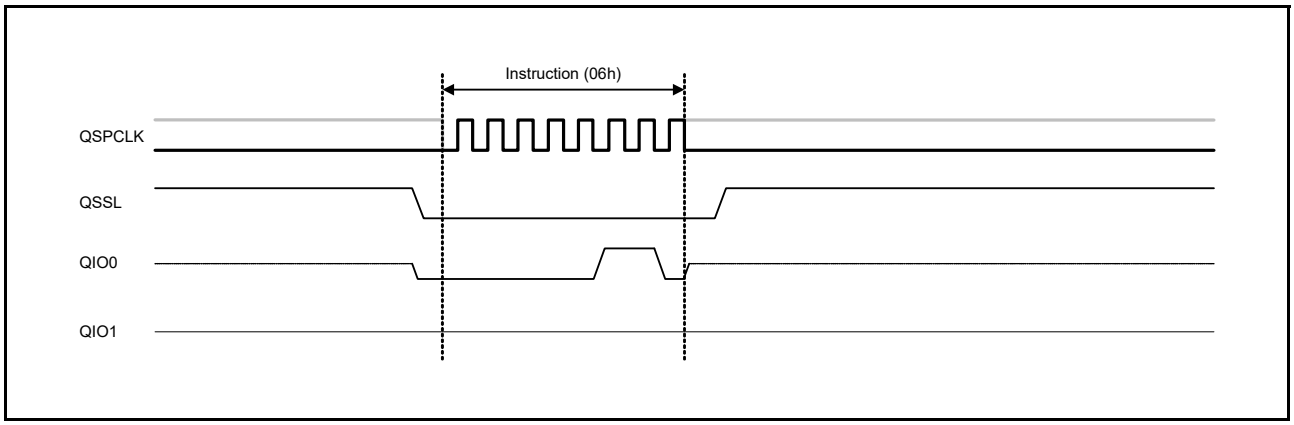


Figure 34.29 Write enable bus cycle

34.7 SPI Bus Cycle Arrangement

34.7.1 Flash Read Based on Individual Conversion

ROM read internal bus cycles are independently converted to SPI bus cycles on a one-to-one basis. When a ROM read bus cycle is detected, the QSSL signal is asserted, and an SPI bus cycle starts. When data is received from the serial flash, the QSSL signal is deasserted and the SPI bus cycle is complete.

When another ROM read bus cycle is detected, the QSSL signal is asserted again after ensuring the minimum high-level width of the QSSL signal is reached. Then another SPI bus cycle starts.

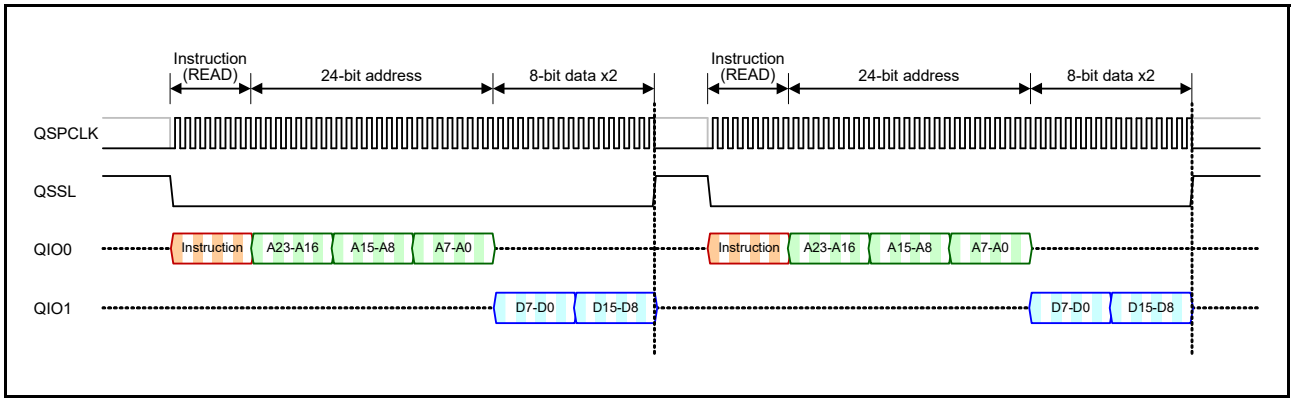


Figure 34.30 Successive data read operations based on individual conversion

34.7.2 Flash Read Using Prefetch Function

In operations such as CPU instruction execution and block data transfer, data is often read in ascending order from contiguous flash addresses. Serial flash provides the ability to repeat data reception without reissuing an instruction code and address. However, if bus cycles issued by the MCU are independently converted, SPI bus cycles are separated from each other, resulting in a failure to take advantage of this feature of a serial flash. The QSPI has a prefetch function for continuous data reception.

To enable the prefetch function, set the SFMPFE bit in the SFMSMD register to 1. When the prefetch function is enabled, data is received continuously and stored in the buffer, without waiting for another flash read request. When the MCU performs a flash read operation, an address check is made. If an address match is confirmed, the data in the buffer is passed to the MCU. If an address mismatch is found, the data in the buffer is discarded and a new SPI bus cycle is issued.

The prefetch buffer is 18 bytes. When the prefetch buffer becomes full, the SPI bus cycle ends. When buffer data is read to create free space, a new SPI bus cycle is automatically started to resume prefetching.

The prefetch function allows for efficient transfer operations when data is read in ascending order from contiguous addresses, as in instruction fetch and block data transfer.

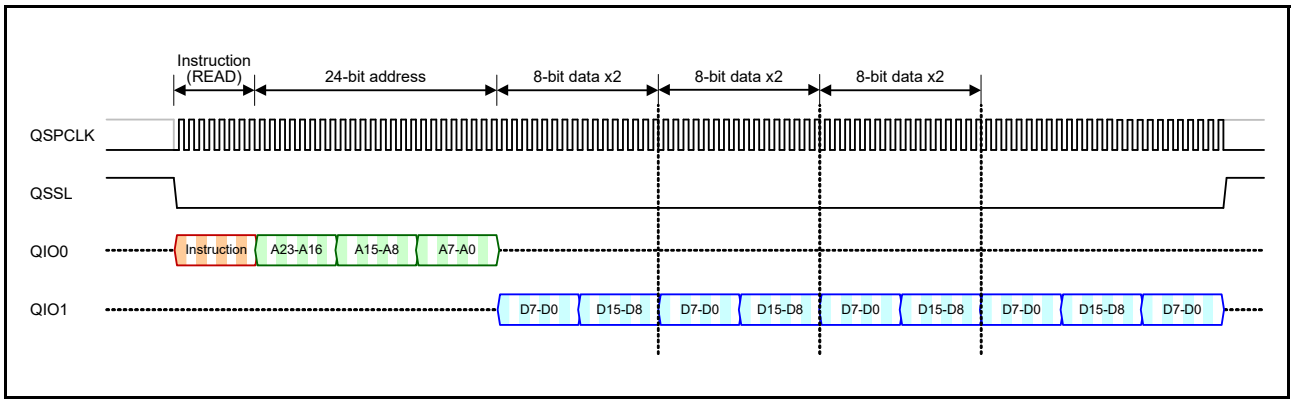


Figure 34.31 Successive data read operations using the prefetch function

34.7.3 Halt of Prefetching

If a ROM read bus cycle for reading from another address occurs during a serial transfer for prefetching, the unnecessary serial transfer being made is halted and a new SPI bus cycle is started. Usually, such a halt of serial transfer occurs on data reception byte boundaries. However, if the SFMPAE bit in the SFMSMD register is set to 1, the halt can occur on locations other than byte boundaries. To use this function, the serial flash device must support halts not on byte boundaries.

34.7.4 Direct Specification of Prefetch Destination

When the SFMPFE bit is set and the QSPI receives internal bus write access to the QSPI window area, the system obtains it as a prefetch address and starts to prefetch. Internal bus write access to the QSPI window area can only be used to obtain prefetch address data. Serial flash write operation cannot be performed.

Combining this function with the prefetch state polling function described in section 34, Prefetch State Polling, can reduce the load on the internal bus when data is read from a low-speed serial flash.

Note: When writing to the QSPI window area to indicate a prefetch destination, write to the first byte of the address where prefetching is to be started. Writes to the QSPI window area with a data size of 2 bytes or more return an ERROR response.

34.7.5 Prefetch State Polling

Reading data from a low-speed serial flash increases system load, because the internal bus is placed in wait status until completion of the SPI reception bus cycle. The prefetch state polling function is provided to reduce this load.

The PFOFF bit in the SFMSST register indicates the state of the prefetch function, and the PFCNT[4:0] bits in the SFMSST register indicate the number of data bytes already prefetched. This allows the prefetch status to be determined with a single CPU operation.

```
//
// copy 1K byte (32bit x 256 word) data from Serial flash to external memory
//
unsigned long *sptr;                    // pointer for the Serial flash
unsigned long *dptr;                    // pointer for the external memory
int i;

SFMSMD |= 0x0040;                       // set SFMPFE bit to enable prefetch
*((volatile unsigned char *) sptr) = 0;  // make the TAG valid to start prefetch

for (i = 0; i < 256; i++){
while ((SFMSST & 0x00FF) < 0x04){};  // waiting for 4 byte data received
*(dptr++) = *(sptr++);
}
```

Note: When executing a polling program, place the program outside of the serial flash or enable the instruction cache. If the polling program is executed when the program is placed on the serial flash or is executed without using the instruction cache, the prefetch target frequently switches to an instruction code. This eliminates the effect of polling, and an infinite loop can result because the prefetch buffer is not filled.

34.7.6 Flash Read Using SPI Bus Cycle Extension Function

If the SFMSE[1:0] bits in the SFMSMD register are set to a value other than 00, the QSPI waits for the next flash read, suspending the SPI bus cycle, while stopping the QSPCLK signal and holding the QSSL signal low even after data is obtained from the serial flash.

If the address of the next flash read is contiguous in ascending order, the toggling of the QSPCLK signal is restarted to continue reception of subsequent data. If the address of the next flash read is not contiguous in ascending order, the QSSL signal is driven high to end the suspended SPI bus cycle. A new SPI bus cycle is then started.

When data is read intermittently from ascending order contiguous addresses, this function enables an efficient transfer operation to be performed by reducing the overhead for instruction code and address transmission.

The SPI bus cycle extension time can be selected in the SFMSE[1:0] bits of the SFMSMD register. When the specified extension time elapses, the QSSL signal returns to high to automatically end the suspended SPI bus cycle.

If the SFMSE[1:0] bits are set to 11, QSSL is extended infinitely. This increases the power consumption of the serial flash.

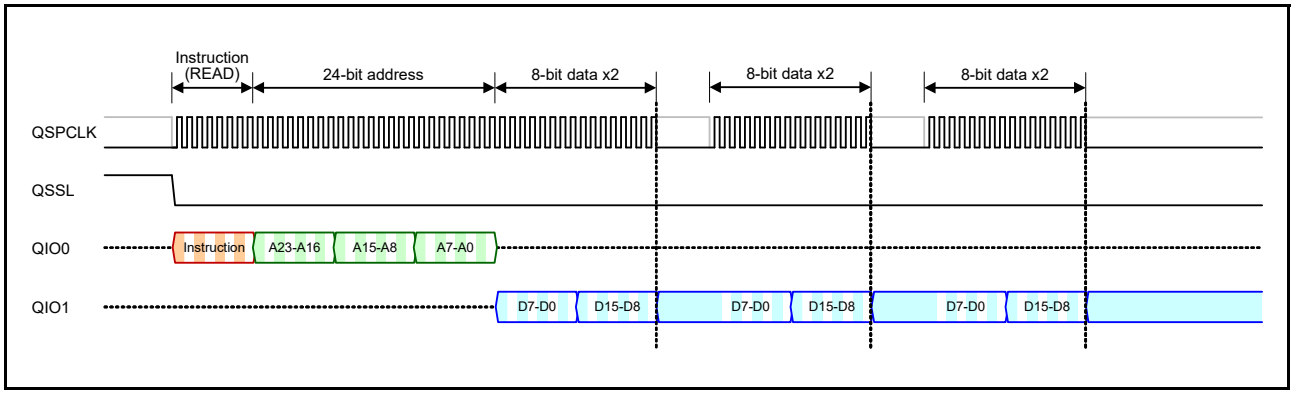


Figure 34.32 Successive data read operations using SPI bus cycle extension

34.8 XIP Control

Some serial flash devices allow latencies to be reduced by skipping instruction code reception for flash reads. This instruction code skip function is selected with mode data received during the dummy cycle period of the previous serial bus cycle.

In the dummy cycle of the Fast Read instructions, the QSPI controls the XIP mode of a serial flash by using the serial data signals to send the mode data set in the SFMXD[7:0] bits of the SFMSDC register during the first 2 cycles, as shown in Figure 34.33.

The mode data to enable XIP mode changes for each serial flash. Take this into account to set the appropriate mode data in the SFMXD[7:0] bits of the SFMSDC register.

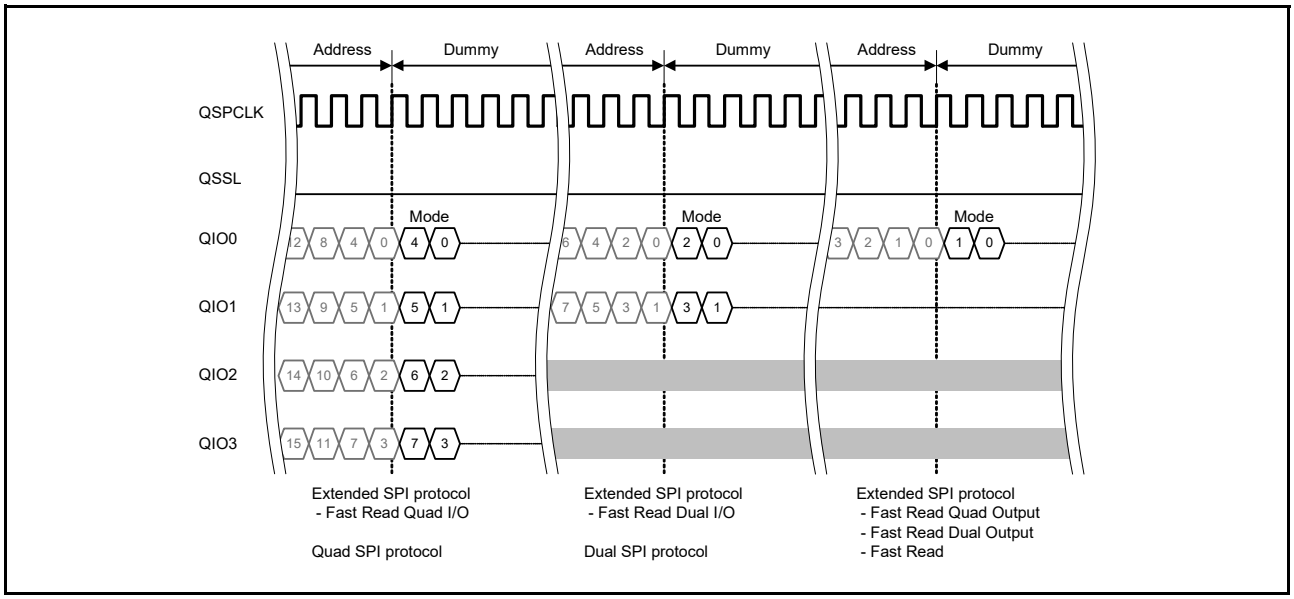


Figure 34.33 XIP mode control data

34.8.1 Setting XIP Mode

Suppose that the XIP mode select configuration specified for a serial flash device is set in the SFMXD[7:0] bits of the SFMSDC register and 1 is set in the SFMXEN. In the dummy cycle of the next Fast Read, the mode data specified in the SFMXD[7:0] bits of the SFMSDC register is transferred to the serial flash device. From that point, the XIP mode is enabled in both the serial flash controller and the serial flash device. To confirm the completion of the actual XIP mode select procedure, read 1 from the SFMXST bit in the SFMSDC register.

Note: Set the SFMXD[7:0] bits in the SFMSDC register to the XIP mode setting data specified for the actual serial flash device. The XIP mode of the serial flash controller is only enabled in the SFMXEN bit, regardless of the SFMXD[7:0] setting in the SFMSDC register.

34.8.2 Releasing XIP Mode

Suppose that the XIP mode release configuration specified for a serial flash is set in the SFMXD[7:0] bits of the SFMSDC register and 0 is set in the SFMXEN bit. In the dummy cycle of the next Fast Read, the mode data specified in the SFMXD[7:0] bits of the SFMSDC register is transferred to the serial flash during first two-cycle period. From that point, the XIP mode is disabled in both the QSPI and the serial flash device. To confirm completion of the actual XIP mode release procedure, read 0 from the SFMXST bit in the SFMSDC register.

Note: Set the SFMXD[7:0] bits in the SFMSDC register to the XIP mode setting data specified for the actual serial flash device. The XIP mode of the serial flash controller is only disabled in the SFMXEN bit, regardless of the SFMXD[7:0] setting in the SFMSDC register.

34.9 QIO2 and QIO3 Pin States

The states of QIO2 pin and QIO3 pin depend on the serial interface read mode specified by the SFMRM[2:0] bits of the SFMSMD register.

Table 34.9 States of QIO2 pin and QIO3 pin

SFMSMD.SFMRM[2:0] bits	QIO2 pin state*1	QIO3 pin state*2	Remark
111	Setting prohibited		
110			
101	Input or output as a serial data signal (standby level is Hi-Z)	Input or output as serial data signal (Standby level is Hi-Z)	Fast Read Quad I/O
100			Fast Read Quad Output
011	Output SFMWPL bit variable of SFMPMD register (initial output variable is low level)	Output high level	Fast Read Dual I/O
010			Fast Read Dual Output
001			Fast Read
000			Read (Initial State)

Note 1. The serial flash can also use the QIO2 pin as the WP function.
Note 2. The serial flash can also use the QIO3 pin as the HOLD or RESET function.

34.10 Direct Communication Mode

34.10.1 About Direct Communication

The QSPI can read the serial flash contents by automatically converting a ROM read bus cycle to an SPI bus cycle. However, serial flash devices have many different functions in addition to memory data read, including ID information read, erase, programming, and status information read. There is no standardized instruction set for using these functions, and more functions are being added rapidly by different vendors to different devices. It is difficult to support these functions by hardware control.

The QSPI flexibly supports those serial flash devices by providing a means for the software to directly communicate with the serial flash, so that the software can create any SPI bus cycle required.

34.10.2 Direct Communication Mode

To communicate directly with a serial flash, transition to the direct communication mode by setting the DCOM bit in the SFMCMD register to 1. While the direct communication mode is selected, ordinary flash read operation is disabled. For standard flash access after direct communication, terminate the direct communication mode by setting the DCOM bit in the SFMCMD register to 0.

Note: If the QSPI is set to the XIP mode, you must terminate the XIP mode before starting direct communication mode.

34.10.3 SPI Bus Cycle Generation in Direct Communication

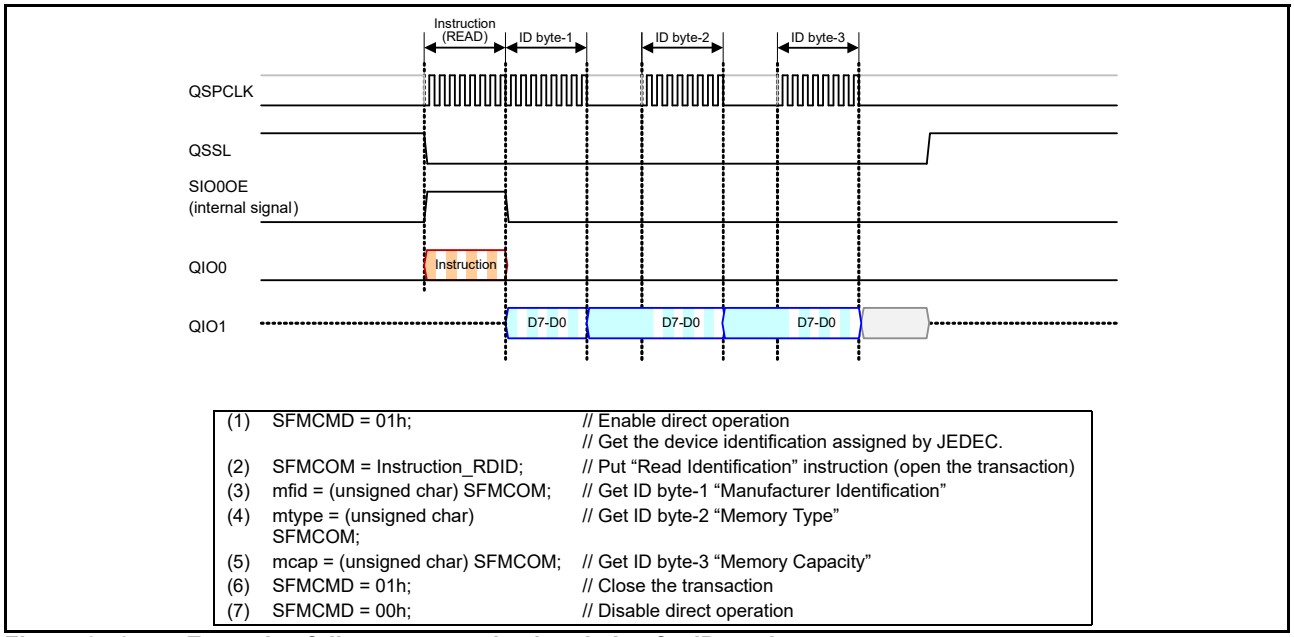
The SPI bus cycle in direct communication starts on the first access to the SFMCOM port and ends by a write to the SFMCMD register, after a series of I/O operations is performed through the SFMCOM port. At that point, a write to the SFMCOM port is converted to a one-byte transmission to the SPI bus, and a read from the SFMCOM port is converted to a one-byte reception from the SPI bus.

During the period from the first access to the SFMCOM port to the last write operation to the SFMCMD register, the serial flash select signal is held active to notify the serial flash that a series of SPI bus cycles is in progress.

Note: In the direct communication mode, all writes to registers other than SFMCMD, including SFMSMD, SFMSSC, SFMSKC, SFMSST, SFMCST, SFMSIC, SFMSAC, SFMSDC, SFMSPC, and SFMPMD, are disabled. With this circuit configuration, writing to a register area other than the SFMCOM port terminates the SPI bus cycle. However, writing to a register area other than SFMCMD as a way to terminate the SPI bus cycle is not guaranteed as a normal function.

The following is an example program for direct communication.

```
//### CAUTION! ### This code must be outside the Serial flash that is going to be operated.

// Define specific instruction codes of the target Serial flash device.
#define Instruction_FREAD 0x0B  // Fast Read
#define Instruction_RDSR 0x05   // Read Status register
#define Instruction_RDID 0x9F   // Read Identification
#define Instruction_WREN 0x06   // Write Enable
#define Instruction_CERA 0xC7   // Chip Erase

unsigned char mfid, mtype, mcap, data, temp;

SFMCMD = 0x01; // Enable direct operation

// Get the device identification assigned by JEDEC.
SFMCOM = Instruction_RDID;        // put “Read Identification” instruction (open SPI bus cycle)
mfid = (unsigned char) SFMCOM;    // get “Manufacturer Identification”
mtype = (unsigned char) SFMCOM;   // get “Memory Type”
mcap = (unsigned char) SFMCOM;    // get “Memory Capacity”
SFMCMD = 0x01h;                   // close SPI bus cycle

// Get one byte from the address 0x012345h.
SFMCOM = Instruction_FREAD;       // put “Fast Read” instruction (open SPI bus cycle)
SFMCOM = 0x01;                    // put upper byte of the address 0x012345
SFMCOM = 0x23;                    // put middle byte of the target address 0x012345
SFMCOM = 0x45;                    // put lower byte of the target address 0x012345
temp = (unsigned char) SFMCOM;    // get one byte dummy code for FAST READ transaction
data = (unsigned char) SFMCOM;    // get the data
SFMCMD = 0x01;                    // close SPI bus cycle

// Erase All contents.
SFMCOM = Instruction_WREN;  // put “Write Enable” instruction (open SPI bus cycle)
SFMCMD = 0x01;              // close SPI bus cycle
SFMCOM = Instruction_CERA;  // put “Chip Erase” instruction (open SPI bus cycle)
SFMCMD = 0x01;              // close SPI bus cycle
SFMCOM = Instruction_RDSR;  // put “Read Status Register” instruction (open SPI bus cycle)
while (SFMCOM & 0x01){};    // Polling “Write Progress Bit” until completion
SFMCMD = 0x01;              // close SPI bus cycle

SFMCMD = 0x00; // Disable direct operation
```




```
(1)  SFMCMD = 01h;                    // Enable direct operation
                                      // Get the device identification assigned by JEDEC.
(2)  SFMCOM = Instruction_RDID;       // Put "Read Identification" instruction (open the transaction)
(3)  mfid = (unsigned char) SFMCOM;   // Get ID byte-1 "Manufacturer Identification"
(4)  mtype = (unsigned char)          // Get ID byte-2 "Memory Type"
     SFMCOM;
(5)  mcap = (unsigned char) SFMCOM;   // Get ID byte-3 "Memory Capacity"
(6)  SFMCMD = 01h;                    // Close the transaction
(7)  SFMCMD = 00h;                    // Disable direct operation
```

Figure 34.34 Example of direct communication timing for ID read

Note: When Extended SPI protocol is used in the direct communication mode, the standard Read or Fast Read instruction must be used to reference the contents of the serial flash. The QSPI does not support Fast Read Dual Output, Fast Read Dual I/O, Fast Read Quad Output, or Fast Read Quad I/O transfers in this configuration. When these high-speed read operation are required, use standard flash access.

34.11 Operation

34.11.1 Procedure for Modifying Settings of Multiple Control Registers

The settings of the QSPI control registers can be modified dynamically during system operation. However, when the settings of multiple control registers are modified sequentially, an SPI bus cycle might occur before all of the registers are updated. The register setting sequence must be carefully designed so that the SPI bus timing specification is satisfied at all stages of register setting modification.

```
//
// Making QSPCLK faster
//
SFMSMD = 0x0041;  // SFMPAE: 0 SFMPFE: 1 SFMSE:00 SFMRM:01 (prefetch enable fast read)
SFMSSC = 0x04;    // SFMSLD: 0 SFMSHD: 0 SFMSW:4 (minimum QSSL high width = 5 sck)
SFMSKC = 0x00;    // SFMDTY: 0 SFMDV: 0 (1/2 mode) ### switch clock speed last ###

//
// Making QSPCLK slower
//
SFMSKC = 0x06;    // SFMDTY: 0 SFMDV:6 (1/8 mode) ### switch clock speed first ###
SFMSSC = 0x01;    // SFMSLD: 0 SFMSHD:0 SFMSW: 1 (minimum QSSL high width = 2 sck)
SFMSMD = 0x0040;  // SFMPAE: 0 SFMPFE:1 SFMSE: 00 SFMRM:00 (prefetch enable, standard read)
```

34.12 Interrupt

When the EROMR bit in the SFMCST register is 1, the QSPI requests an interrupt. The EROMR bit becomes 1 when ROM read access is detected in the direct communication mode. Interrupt requests are retained until the EROMR bit is cleared by writing 0. For details, see section 14, Interrupt Controller Unit (ICU).

34.13 Usage Note

34.13.1 Setting for the Module-Stop State

QSPI operation can be disabled or enabled using the Module Stop Control Register B (MSTPCRB). The QSPI is initially stopped after reset. Register access is enabled by releasing the module-stop state. For details, see section 11, Low Power Modes.

35. Cyclic Redundancy Check (CRC) Calculator

The Cyclic Redundancy Check (CRC) calculator 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 monitoring reads from and writes 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.

35.1 Overview

Table 35.1 lists the specifications of the CRC calculator and Figure 35.1 shows the block diagram.

Table 35.1 CRC calculator specifications

Parameter	Description	
Data size	8-bit	32-bit
Data for CRC calculation*1	CRC code generated for any desired data in 8n-bit units (where n is a whole number)	CRC code generated for any desired data in 32n-bit units (where n is a whole 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	
CRC snoop	Monitor reads from and writes to a certain register address	-

Note 1. The circuit cannot divide data used in CRC calculations. Write data in 8-bit or 32-bit units.

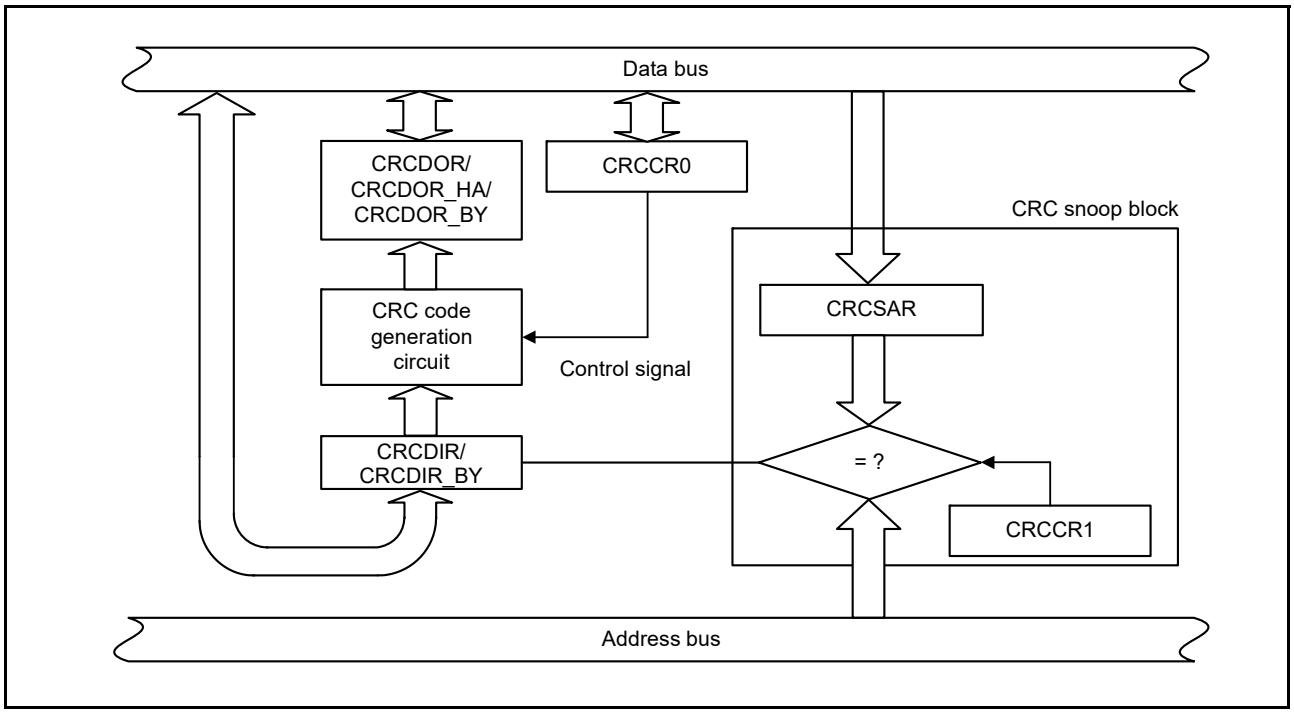


Figure 35.1 CRC calculator block diagram

35.2 Register Descriptions

35.2.1 CRC Control Register 0 (CRCCR0)

Address(es): CRC.CRCCR0 4007 4000h

	b7	b6	b5	b4	b3	b2	b1	b0
	DORCLR	LMS	—	—	—	GPS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	GPS[2:0]	CRC Generating Polynomial Switching	b2 b0 0 0 0: No calculation is executed 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 ($X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$) Other: No calculation is executed.	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	LMS	CRC Calculation Switching	0: Generates CRC for LSB-first communication 1: Generates CRC for MSB-first communication.	R/W
b7	DORCLR	CRCDOR/CRCDOR_HA/ CRCDOR_BY Register Clear	1: Clears the CRCDOR/CRCDOR_HA/CRCDOR_BY register. This bit is read as 0.	W*1

Note 1. This bit must always be set to 1 when writing to this register.

DORCLR bit (CRCDOR/CRCDOR_HA/CRCDOR_BY)

Write 1 to this bit so that the CRCDOR/CRCDOR_HA/CRCDOR_BY register is set to 0000 0000h. This bit is read as 0. Only 1 can be written.

LMS bit (CRC Calculation Switching)

Set this bit to select 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 35.3, Operation.

GPS[2:0] bit (CRC Generating Polynomial Switching)

Set these bits to select the CRC Generating Polynomial.

35.2.2 CRC Control Register 1 (CRCCR1)

Address(es): CRC.CRCCR1 4007 4001h

	b7	b6	b5	b4	b3	b2	b1	b0
	CRCSEN	CRCSWR	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CRCSWR	Snoop-On-Write/Read Switch	0: Snoop-on-read 1: Snoop-on-write.	R/W
b7	CRCSEN	Snoop Enable	0: Disabled 1: Enabled.	R/W

CRCSWR bit (Snoop-On-Write/Read Switch)

This bit selects the direction of the access in the address monitoring function.

When setting this bit to 0 (initial value), the CRC snoop operation to read a specific register address is valid. Similarly, when setting this bit to 1, the CRC snoop operation to write a specific register address is valid.

CRCSEN bit (Snoop Enable)

When setting this bit to 1, the CRC snoop operation is valid. When setting this bit to 0, the CRC snoop operation is invalid.

35.2.3 CRC Data Input Register (CRCDIR/CRCDIR_BY)

Address(es): CRC.CRCDIR/CRCDIR_BY 4007 4004h

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

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

The CRCDIR register is a read/write 32-bit register to write data for CRC-32 or CRC-32C calculation. The CRCDIR_BY register is a read/write 8-bit register to write data for CRC-8, CRC-16, or CRC-CCITT calculation.

35.2.4 CRC Data Output Register (CRCDOR/CRCDOR_HA/CRCDOR_BY)

Address(es): CRC.CRCDOR/CRCDOR_HA/CRCDOR_BY 4007 4008h

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

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

The CRCDOR register is a read/write 32-bit register for CRC-32 or CRC-32C.

The CRCDOR_HA register is a read/write 16-bit register for CRC-16 or CRC-CCITT.

The CRCDOR_BY register is a read/write 8-bit register for CRC-8.

Because its initial value is 0000 0000h, rewrite the CRCDOR/CRCDOR_HA/CRCDOR_BY register to perform calculation 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 transferred data and the result is 0000 0000h, there is no CRC error.

When an 8-bit CRC ($X^8 + X^2 + X + 1$ polynomial) is in use, the valid CRC code is obtained in CRCDOR_BY.

When a 16-bit CRC ($X^{16} + X^{15} + X^2 + 1$ or $X^{16} + X^{12} + X^5 + 1$ polynomial) is in use, the valid CRC code is obtained in CRCDOR_HA.

35.2.5 Snoop Address Register (CRCSAR)

Address(es): CRC.CRCSAR 4007 400Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CRCSA[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CRCSA[13:0]	Register Snoop Address	Set the TDR or RDR address in the SCI module to snoop	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CRCSA[13:0] bits (Register Snoop Address)

Set these bits to the lower 14-bit of register address monitored by the CRC snoop operation.

Only the following address can be used for the CRCSA[13:0] bits:

- 4007 0003h: SCI0.TDR, 4007 0005h: SCI0.RDR
- 4007 0023h: SCI1.TDR, 4007 0025h: SCI1.RDR
- 4007 0043h: SCI2.TDR, 4007 0045h: SCI2.RDR
- 4007 0063h: SCI3.TDR, 4007 0065h: SCI3.RDR
- 4007 0083h: SCI4.TDR, 4007 0085h: SCI4.RDR
- 4007 0123h: SCI9.TDR, 4007 0125h: SCI9.RDR
- 4007 000Fh: SCI0.FTDRL,4007 0011h: SCI0.FRDRL
- 4007 002Fh: SCI1.FTDRL,4007 0031h: SCI1.FRDRL
- 4007 004Fh: SCI2.FTDRL,4007 0051h: SCI2.FRDRL
- 4007 006Fh: SCI3.FTDRL,4007 0071h: SCI3.FRDRL
- 4007 008Fh: SCI4.FTDRL,4007 0091h: SCI4.FRDRL
- 4007 012Fh: SCI9.FTDRL,4007 0131h: SCI9.FRDRL.

35.3 Operation

35.3.1 Basic Operation

The CRC calculator generates CRC codes for use in LSB-first or MSB-first transfer.

The following examples illustrate CRC code generation for input data (F0h) 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 an 32-bit CRC is in use, the valid bits of the CRC code are obtained in CRCDOR.

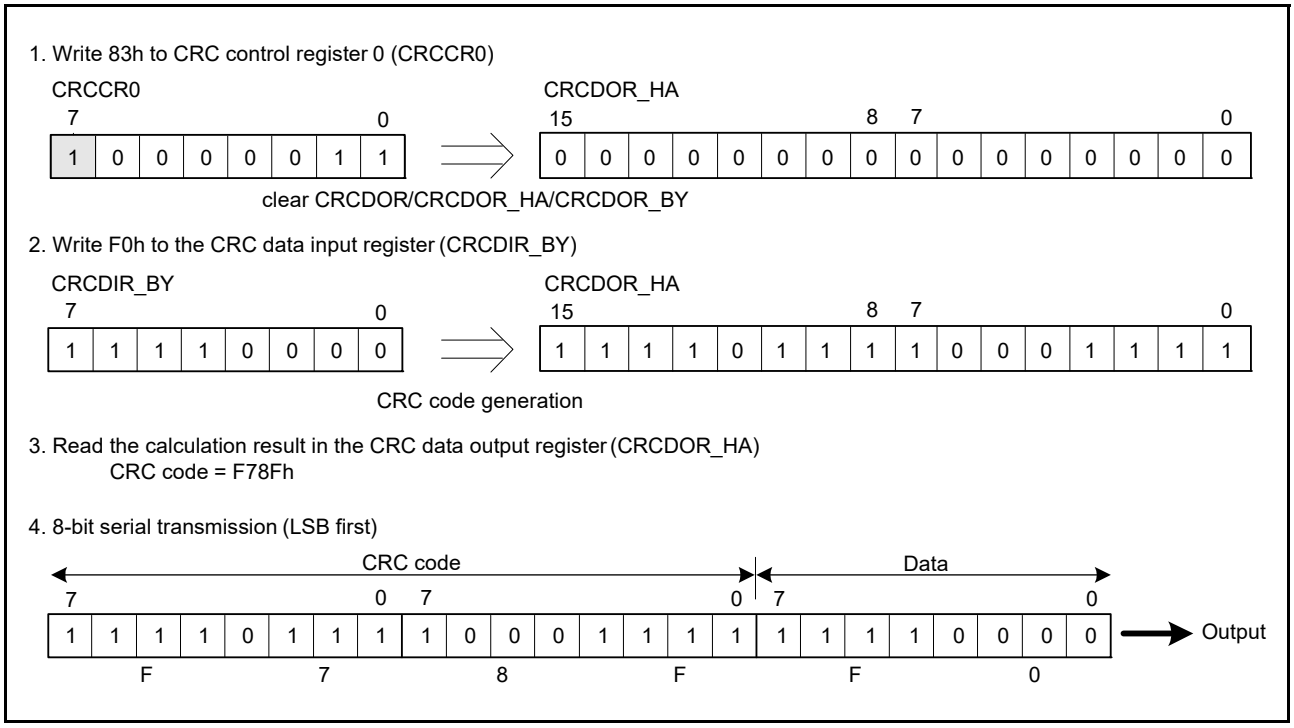


Figure 35.2 LSB-first data transmission

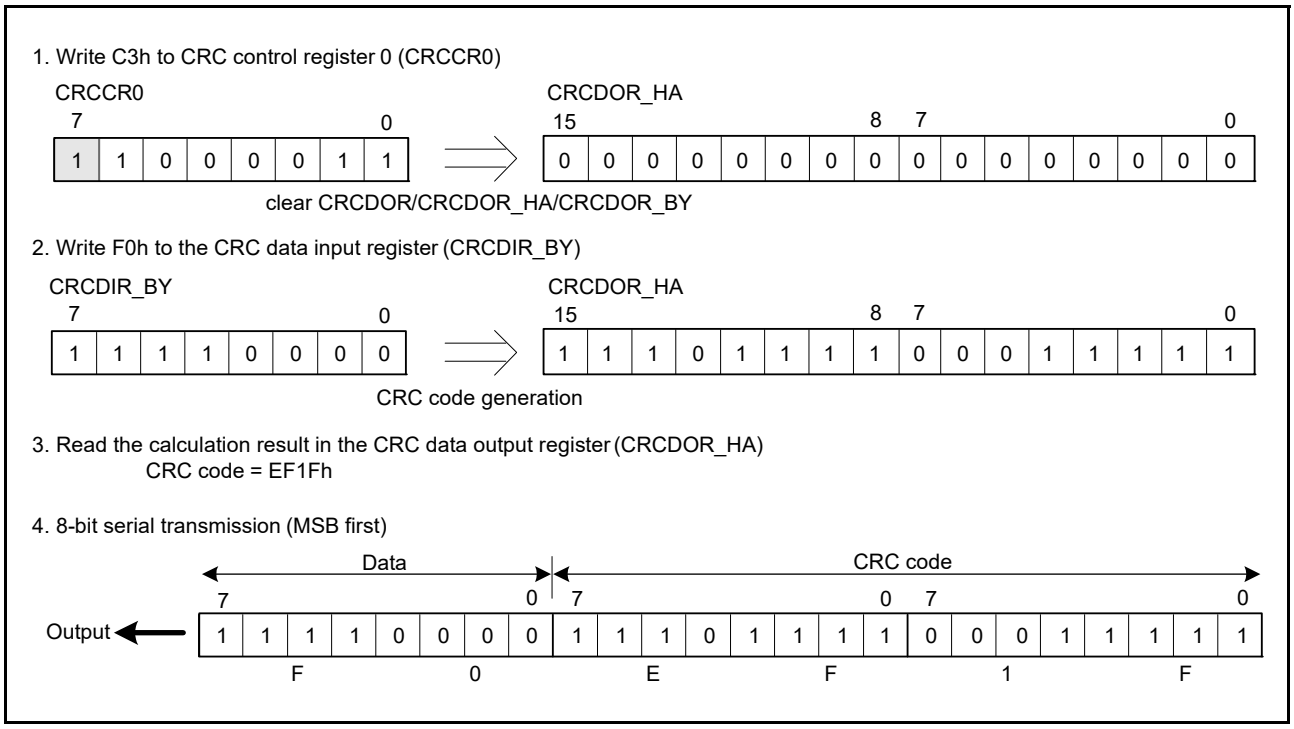


Figure 35.3 MSB-first data transmission

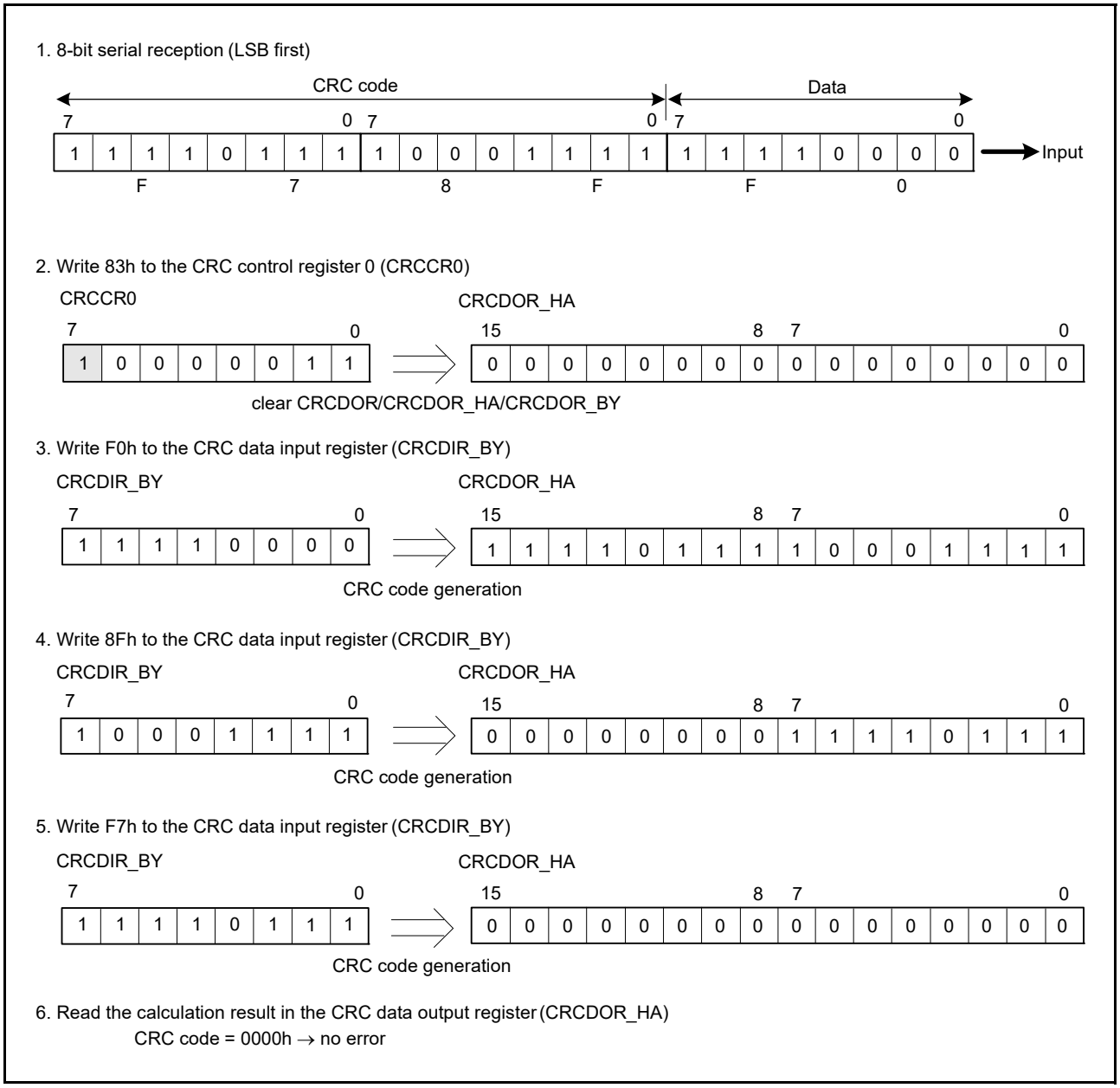


Figure 35.4 LSB-first data reception

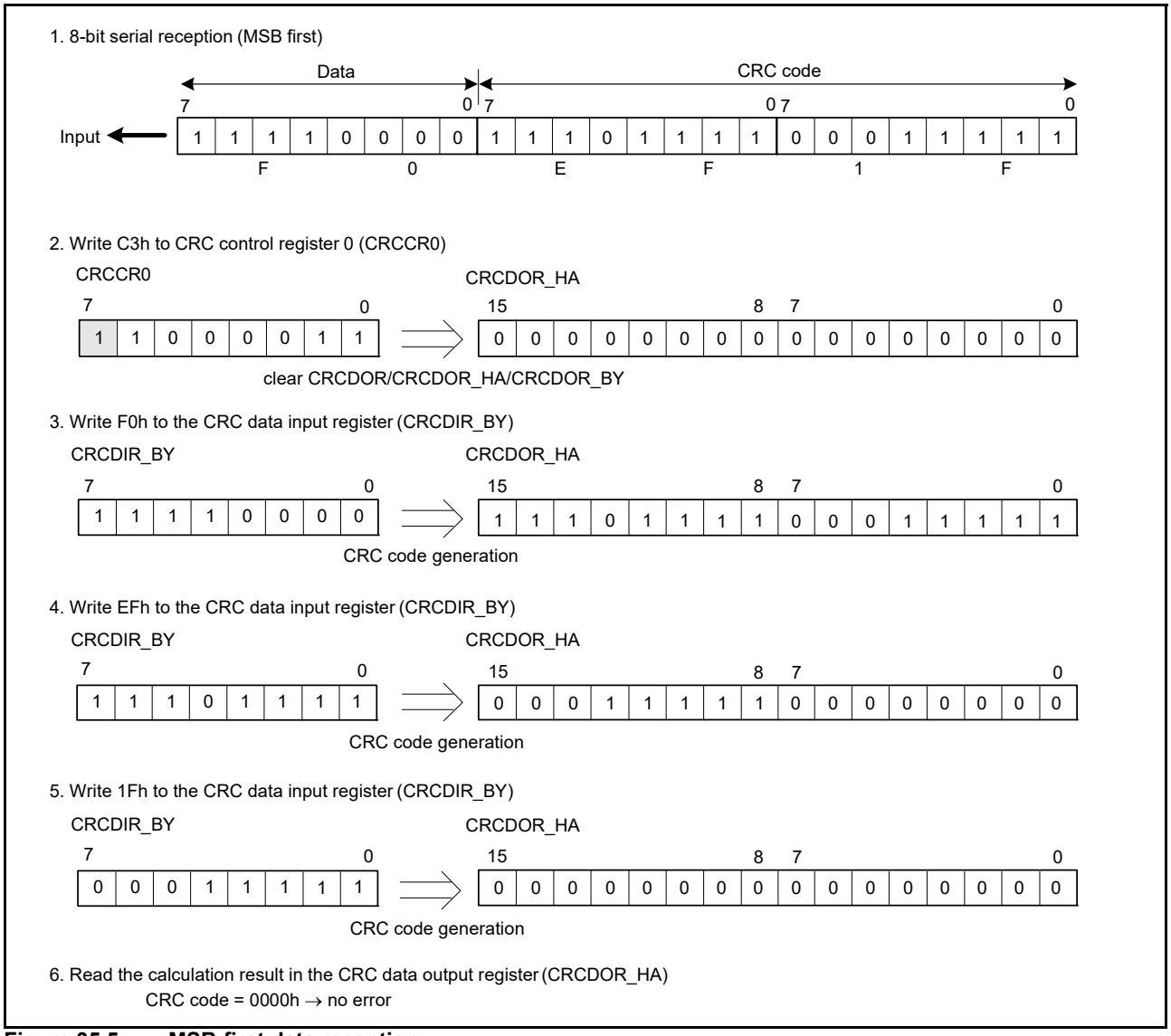


Figure 35.5 MSB-first data reception

35.3.2 CRC Snoop

The CRC snoop function monitors reads from and writes to a specific register address and performs CRC calculation on the data read from and written to that register address automatically. Because the CRC snoop 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 addresses specified in the Snoop Address Register (CRCSAR) are subject to the CRC snoop. The CRC snoop is useful in monitoring writes to the serial transmit buffer, and reads from the serial receive buffer.

To use this function, write a target I/O register address to bits CRCSA13 to CRCSA0 in the CRCSAR register, and set the 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 address, or set the CRCSWR bit in the CRCCR1 register to 0 to enable snooping on reads from the target address.

When setting the CRCSEN bit to 1, CRCSWR bit to 1, and writing data to a target I/O register address in a bus master module such as CPU, DMA, and DTC, the CRC calculator stores the data in the CRCDIR_BY register and performs CRC calculation. Similarly, when setting the CRCSEN bit to 1, CRCSWR bit to 0, and reading data in a target I/O register address in a bus master module such as CPU, DMA, and DTC, the CRC calculator stores the data in the CRCDIR_BY register and performs CRC calculation.

CRC calculation is performed 1 byte at a time. When the target I/O register address is accessed in words (16 bits) or long words (32 bits), the CRC code is generated on the lower byte (1 byte) of data.

35.4 Usage Notes

35.4.1 Settings for the Module-Stop State

The Module Stop Control Register C (MSTPCRC) can enable or disable operation of the CRC calculator. The CRC is stopped after a reset. The registers become accessible on release from the module-stop state. For details, see section 11, Low Power Modes.

35.4.2 Note on Transmission

The sequence of transmission for the CRC code changes according to whether transmission is LSB-first or MSB-first.

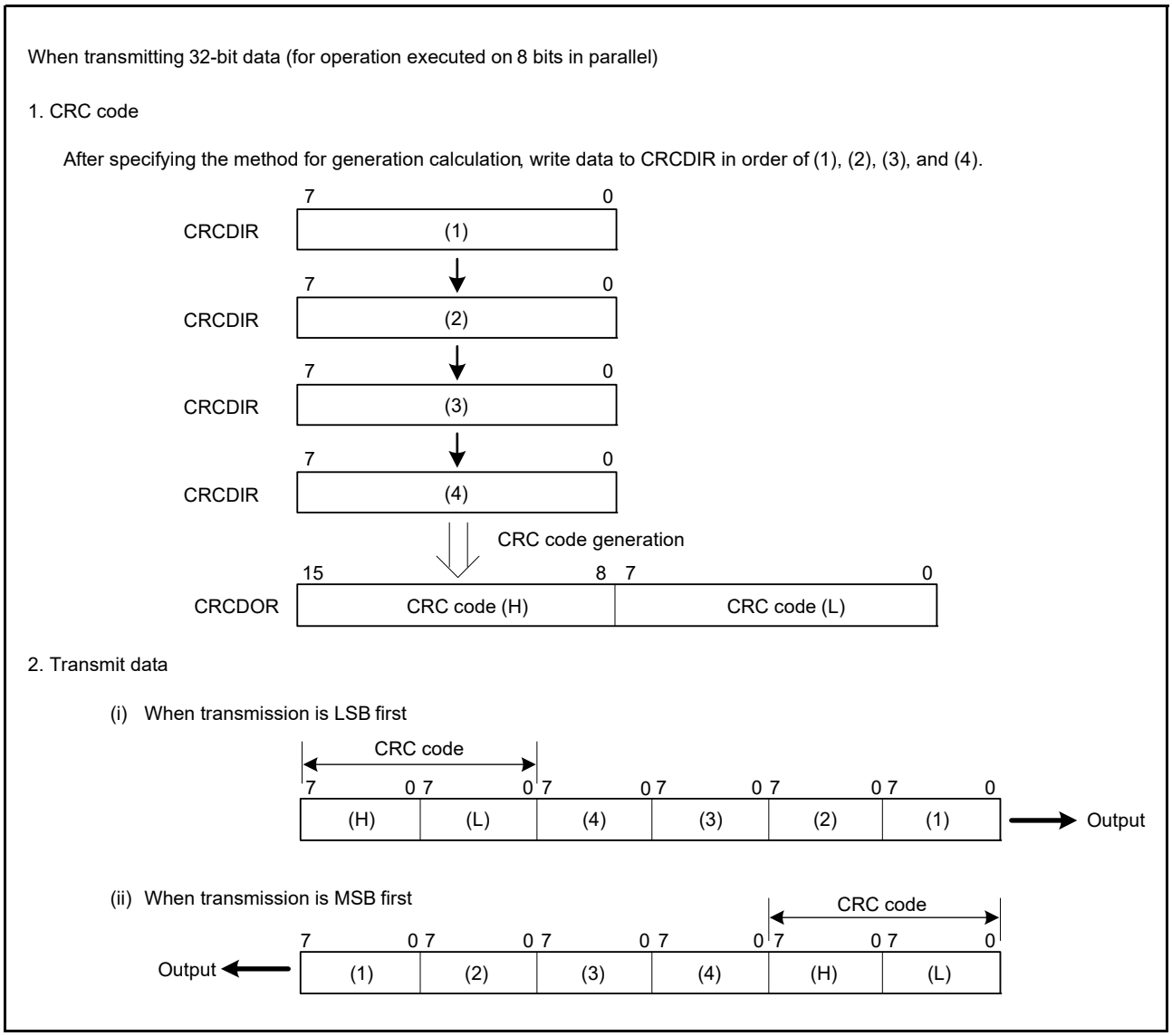


Figure 35.6 LSB-first and MSB-first data transmission

36. Serial Sound Interface (SSI)

The Serial Sound Interface (SSI) peripheral interfaces with digital audio devices to transmit PCM audio data over a serial bus with the MCU. The SSI can be operated as a slave or master receiver, transmitter, or transceiver to suit multiple applications. The SSI includes 8-stage FIFO buffers in the receiver and transmitter, and it supports interrupts and DMA-driven data reception and transmission.

36.1 Overview

Table 36.1 SSI specifications

Parameter	Description
Number of channels	Two channels, SSI0 and SSI1
Operating mode	Non-compression mode
Transfer formats	• SSI format • MSB-first format, with selectable left and right alignment
Function	• Serves as both a transmitter and a receiver, with full-duplex communications on channel 0 • Support for multiple audio formats • Serial bit clock, SSISCK, configurable to 16, 32, 48, and 64 fs (fs: sampling rate) • Master clock input from the master clock pin for audio (AUDIO_CLK) or GPT output (GTIOC1A) • 8-stage FIFO buffers in transmitter and receiver • Stop word select (SSIWS) selectable when data transfer is stopped.
Interrupt sources	Three sources • Communication error for transmit underflow, transmit overflow, receive underflow, receive overflow, and idle • Receive data full • Transmit data empty.
Module-stop function	Module-stop state can be set.

Figure 36.1 shows a block diagram of SSI0, and Figure 36.2 shows a block diagram of SSI1.

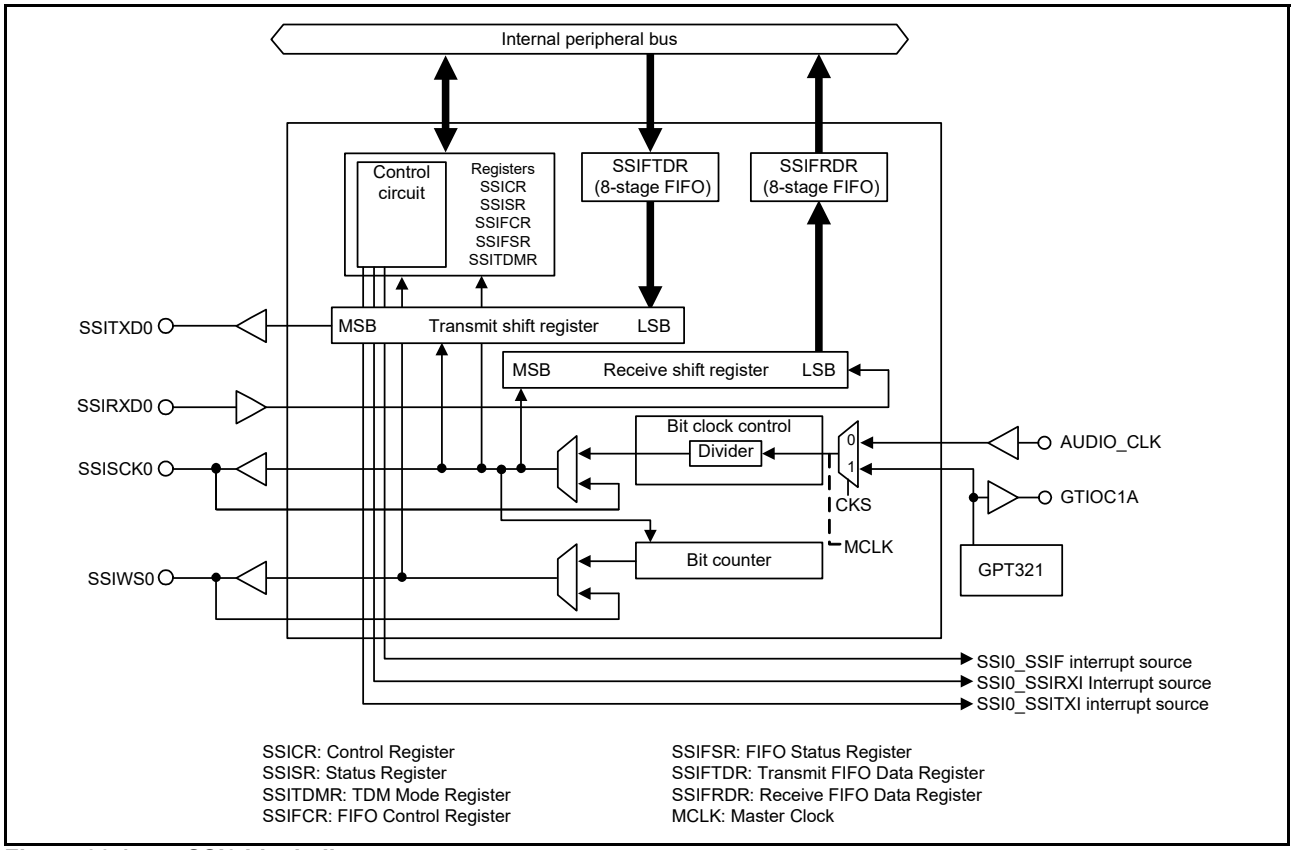


Figure 36.1 SSI0 block diagram

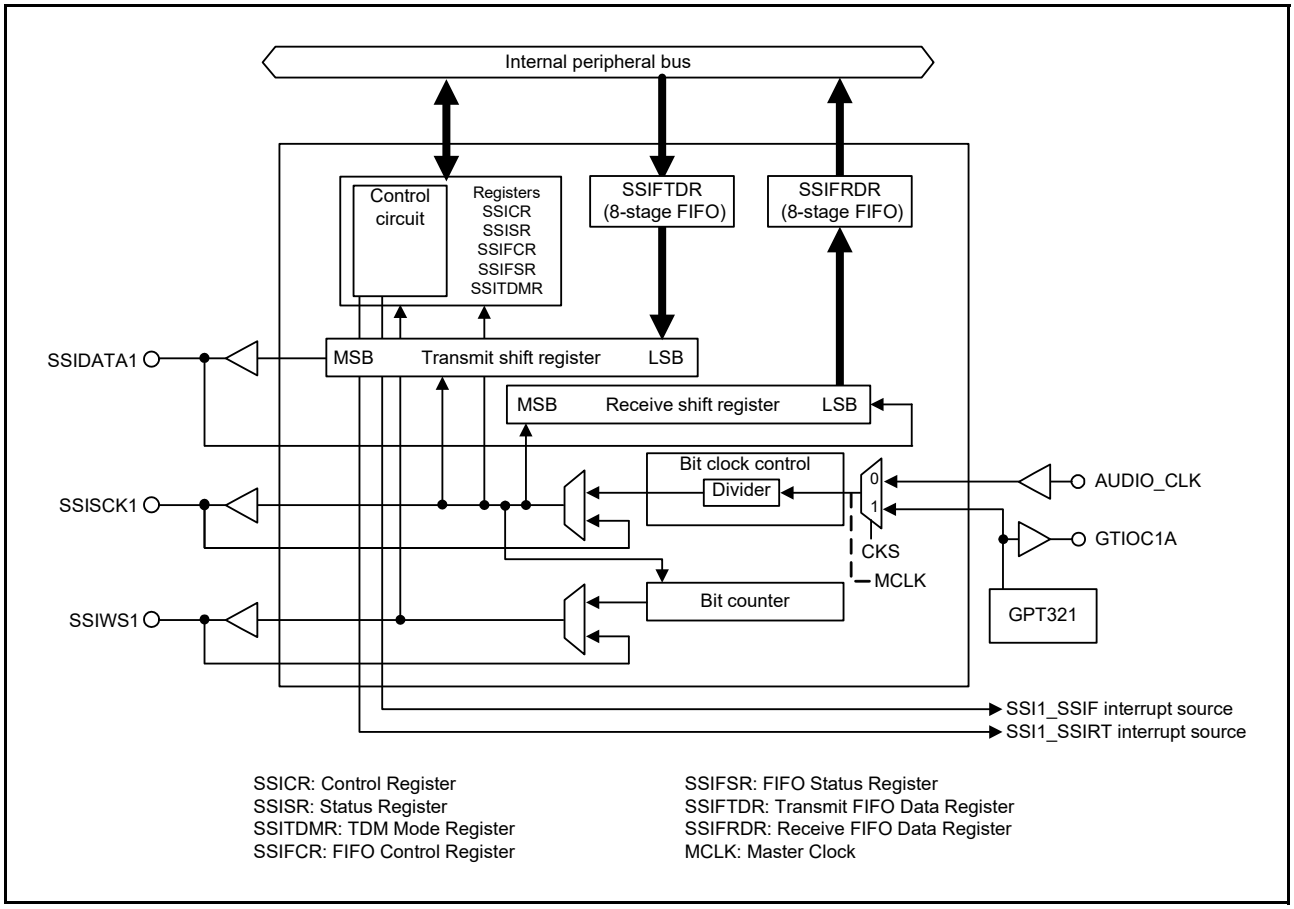


Figure 36.2 SSI1 block diagram

Table 36.2 lists the I/O pins of the SSI.

Table 36.2 SSI I/O pins

Module	Pin name	I/O	Description
SSI0	SSISCK0	I/O	Serial bit clock pin
	SSIWS0	I/O	Word selection pin
	SSITXD0	Output	Serial data output pin
	SSIRXD0	Input	Serial data input pin
SSI1	SSISCK1	I/O	Serial bit clock pin
	SSIWS1	I/O	Word selection pin
	SSIDATA1	I/O	Serial data input/output pin
SSI0, SSI1	AUDIO_CLK	Input	Master clock for audio pin (input master clock)

36.2 Register Description

36.2.1 Control Register (SSICR)

Address(es): SSI0.SSICR 4004 E000h, SSI1.SSICR 4004 E100h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	CHNL[1:0]		DWL[2:0]			SWL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	REN	Receive Enable	0: Reception disabled 1: Reception enabled.	R/W
b1	TEN	Transmit Enable	0: Transmission disabled 1: Transmission enabled.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	MUEN	Mute Enable*1	0: Not muted 1: Muted.	R/W
b7 to b4	CKDV[3:0]	Serial Bit Clock Frequency Setting*3	b7 b6 b5 b4 0 0 0 0: MCLK 0 0 0 1: MCLK/2 0 0 1 0: MCLK/4 0 0 1 1: MCLK/8 0 1 0 0: MCLK/16 0 1 0 1: MCLK/32 0 1 1 0: MCLK/64 0 1 1 1: MCLK/128 1 0 0 0: MCLK/6 1 0 0 1: MCLK/12 1 0 1 0: MCLK/24 1 0 1 1: MCLK/48 1 1 0 0: MCLK/96. Other settings are prohibited.	R/W
b8	DEL	Serial Data Delay*3	0: Compatible with SSI format: one clock cycle delay between SSIWS and SSIDATA 1: Compatible with MSB-first and left- or right-aligned format: no delay between SSIWS and SSIDATA.	R/W
b9	PDTA	Parallel Data Allocation*3	When data word length is 8 or 16 bits: 0: Parallel data allocated at the lower bytes/half word (SSIFTDR, SSIFRDR) are transferred prior to the upper bytes/half word 1: Parallel data allocated at the upper bytes/half word (SSIFTDR, SSIFRDR) are transferred prior to the lower bytes/half word. When data word length is 18, 20, 22, or 24 bits: 0: Parallel data (SSIFTDR, SSIFRDR) left-aligned 1: Parallel data (SSIFTDR, SSIFRDR) right-aligned.	R/W
b10	SDTA	Serial Data Alignment*3	0: Transmitting and receiving order: serial data, then padding bits 1: Transmitting and receiving order: padding bits then serial data.	R/W
b11	SPDP	Serial Padding Polarity*3	0: Padding data is 0 1: Padding data is 1.	R/W
b12	SWSP	Word Select Polarity	0: SSIWS is low for 1st system word, high for 2nd system word 1: SSIWS is high for 1st system word, low for 2nd system word.	R/W

Bit	Symbol	Bit name	Description	R/W
b13	SCKP	Serial Bit Clock Polarity*3	0: SSIWS and SSIDATA change on SSISCK falling edge (sampled on SCK rising edge) 1: SSIWS and SSIDATA change on SSISCK rising edge (sampled on SCK falling edge).	R/W
b14	SWSD	Word Select Direction*2, *3	0: SSIWS pin is input (slave mode) 1: SSIWS pin is output (master mode).	R/W
b15	SCKD	Serial Bit Clock Direction *2, *3	0: SSISCK pin is input (slave mode) 1: SSISCK pin is output (master mode).	R/W
b18 to b16	SWL[2:0]	System Word Length*3	Sets the system word length to the bit clock frequency/2 fs: b18 b16 0 0 0: 8 bits (serial bit clock frequency = 16 fs) 0 0 1: 6 bits (serial bit clock frequency = 32 fs) 0 1 0: 24 bits (serial bit clock frequency = 48 fs) 0 1 1: 32 bits (serial bit clock frequency = 64 fs). Other settings are prohibited.	R/W
b21 to b19	DWL[2:0]	Data Word Length*3	b21 b19 0 0 0: 8 bits 0 0 1: 16 bits 0 1 0: 18 bits 0 1 1: 20 bits 1 0 0: 22 bits 1 0 1: 24 bits. Other settings are prohibited.	R/W
b23, b22	CHNL[1:0]	Channels*3	b23 b22 0 0: One channel. Other settings are prohibited.	R/W
b24	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b25	IIEN	Idle Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled.	R/W
b26	ROIEN	Receive FIFO Overflow Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled.	R/W
b27	RUIEN	Receive FIFO Underflow Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled.	R/W
b28	TOIEN	Transmit FIFO Overflow Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled.	R/W
b29	TUIEN	Transmit FIFO Underflow Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled.	R/W
b30	CKS	Audio Clock Select*3	0: AUDIO_CLK input 1: GTIOC1A (GPT output).	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

	SCKP bit = 0	SCKP bit = 1
SSIDATA input sampling timing for reception	SSISCK rising edge	SSISCK falling edge
SSIDATA output change timing for transmission	SSISCK falling edge	SSISCK rising edge
SSIWS input sampling timing in slave mode (SWSD bit = 0)	SSISCK rising edge	SSISCK falling edge
SSIWS output change timing in master mode (SWSD bit = 1)	SSISCK falling edge	SSISCK rising edge

Note 1. When the SSI module is muted, 0 is transmitted regardless of the value of the serial data, but data transfer is not stopped. Because the number of data units in the transmit FIFO decreases, write dummy data to the SSIFTDR register to prevent a transmit underflow. When the MUEN bit is set to 1, the SSITXD0 and SSIDATA1 pins are immediately is set to 0 without synchronizing SSIWS.

Note 2. Set the SCKD and SWSD bits to the same value. Other settings are prohibited.

Note 3. Rewriting is allowed only in the idle state.

REN bit (Receive Enable)

The REN bit enables or disables receive operation. Setting this bit to 1 starts a receive operation.

TEN bit (Transmit Enable)

The TEN bit enables or disables transmit operation. Setting this bit to 1 starts a transmit operation.

The SSITXD0 pin of SSI0 is always output when the I/O port function is selected, regardless of the TEN bit setting. The SSIDATA1 pin of SSI1 is output when the TEN bit is 1 and input when the TEN bit is 0, when the I/O port function is selected for SSIDATA1.

Table 36.3 SSITXD0, SSIRXD0, and SSIDATA1 pin states

Register settings			SSI0		SSI1
SSICR setting	TEN	REN	SSITXD0	SSIRXD0	SSIDATA1
SSI selected	0	0	Output	Input	Input
	0	1	Output	Input	Input
	1	0	Output	Input	Output
	1	1	Output	Input	—
SSI deselected	x	x	I/O port	I/O port	I/O port

x: Don't care
—: Settings prohibited.
I/O port: Depends on the settings of the I/O port and multi-function pin controller.

CKDV[3:0] bits (Serial Bit Clock Frequency Setting)

The CKDV[3:0] bits select the frequency of the serial bit clock in master mode. The setting in these bits is ignored in slave mode, because the slave mode uses the input clock from the SSISCK pin. The serial bit clock is the operating clock for the shift register.

Calculation example when fs (sampling rate) = the SSIWS frequency = 96 kHz, and the system word length = 32 bits:

The bit clock frequency = 96 kHz × 32 bits × 2 = 6.144 MHz.

For this example, set CKDV[3:0] = 0001b (MCLK/2) when MCLK = 12.288 MHz.

PDTA bit (Parallel Data Allocation)

The PDTA setting specifies how data is stored in the SSIFRDR register in receive mode and in the SSIFTDR register in transmit mode. During reception, the SSI stores the data received from the serial bus in SSIFRDR according to the PDTA bit setting. During transmission, the SSI shifts the data stored in SSIFTDR in the transmit shift register, and transmits the data to the serial bus according to the PDTA bit setting.

When PDTA = 0

DWL[2:0] Bits	SSIFTDR[31:0]/SSIFRDR[31:0] Registers
000b	31 24 23 16 15 8 7 0 4th word | 3rd word | 2nd word | 1st word
001b	31 16 15 0 2nd word | 1st word
010b	31 14 13 0 Valid | Invalid
011b	31 12 11 0 Valid | Invalid
100b	31 10 9 0 Valid | Invalid
101b	31 8 7 0 Valid | Invalid

When PDTA = 1

DWL[2:0] Bits	SSIFTDR[31:0]/SSIFRDR[31:0] Registers
000b	31 24 23 16 15 8 7 0 1st word | 2nd word | 3rd word | 4th word
001b	31 16 15 0 1st word | 2nd word
010b	31 18 17 0 Invalid | Valid
011b	31 20 19 0 Invalid | Valid
100b	31 22 21 0 Invalid | Valid
101b	31 24 23 0 Invalid | Valid

CHNL[1:0] bits (Channels)

The CHNL[1:0] bits select the number of channels to be decoded in each system word. Set these bits to 00b for this SSI.

36.2.2 Status Register (SSISR)

Address(es): SSI0.SSISR 4004 E004h, SSI1.SSISR 4004 E104h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	TCHNO[1:0]		TSWNO	RCHNO[1:0]		RSWNO	IDST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	IDST	Idle Status Flag	0: SSI communication in progress 1: SSI communication idle.	R
b1	RSWNO	Receive System Word Number	Receive word number	R
b3, b2	RCHNO[1:0]	Receive Channel Number	These bits are read as 0	R
b4	TSWNO	Transmit System Word Number	Transmit word number	R
b6, b5	TCHNO[1:0]	Transmit Channel Number	These bits are read as 0	R
b24 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25	IIRQ	Idle Interrupt Status Flag	0: Not in idle state 1: In idle state.	R
b26	ROIRQ	Receive Overflow Interrupt Status Flag	0: No receive overflow occurred 1: Receive overflow occurred.	R/(W)*1
b27	RUIRQ	Receive Underflow Interrupt Status Flag	0: No receive underflow occurred 1: Receive underflow occurred.	R/(W)*1
b28	TOIRQ	Transmit Overflow Interrupt Status Flag	0: No transmit overflow occurred 1:Transmit overflow occurred.	R/(W)*1
b29	TUIRQ	Transmit Underflow Interrupt Status Flag	0: No transmit underflow occurred 1: Transmit underflow occurred.	R/(W)*1
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit can be set to 0 by writing 0 after reading it as 1.

IDST flag (Idle Status Flag)

The IDST flag indicates that the SSI is in the idle state when communication is stopped. It is set to 0 when communication starts after the SSICR.TEN bit or SSICR.REN bit is set to 1. This flag also is set to 1 if both the SSICR.TEN and SSICR.REN bits are set to 0 and system word communication is complete.

If an external device stops inputting the serial bit clock before communication is complete, this flag does not set to 1.

RSWNO bit (Receive System Word Number)

The initial value of the RSWNO bit is 1, and its value is inverted when data is transferred from the receive shift register to the SSIFRDR register. This bit is initialized to 1 when the REN bit value changes from 0 to 1. When the data word length specified in the SSICR.DWL[2:0] bits is 18 bits or more, this bit indicates which system word the data in the SSIFRDR register represents.

TSWNO bit (Transmit System Word Number)

The TSWNO bit indicates the current word number. The initial value of this bit is 1, and its value is inverted when data is transferred from the SSIFTDR register to the transmit shift register. This bit is initialized to 1 when the TEN bit value changes from 0 to 1. When the data word length specified in the SSICR.DWL[2:0] bits is 18 bits or more, this bit indicates the system word that is in the data transferred from the SSIFTDR register to the transmit shift register.

IIRQ flag (Idle Interrupt Status Flag)

The IIRQ flag indicates whether the SSI module is in idle state. To allow polling, this flag is set regardless of the SSICR.IIEN setting. This interrupt can be masked by setting the SSICR.IIEN bit to 0, but it cannot be cleared by writing 0 to this flag. If IIRQ flag = 1 and SSICR.IIEN bit = 1, an interrupt occurs.

ROIRQ flag (Receive Overflow Interrupt Status Flag)

The ROIRQ flag indicates that receive data was supplied at a higher rate than required. If a receive overflow occurs, stop reception and restart from the beginning of the operation flow. This flag is set to 1 regardless of the SSICR.ROIEN setting. This flag can be set to 0 by writing 0 after reading it as 1.

If ROIRQ = 1 and the SSICR.ROIEN bit = 1, an interrupt occurs.

If ROIRQ = 1, the data was transferred from the receive shift register to the SSIFRDR register while the receive FIFO was full (SSIFSR.RDC[3:0] flags = 8h). This might lead to the loss of data.

Note: When an overflow occurs, the current data in the SSI data buffer is overwritten by the next incoming data from the SSI interface.

RUIRQ flag (Receive Underflow Interrupt Status Flag)

The RUIRQ flag indicates that receive data was supplied at a lower rate than required. If a receive underflow occurs, stop reception and restart from the beginning of the operation flow. This flag is set to 1 regardless of the SSICR.RUIEN setting. This flag can be set to 0 by writing 0 after reading it as 1.

If RUIRQ flag = 1 and SSICR.RUIEN bit = 1, an interrupt occurs.

If RUIRQ flag = 1, the SSIFRDR register was read while the receive FIFO is empty (SSIFSR.RDC[3:0] flags = 0h). This might cause invalid receive data to be stored.

TOIRQ flag (Transmit Overflow Interrupt Status Flag)

The TOIRQ flag indicates that transmit data was supplied at a higher rate than required. If a transmit overflow occurs, stop transmission and restart from the beginning of the operation flow.

This flag is set to 1 regardless of the SSICR.TOIEN setting. This flag can be set to 0 by writing 0 after reading it as 1.

If TOIRQ flag = 1 and SSICR.TOIEN bit = 1, an interrupt occurs.

If TOIRQ flag = 1, the SSIFTDR register had data written to it while the transmit FIFO is full (SSIFSR.TDC[3:0] flags = 8h). This might lead to the loss of data.

TUIRQ flag (Transmit Underflow Interrupt Status Flag)

The TUIRQ flag indicates that transmit data was supplied at a lower rate than required. If a transmit underflow occurs, stop transmission and restart from the beginning of the operation flow.

This flag is set to 1 regardless of the SSICR.TUIEN setting. This flag can be set to 0 by writing 0 after reading it as 1.

If TUIRQ flag = 1 and SSICR.TUIEN bit = 1, an interrupt occurs.

If TUIRQ flag = 1, the SSIFTDR register did not have data written to it before it was required for transmission. This might lead to the same data being transmitted a second time.

Note: When a transmit underflow occurs, the last data input to SSIFTDR is transmitted until the SSI module is in the idle state after transmission stops.

36.2.3 FIFO Control Register (SSIFCR)

Address(es): SSI0.SSIFCR 4004 E010h, SSI1.SSIFCR 4004 E110h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	TTRG[1:0]		RTRG[1:0]		TIE	RIE	TFRST	RFRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RFRST	Receive FIFO Data Register Reset*4	0: Clear receive FIFO data reset 1: Initiate receive FIFO data reset.	R/W
b1	TFRST	Transmit FIFO Data Register Reset*4	0: Clear transmit FIFO data reset 1: Initiate transmit FIFO data reset.	R/W
b2	RIE	Receive FIFO Data Full Interrupt Enable	0: Receive FIFO data full interrupt request (SSI0_SSIRXI, SSI1_SSIRT) disabled 1: Receive FIFO data full interrupt request (SSI0_SSIRXI, SSI1_SSIRT) enabled.*1	R/W
b3	TIE	Transmit FIFO Data Empty Interrupt Enable	0: Transmit FIFO data empty interrupt request (SSI0_SSITXI, SSI1_SSIRT) disabled 1: Transmit FIFO data empty interrupt request (SSI0_SSITXI, SSI1_SSIRT) enabled.*2	R/W
b5, b4	RTRG[1:0]	Receive FIFO Threshold Setting Trigger*4	b5 b4 0 0: 1 0 1: 2 1 0: 4 1 1: 6.	R/W
b7, b6	TTRG[1:0]	Transmit FIFO Threshold Setting Trigger*4	b7 b6 0 0: 7 (1)*3 0 1: 6 (2)*3 1 0: 4 (4)*3 1 1: 2 (6).*3	R/W
b15 to b8	—	Reserved	These bits are read as undefined. The write value should be 0.	R/W
b16	SSIRST	SSI Software Reset	0: Clear SSI software reset 1: Initiate SSI software reset.	R/W
b30 to b17	—	Reserved	These bits are read as undefined. The write value should be 0.	R/W
b31	AUCKE	Master Clock Enable*4	0: Master clock disabled 1: Master clock enabled.	R/W

Note 1. The receive data full request can be cleared by setting the SSIFSR.RDF flag to 0 (see the description of the SSIFSR.RDF flag for details) or RIE bit to 0.

Note 2. The transmit data empty request can be cleared by setting the SSIFSR.TDE flag to 0 (see the description of the SSIFSR.TDE flag for details) or TIE bit to 0.

Note 3. The values in parentheses are the number of empty stages in SSIFTDR at which the SSIFSR.TDE flag is set.

Note 4. Rewriting is allowed only in the idle state.

The SSIFCR register resets the number of the data stored in the SSIFTDR and SSIFRDR registers, and specifies transmit FIFO and receive FIFIO threshold values.

RFRST bit (Receive FIFO Data Register Reset)

The RFRST bit invalidates the data in the SSIFRDR register to reset the FIFO to an empty state.

TFRST bit (Transmit FIFO Data Register Reset)

The TFRST bit invalidates the data in the SSIFTDR register to reset the FIFO to an empty state.

RIE bit (Receive FIFO Data Full Interrupt Enable)

The RIE bit enables or disables generation of receive FIFO data full interrupt (SSI0_SSIRXI or SSI1_SSIRT) requests when the SSIFSR.RDF flag is set to 1 during reception.

TIE bit (Transmit FIFO Data Empty Interrupt Enable)

The TIE bit enables or disables generation of transmit FIFO data empty interrupt (SSI0_SSITXI or SSI1_SSIRT) requests when the SSIFSR.TDE flag is set to 1 during transmit operation.

RTRG[1:0] bits (Receive FIFO Threshold Setting Trigger)

The RTRG bits specify the receive FIFO threshold value. When the amount of received data stored in the SSIFRDR register (receive FIFO) is equal to or greater than the value specified by the RTRG[1:0] bits, the SSIFSR.RDF flag is set to 1 and reading the received data is requested. If the SSIFCR.RIE bit is 1, a receive FIFIO data full interrupt (SSI0_SSIRXI or SSI1_SSIRT) request is generated.

TTRG[1:0] bits (Transmit FIFO Threshold Setting Trigger)

The TTRG bits specify the transmit FIFO threshold value. When the amount of transmit data stored in the SSIFTDR register (transmit FIFO) is equal to or less than the value specified by the TTRG[1:0], the SSIFSR.TDE flag is set to 1 and writing the transmit data is requested. If the SSIFCR.TIE bit is 1, a transmit FIFO data empty interrupt (SSI0_SSITXI or SSI1_SSIRT) request is generated.

SSIRST bit (SSI Software Reset)

Writing 1 to the SSIRST bit initializes the SSI internal status, registers other than the SSIFCR register, and bits other than this bit in the SSIFCR register. Because this bit is not automatically set to 0, confirm that 1 is written to it before writing 0. Do not write 0 to this bit and 1 to other bits at the same time. After modifying this bit, confirm that its value is modified before proceeding to the next operation.

36.2.4 FIFO Status Register (SSIFSR)

Address(es): SSI0.SSIFSR 4004 E014h, SSI1.SSIFSR 4004 E114h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	RDC[3:0]				—	—	—	—	—	—	—	RDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	RDF	Receive Data Full Flag	0: Amount of received data in the SSIFRDR register is less than the specified receive trigger number 1: Amount of received data in the SSIFRDR register is equal to or greater than the specified receive trigger number.	R/(W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	RDC[3:0]	Receive Data Indicate Flag	Indicates the number of data units stored in the SSIFRDR register.	R
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	TDE	Transmit Data Empty Flag	0: Amount of data for transmission in the SSIFTDR register is greater than the specified transmit trigger number 1: Amount of data for transmission in the SSIFTDR register is equal to or less than the specified transmit trigger number.*2	R/(W)*1
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit name	Description	R/W
b27 to b24	TDC[3:0]	Transmit Data Indicate Flag	Indicates the number of data units stored in the SSIFTDR register.	R
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit can be set to 0 by writing 0 after reading it as 1.

Note 2. Because the SSIFTDR register is an 8-stage FIFO register, the amount of data that can be written to it while TDE flag = 1 is (8 - TDC[3:0] setting). Writes of additional data are ignored. The number of data bytes in the SSIFTDR register is indicated in the TDC[3:0] flags.

The SSIFSR register consists of flags that indicate the operating status of the SSIFTDR and SSIFRDR registers.

RDF flag (Receive Data Full Flag)

When the received data is transferred to the SSIFRDR register, the RDF flag indicates that the number of data bytes in the SSIFRDR register is equal to or greater than the receive FIFO threshold value, and so reading the received data from the SSIFRDR register is enabled.

[Setting condition]

- The number of receive data bytes stored in the SSIFRDR register is equal to or greater than the receive FIFO threshold value.

[Clearing conditions]

- 0 is written to the RDF flag after the RDF flag is confirmed to be 1
- Received data is read from the SSIFRDR register using a DMA or DTC transfer (transfer of the last block in block transfer). Do not clear the RDF flag to 0 during DMA or DTC transfer.

Note: Because the SSIFRDR register is a 32-byte FIFO register, the maximum number of data bytes that can be read from it while the RDF flag is 1 is indicated in the RDC[3:0] flags. If reading data from the SSIFRDR register continues after all the data is read, the values read as undefined.

RDC[3:0] flags (Receive Data Indicate Flag)

The RDC[3:0] flags indicate the number of data bytes stored in the SSIFRDR register.

RDC[3:0] flags = 0h indicates no received data. RDC[3:0] flags = 8h indicates that 32 bytes of received data is stored in the SSIFRDR register.

TDE flag (Transmit Data Empty Flag)

When data is transferred from the SSIFTDR register to the SSITDR register, the TDE flag indicates that the number of data bytes in the SSIFTDR register is less than the transmit FIFO threshold value, and so writing transmit data to the SSIFTDR register is enabled.

[Setting condition]

- The amount of transmit data written to the SSIFTDR register is equal to or less than the transmit FIFO threshold value.

[Clearing conditions]

- 0 is written to the TDE flag after the TDE flag is confirmed to be 1
- Transmit data is written to the SSIFTDR register using DMA or DTC transfer. Do not clear the TDE flag to 0 during DMA or DTC transfer.

Note: Because the SSIFTDR register is a 32-bit register with an 8-stage FIFO, the maximum number of bytes that can be written to it while the TDE flag is 1, is 8 - TDC[3:0]. If writing data to the SSIFTDR register continues after all the data is written, writes are invalid and an overflow occurs.

TDC[3:0] flags (Transmit Data Indicate Flag)

The TDC[3:0] flags indicate the amount of data stored in the SSIFTDR register.

TDC[3:0] flags = 0h indicates no data for transmission. TDC[3:0] flags = 8h indicates that 32 bytes of data for transmission is stored in the SSIFTDR register.

36.2.5 Transmit FIFO Data Register (SSIFTDR)

Address(es): SSI0.SSIFTDR 4004 E018h, SSI1.SSIFTDR 4004 E118h

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

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

The SSIFTDR register is a write-only FIFO register consisting of eight stages of 32-bit registers for storing data to be serially transmitted. Write transmit data to the SSIFTDR register in 64-bit (2-stage FIFO) units regardless of the data word length setting. If transmit data ends on a 32-bit boundary, write 32-bit 0-data (0000 0000h) after the last transmit data is written, and stop transmission when 64-bit are written. When the transmit shift register is empty, the SSI transfers the transmit data written to the SSIFTDR register to start serial transmission, which can be continued until the SSIFTDR register becomes empty.

When the SSIFTDR register is full of data (32 bytes), the next data cannot be written to it. Writes are ignored and an overflow occurs.

36.2.6 Receive FIFO Data Register (SSIFRDR)

Address(es): SSI0.SSIFRDR 4004 E01Ch, SSI1.SSIFRDR 4004 E11Ch

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

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

The SSIFRDR register is a read-only FIFO register consisting of eight stages of 32-bit registers for storing serially received data. Each time 4 bytes of serial data is received, the SSI stores the received serial data in the SSIFRDR register from the receive shift register according to the PDTA bit setting. Receive operation can be continued until a maximum 32 bytes of data are stored to the SSIFRDR register. The SSIFRDR register can be read but cannot be written to. If the SSIFRDR register is read when it contains no received data, undefined values are read and a receive underflow occurs.

When the SSIFRDR register is full of received data, any subsequent data received is lost and a receive overflow occurs.

36.2.7 TDM Mode Register (SSITDMR)

Address(es): SSI0.SSITDMR 4004 E020h, SSI1.SSITDMR 4004 E120h

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

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

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	CONT	WS Continue Mode*1	0: WS continue mode disabled 1: WS continue mode enabled.	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit can be set only in master mode (SSICR.SCKD bit = 1 and SSICR.SWSD bit = 1).

The SSITDMR register is a read/write 32-bit register that enables or disables WS continue mode.

36.3 Operation

36.3.1 Bus Format

The SSI can operate as a transmitter or receiver and can be configured into multiple serial bus formats in either mode, as shown in Table 36.4.

Table 36.4 Bus format

	TEN	REN	SCKD	SWSD	MUEN	IIEN	TOIEN	TUIEN	ROIEN	RUIEN	CONT	SWSP	DEL	PDTA	SDTA	SPDP	SCKP	SWL[2:0]	DWL[2:0]	CHNL[1:0]
Non-Compression Slave Receiver	0	1	0	0	Control Bits							Configuration Bits								
Non-Compression Slave Transmitter	1	0	0	0																
Non-Compression Slave Transceiver	1	1	0	0																
Non-Compression Master Receiver	0	1	1	1																
Non-Compression Master Transmitter	1	0	1	1																
Non-Compression Master Transceiver	1	1	1	1																

36.3.2 Non-compression Mode

The SSI only supports non-compression mode. It supports the SSI-compatible format in addition to MSB-first order and left and right alignment.

(1) Slave Receiver

This mode allows the SSI to receive serial data from another device. The clock and word select signals used for the serial data stream are also supplied from an external device. Operation is not guaranteed if these signals do not conform to the format specified in the configuration fields of the SSI.

(2) Slave Transmitter

This mode allows the SSI to transmit serial data to another device. The clock and word select signals used for the serial data stream are also supplied from an external device. Operation is not guaranteed if these signals do not conform to the format specified in the configuration fields of the SSI.

(3) Slave Transceiver

This mode allows serial data transmission and reception between the SSI and another device. The clock and word select signals used for the serial data stream are also supplied from an external device. Operation is not guaranteed if these signals do not conform to the format specified in the configuration fields of the SSI.

(4) Master Receiver

This mode allows the SSI to receive serial data from another device. The clock and word select signals are internally derived from the master clock. The format of these signals is defined in the configuration fields of the SSI. Operation is not guaranteed if the incoming data does not follow the configured format.

(5) Master Transmitter

This mode allows the SSI to transmit serial data to another device. The clock and word select signals are internally derived from the master clock. The format of these signals is defined in the configuration fields of the SSI.

(6) Master Transceiver

This mode allows serial data transmission and reception between the SSI and another device. The clock and word select signals are internally derived from the master clock. The format of these signals is defined in the configuration fields of the SSI.

(7) Operating Setting for Word Length

All bits related to the SSICR register word length are valid in non-compression modes. The SSI supports multiple configurations. Figure 36.3 and Figure 36.4 show some of the combinations for the SSI-compatible format, MSB-first and left-aligned format, and MSB-first and right-aligned format.

SSI compatible format

Figure 36.3 and Figure 36.4 show the SSI compatible format both without and with padding. Padding occurs when the data word length is smaller than the system word length.

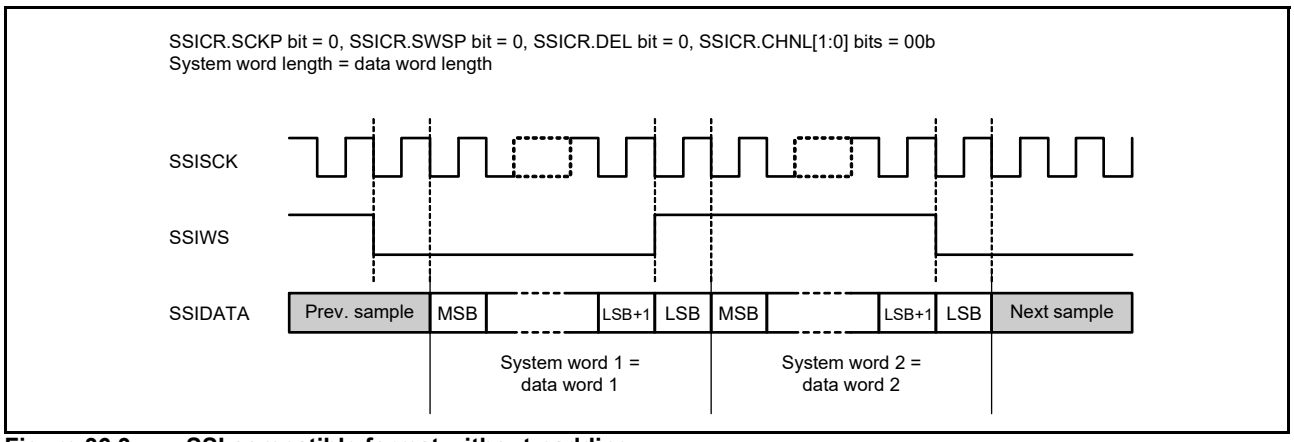


Figure 36.3 SSI compatible format without padding

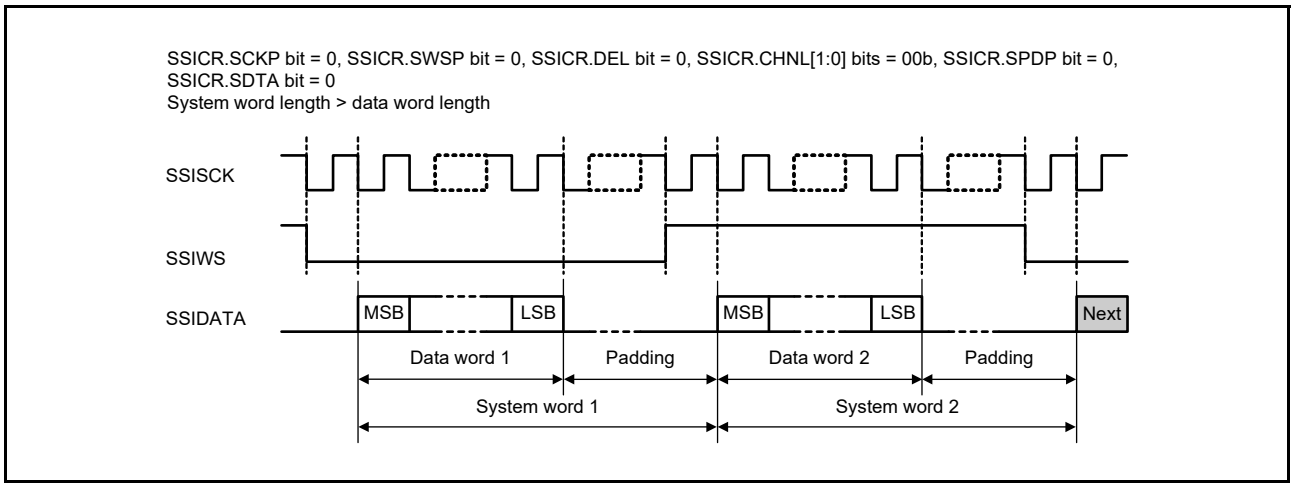


Figure 36.4 SSI compatible format with padding

Figure 36.5 shows the MSB-first and left-aligned format and Figure 36.6 shows the MSB-first and right-aligned format.

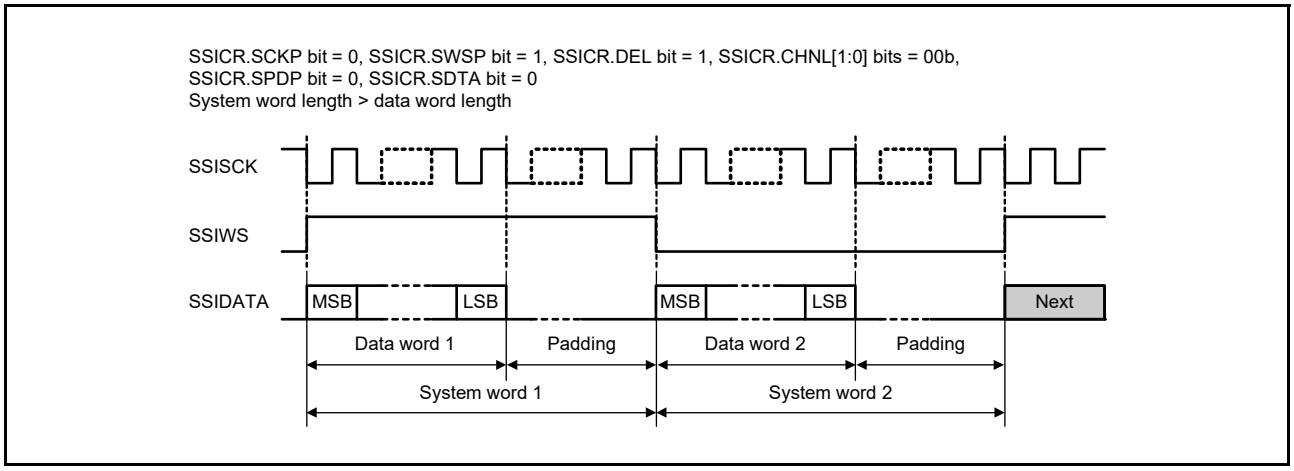


Figure 36.5 MSB-first and left-aligned format transmitted and received in the order of serial data and padding bits

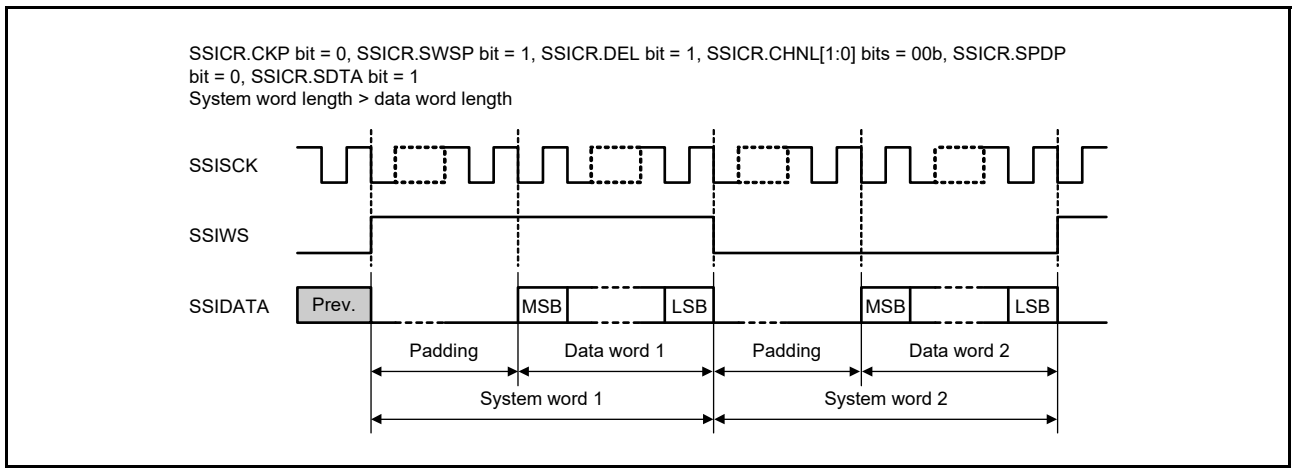


Figure 36.6 MSB-first and right-aligned format, transmitted and received in the order of padding bits and serial data

Table 36.5 shows the number of padding bits for each of the valid setting.

Table 36.5 Number of padding bits for each valid setting

Padding bits per system word			SSICR.DWL[2:0] bits	000b	001b	010b	011b	100b	101b
SSICR.CHNL[1:0] bits	Decoded channels per system word	SSICR.SWL[2:0] bits	Data word length	8	16	18	20	22	24
			System word length						
00b	1	000b	8	0	-	-	-	-	-
		001b	16	8	0	-	-	-	-
		010b	24	16	8	6	4	2	0
		011b	32	24	16	14	12	10	8

(8) Operating Settings Other Than for Word Length

This section describes several more bits used to configure non-compression mode. These bits are not mutually exclusive and some combinations might not be useful for any other device. These configuration bits are also described in this section with reference to the basic format example in Figure 36.7.

In Figure 36.7 to Figure 36.15, a system word length of 6 bits and a data word length of 4 bits are used for simplification.

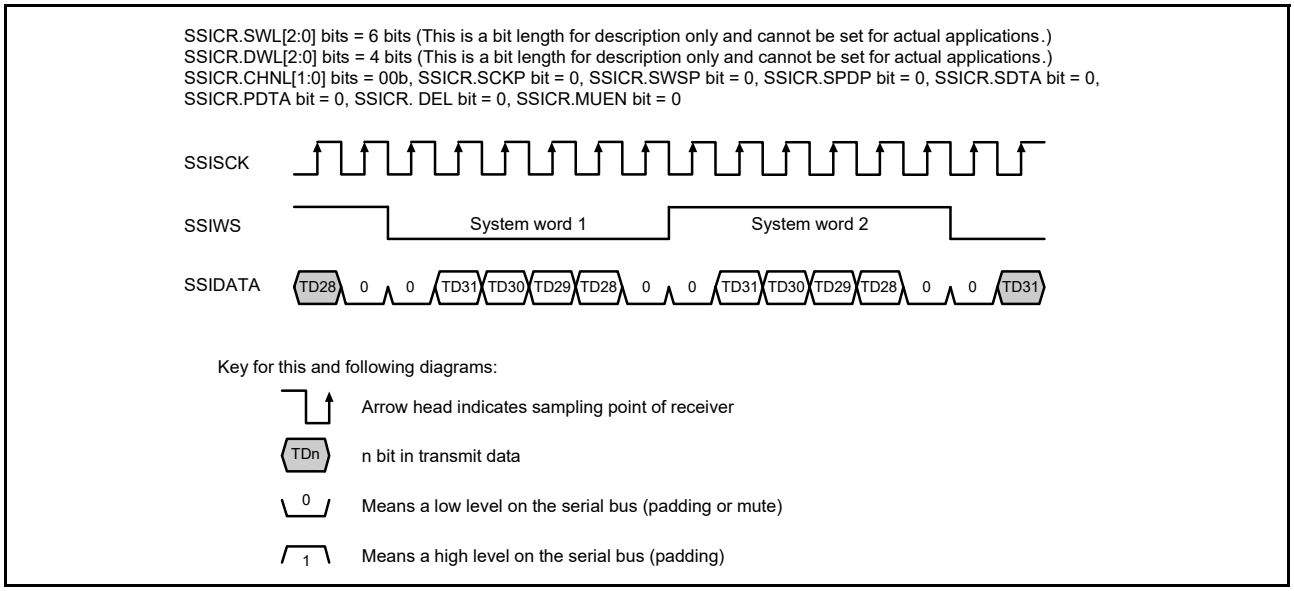


Figure 36.7 Basic format example in transmit mode

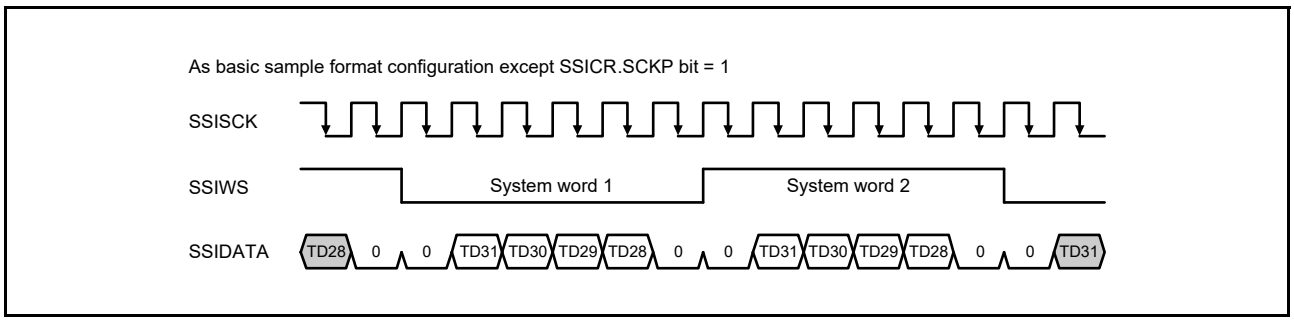


Figure 36.8 Inverted clock

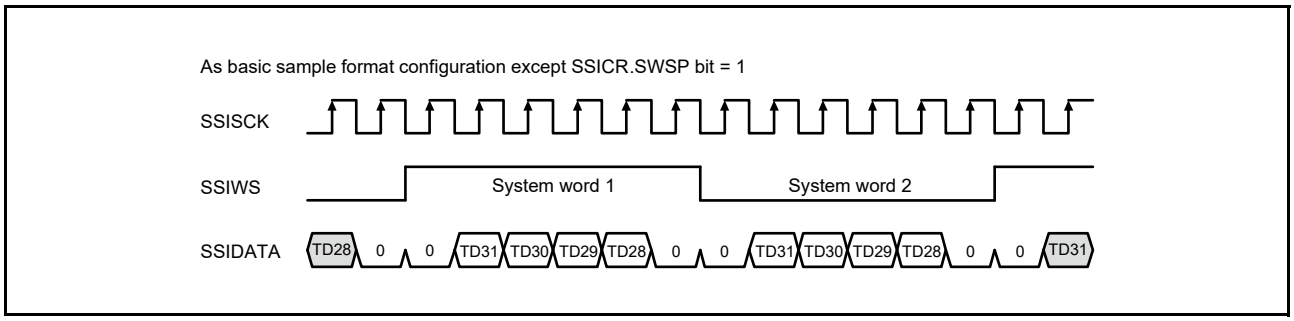


Figure 36.9 Inverted word select

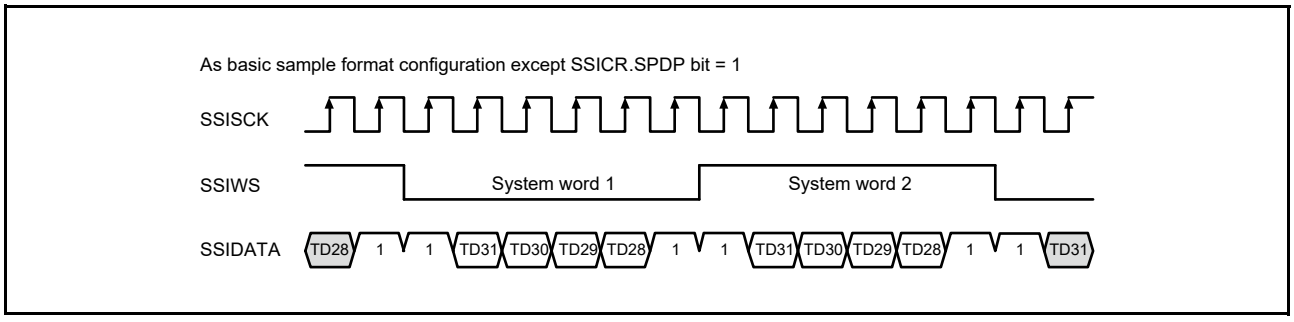


Figure 36.10 Inverted padding polarity

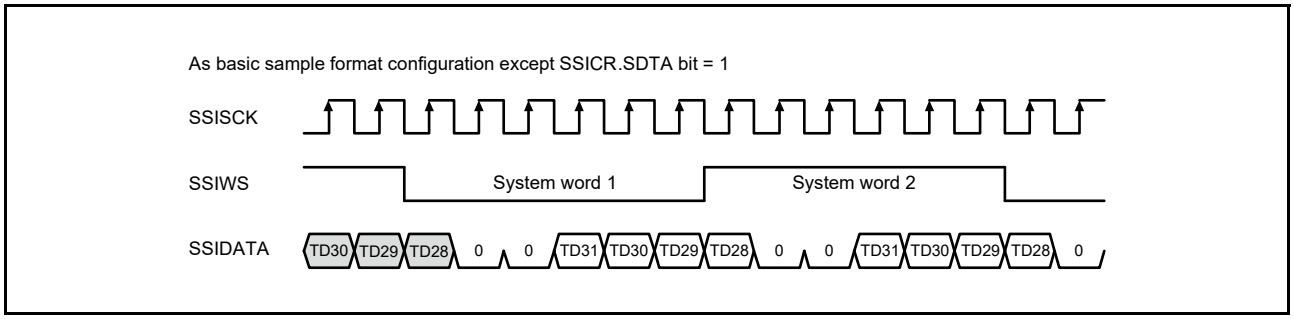


Figure 36.11 Basic format example with transmit and receive in the order from padding bits to serial data, with delay

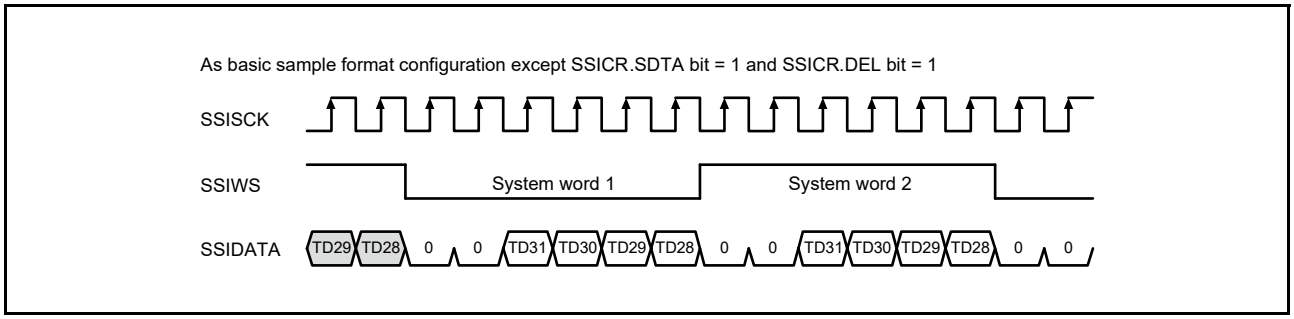


Figure 36.12 Basic format example with transmit and receive in the order from padding bits to serial data, without delay

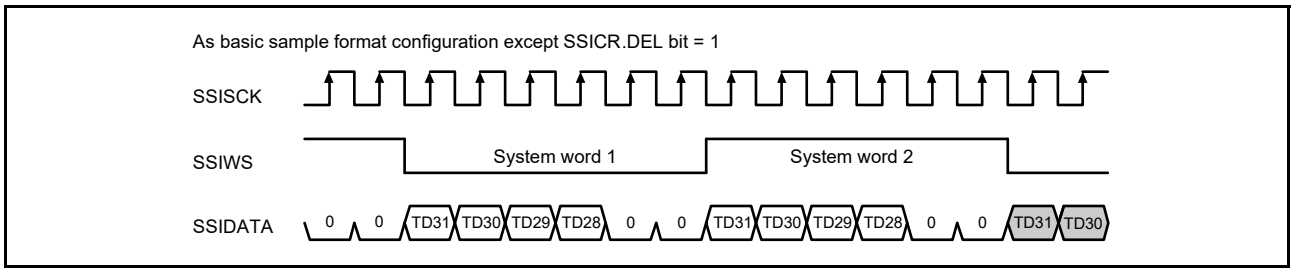


Figure 36.13 Basic format example with transmit and receive in the order from serial data to padding bits, without delay

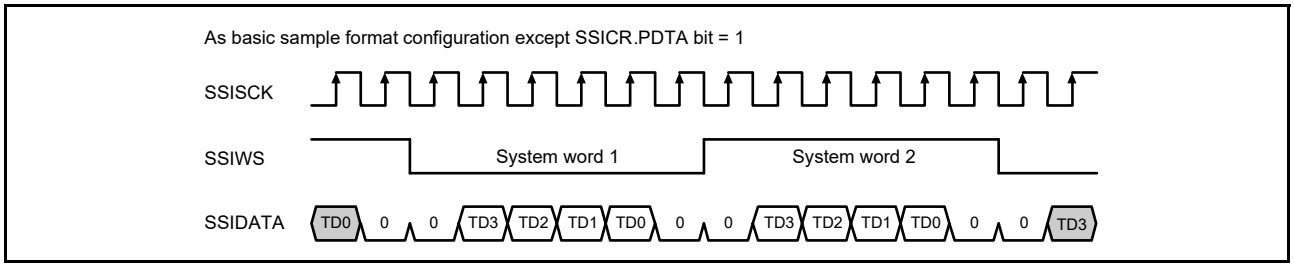


Figure 36.14 Basic format example with parallel right alignment, with delay

Mute enabled

When the SSICR.MUEN bit is set, the SSITXD0 and SSIDATA1 pins are set to 0 without synchronizing SSIWS.

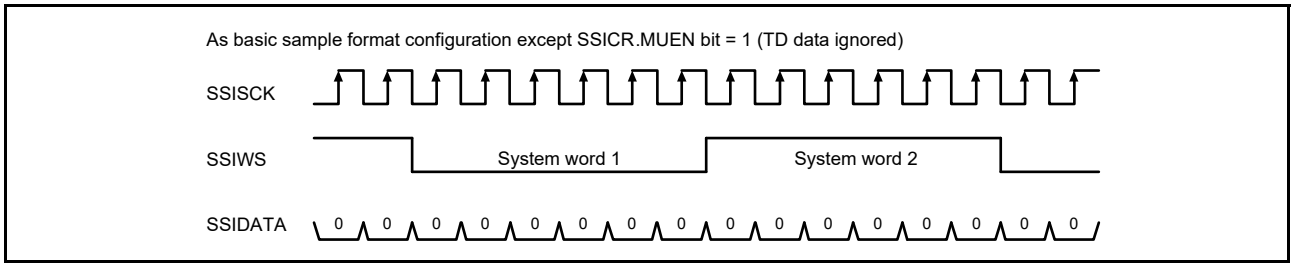


Figure 36.15 Basic format example with mute enabled

36.3.3 WS Continue Mode

In WS continue mode, the SSIWS signal continues to be output whether data transfer is enabled or disabled. This mode can be set using the SSITDMR.CONT bit. With this mode enabled, the SSIWS signal does not stop but continues operating even if the SSICR.TEN and REN bits are both set to 0 (transfer disabled). With this mode disabled, the SSIWS signal stops if the SSICR.TEN and REN bits are both set to 0.

Figure 36.16 and Figure 36.17 show the operations with WS continue mode enabled and disabled, respectively.

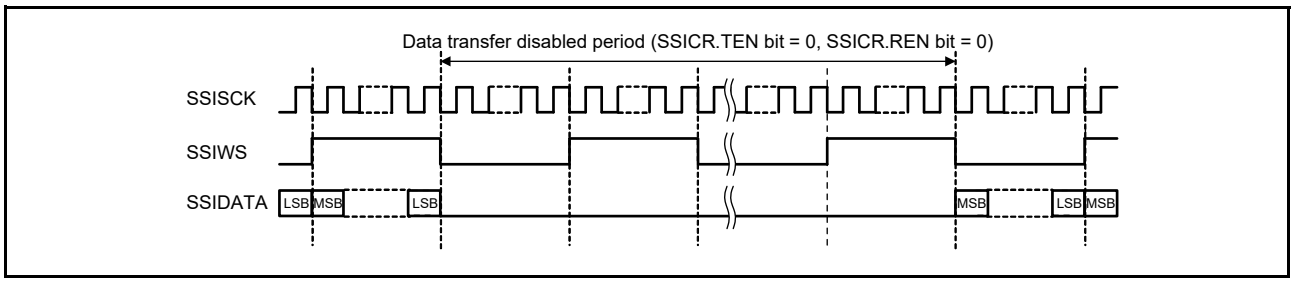


Figure 36.16 SSI operation with WS continue mode enabled

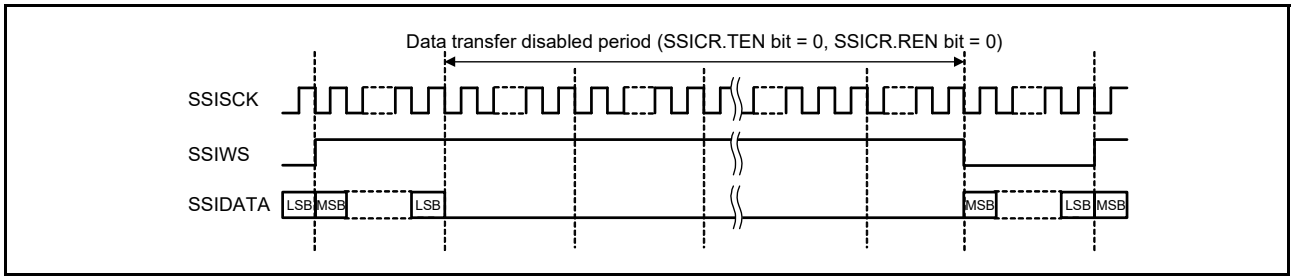


Figure 36.17 SSI operation with WS continue mode disabled

36.3.4 Operating States

The SSI has three operating states:

- Idle
- Communication
- Waiting for idle.

Figure 36.18 shows the operating state transitions.

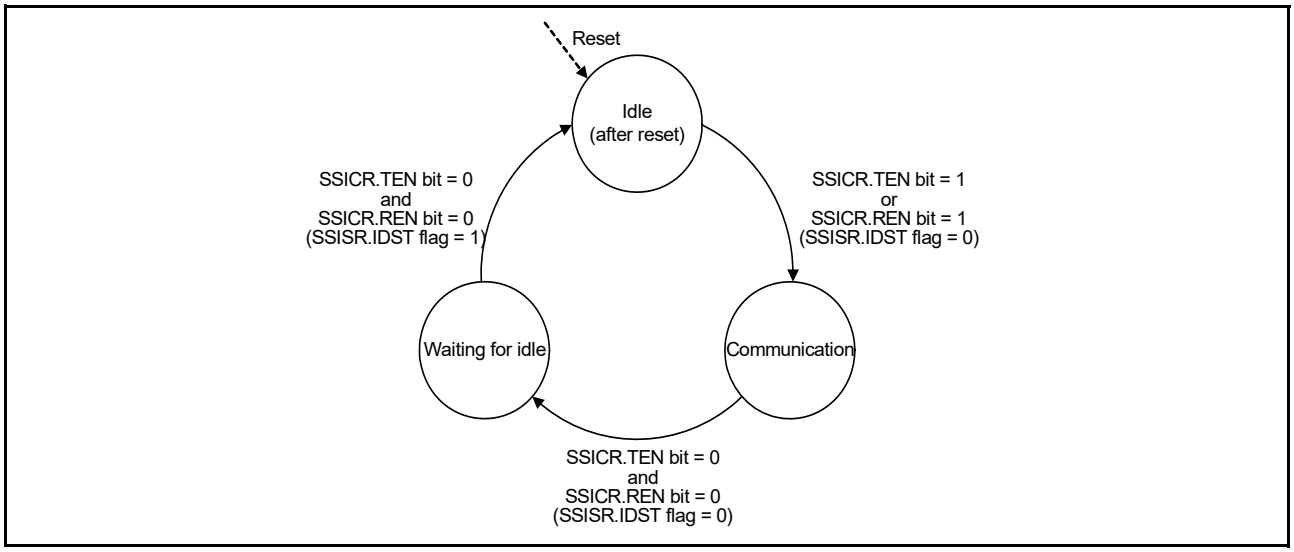


Figure 36.18 Operating state transitions

(1) Idle State

The SSI enters this state when the MSTPCRC.MSTPC7 and MSTPC8 bits are set to 0 on release of a reset. The software defines all required configuration fields in the control registers in this state. After the settings are made, the SSI enters the communication state when the SSICR.TEN bit or SSICR.REN bit is set to 1.

(2) Communication State

Communication in this state depends on the selected operating state. For details, see section 36.3.5, Transmit Operation and section 36.3.6, Receive Operation.

(3) Waiting for Idle

The SSI enters this state when both the SSICR.TEN and SSICR.REN bits are set to 0 in communication state. If system word communication is completed in this state, the SSISR.IDST flag is set to 1 and the SSI enters the idle state.

36.3.5 Transmit Operation

Transmission can be controlled either by a DMA or DTC transfer or by interrupts.

Renesas recommends using DMAC or DTC control to reduce the processor load. In transmission using the DMAC or DTC, the processor receives interrupts only if there is an underflow or overflow of data or if the DMA or DTC transfer is complete. In transmission using DMA or DTC transfer, set the number of DMA or DTC transfers to multiples of 2 to write transmit data to the SSIFTDR register in 64-bit (2-stage FIFO) units.

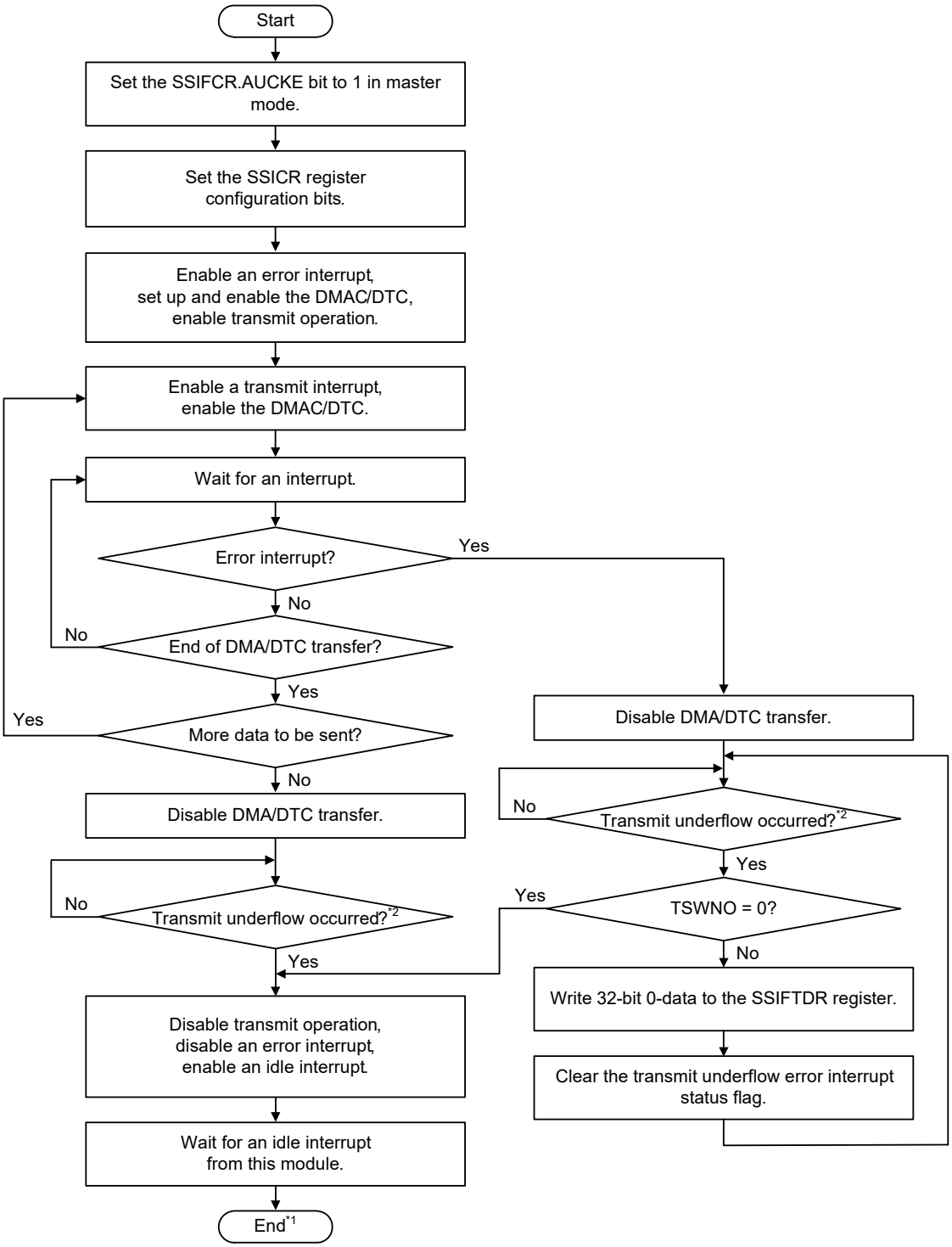
The alternative method is using the interrupts that the SSI generates to supply data as required. In transmission using interrupts, write transmit data in 64-bit units regardless of the data format. If the transmit data ends on a 32-bit boundary, write 32-bit 0-data (0000 0000h) after the last transmit data is written, and complete writing on a 64-bit boundary.

When stopping transmission, stop writing to the SSIFTDR register when 64-bit writing is complete. After writing is stopped, wait until a transmit underflow occurs before setting the TEN bit to 0. During transmit underflow, the last data input to SSIFTDR is continuously transmitted until the SSI enters the idle state. After the TEN bit is set to 0, the clock*1 must continue to be supplied until the SSISR.IIRQ flag indicates that the module is in the idle state. If a transmit underflow error or transmit overflow error occurs during data transmission, transmit data might not be written to SSIFTDR in 64-bit units. In this case, stop writing data, wait until a transmit underflow error occurs, then check the TSWNO. When the TSWNO bit is 1, write 32-bit 0-data (0000 0000h) to SSIFTDR and wait until an underflow occurs again. After the TSWNO bit is confirmed to be 0, and after the TEN bit is set to 0, the clock*1 must continue to be supplied until the SSISR.IIRQ flag indicates that the SSI is in the idle state.

Note 1. Input clock from the SSISCK pin when SSICR.SCKD bit = 0. For the master clock, when SSICR.SCKD bit = 1.

Figure 36.19 shows transmission using the DMA or DTC and Figure 36.20 shows transmission using interrupts.

(1) Transmission Using the DMAC or DTC

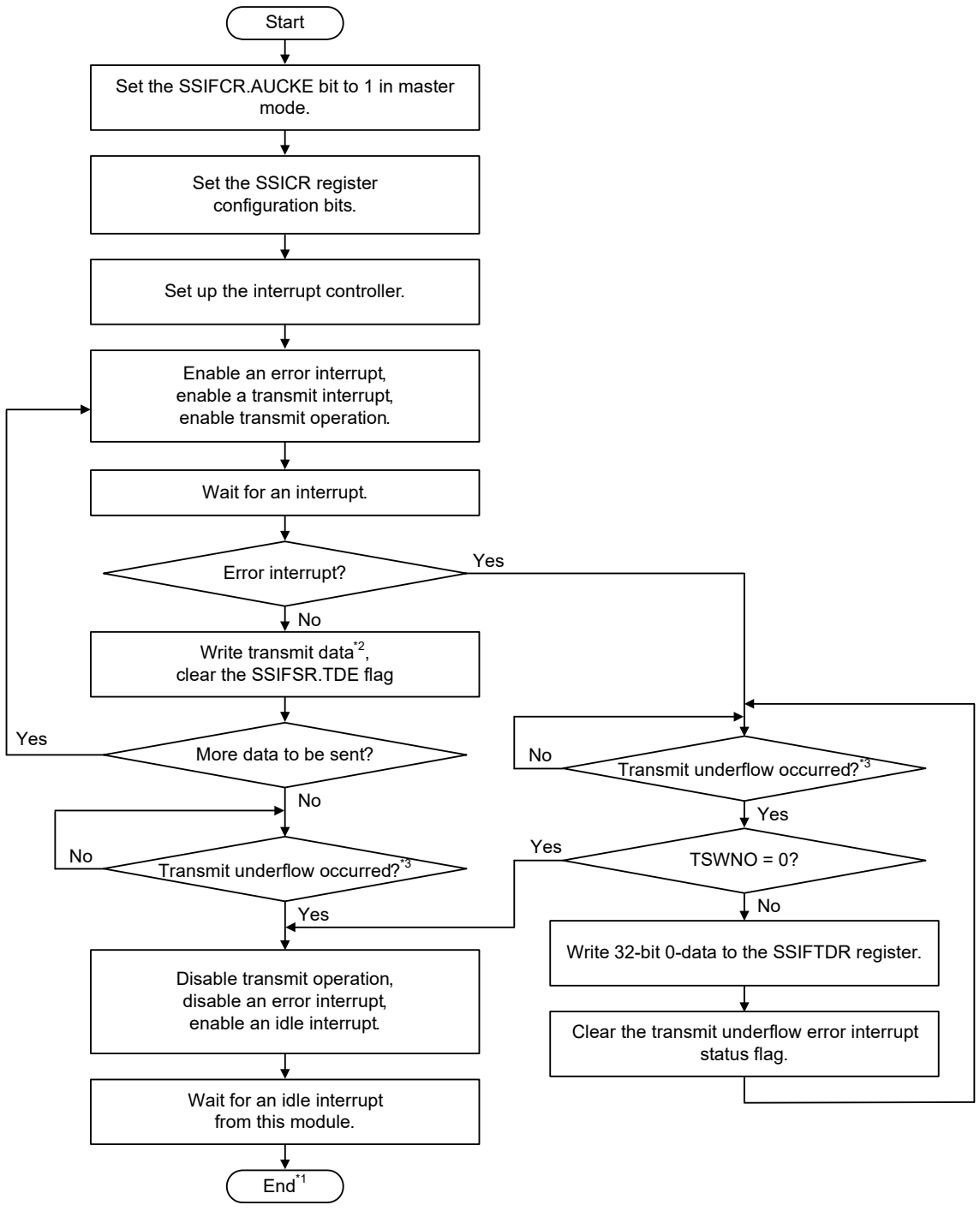


Note 1. When restarting transmission after transmit operation is disabled (TEN = 0) while WS continue mode is disabled, execute a software reset and go back to the start of the flow.

Note 2. Stop writing transmit data to the SSIFTDR register and wait until a transmit underflow occurs.On transmit underflow, the last data input to SSIFTDR is continuously transmitted until the SSI is in the idle state.

Figure 36.19 Transmission flow using the DMAC or DTC

(2) Transmission Using Interrupts



Note 1. When restarting transmission after transmit operation is disabled (TEN = 0) while WS continue mode is disabled, execute a software reset and go back to the start of the flow.
Note 2. Write transmit data in 64-bit units regardless of the data word setting.
Note 3. Stop writing transmit data to the SSIFTDR register and wait until a transmit underflow occurs.On transmit underflow, the last data input to SSIFTDR is continuously transmitted until the SSI is in the idle state.

Figure 36.20 Transmission flow using interrupts

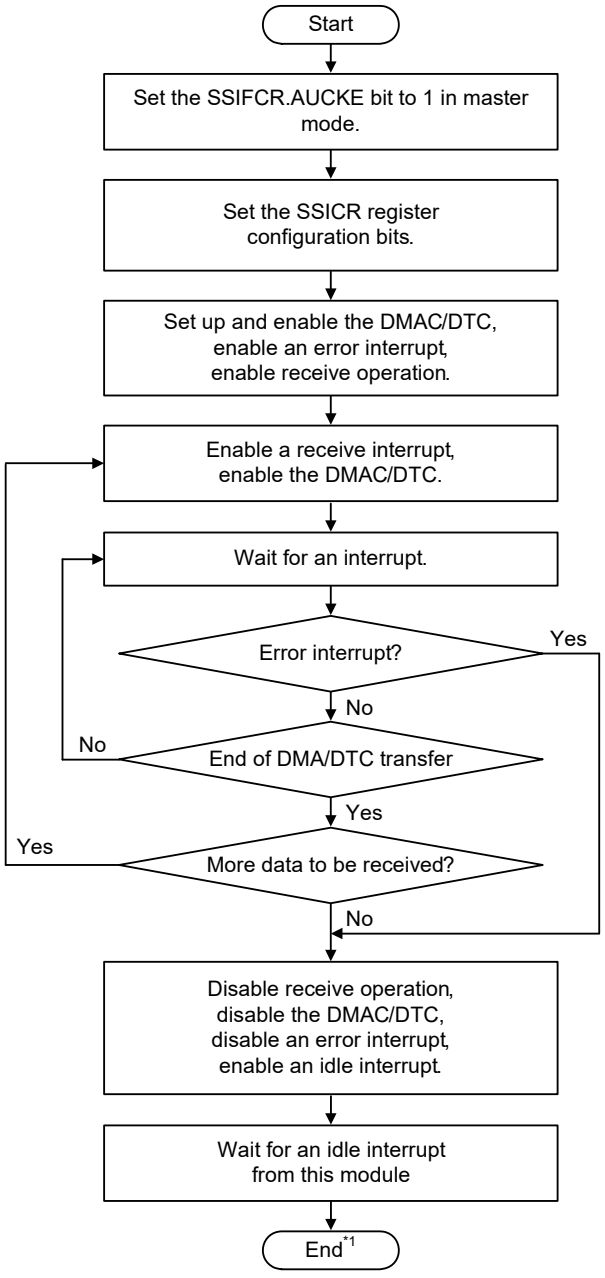
36.3.6 Receive Operation

As with transmission, reception can be controlled either by a DMA or DTC transfer or interrupt. Figure 36.21 and Figure 36.22 show the flow of operation.

When stopping reception, the clock*1 supply must continue from when the REN bit is set to 0 until the SSISR.IIRQ flag indicates that the SSI is in the idle state.

Note 1. Input clock from the SSISCK pin when SSICR.SCKD bit = 0. For the master clock, when SSICR.SCKD bit = 1.

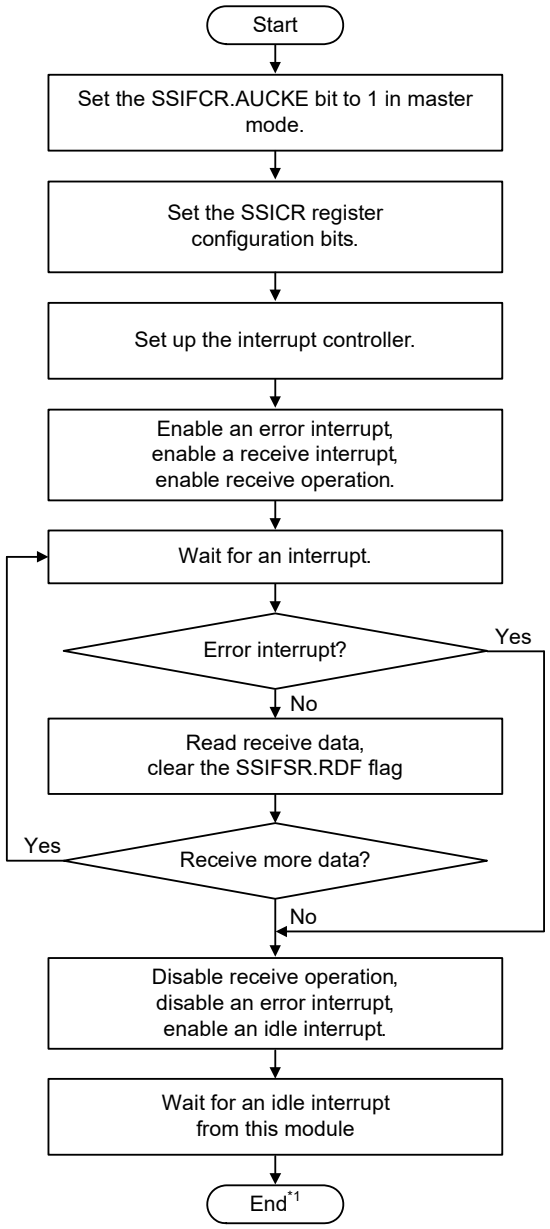
(1) Reception Using the DMAC or DTC



Note 1. If an error interrupt (underflow or overflow) occurs, go back to the start of the flow.

Figure 36.21 Reception flow using the DMAC or DTC

(2) Reception Using Interrupt-Driven Data Flow Control



Note 1. If an error interrupt (underflow or overflow) occurs, go back to the start of the flow.

Figure 36.22 Reception flow using interrupts

36.3.7 Serial Bit Clock Control

The SSI controls and selects the clock to be used for the serial bus interface using the SCKD and CKDV bit settings. If the serial bit clock direction is set to input (SSICR.SCKD bit = 0), the SSI is in clock slave mode and the shift register uses the bit clock that was input to the SSISCK pin. If the serial bit clock direction is set to output (SSICR.SCKD bit = 1), the SSI is in clock master mode, and the shift register uses the master clock or a divided master clock as the bit clock. The master clock is divided by the ratio specified by the SSICR.CKDV[3:0] bits for use as the bit clock by the shift register. In either case, the SSISCK pin, is the same as the bit clock.

36.4 Interrupt Sources

Table 36.6 lists the interrupt sources of the SSI. Each interrupt source can be enabled or disabled by the SSICR.TUIEN, TOIEN, RUIEN, ROIEN, and IIEN bits, and the SSIFCR.TIE and RIE bits.

Table 36.6 SSI interrupt sources

Channel	Interrupt source	Description	Interrupt flag	DMAC/DTC activation
SSI0	SSI0_SSIF	Transmit underflow interrupt, Transmit overflow interrupt, Receive underflow interrupt, Receive overflow interrupt, Idle interrupt	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIRQ SSISR.IIRQ	Not possible
	SSI0_SSIRXI	Receive data full interrupt	SSIFSR.RDF	Possible
	SSI0_SSITXI	Transmit data empty interrupt	SSIFSR.TDE	Possible
SSI1	SSI1_SSIF	Transmit underflow interrupt, Transmit overflow interrupt, Receive underflow interrupt, Receive overflow interrupt, Idle interrupt	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIRQ SSISR.IIRQ	Not possible
	SSI1_SSIRT	Receive data full interrupt, Transmit data empty interrupt	SSIFSR.RDF/ SSIFSR.TDE	Possible

36.5 Usage Notes

36.5.1 Setting for the Module-Stop State

SSI operation can be enabled or disabled using the Module Stop Control Register C (MSTPCRC). The SSI is initially stop after reset. Register access is enabled by releasing the module-stop state. For details on the MSTPCRC register, see section 11, Low Power Modes.

36.5.2 Notes on Changing Transfer Modes

For mode transitions between the transmitter, receiver, and transceiver while the WS continue mode is disabled (SSITDMR.CONT = 0), set the SSICR.TEN and SSICR.REN bits to 0 and transition to the idle state one time. Set the SSICR.TEN and SSICR.REN bits again while the SSI is in the idle state and restart transfer.

36.5.3 Constraints on the WS Continue Mode

If the WS continue mode setting is changed, the operation of the SSISCK and SSIWS signals immediately after switching is not guaranteed. If this affects the device to be connected, do not change the setting dynamically.

37. SD/MMC Host Interface (SDHI)

37.1 Overview

The Secure Digital Host Interface (SDHI) and Multi-Media Card (MMC) interface provide the functionality needed to connect a variety of external memory cards with the MCU. The SDHI supports both 1-bit and 4-bit buses for connecting different memory cards that support SD, SDHC, and SDXC formats. When developing host devices that are compliant with the SD Specifications, you must comply with the SD Host/Ancillary Product License Agreement (SD HALA).

The MMC interface supports 1-bit, 4-bit, and 8-bit MMC buses that provide eMMC 4.51 (JEDEC Standard JESD 84-B451) device access. The MMC interface also provides backward compatibility and support for high-speed SDR transfer modes. Table 37.1 lists the SD/MMC host interface specifications and Figure 37.1 shows the block diagram.

Table 37.1 SD/MMC host interface specifications

Interface	Parameter	Description
SD	SD bus interface	• Compatible with SD memory card and SDIO card • Transfer bus mode selectable from 4-bit wide bus mode or 1-bit default bus mode • Compatible with SD, SDHC, and SDXC formats.
SD/MMC common	SDHI clock frequency	The SDHI clock is generated by dividing PCLKA by 2^n (n = 1 to 9)
	Error check functions	CRC7 (command/response), CRC16 (transfer data)
	Interrupt sources	• Card access interrupt (SDHI_MMC0_ACCS) • SDIO access interrupt (SDHI_MMC0_SDIO) • Card detection interrupt (SDHI_MMC0_CARD).
	DMA transfer sources	DMAC and DTC triggerable by the SBFAI interrupt SD buffer is read and write accessible using the DMAC
	Other functions	• Card detect function • Write protect support.
MMC	MMC bus interface	Transfer bus mode selectable from 1-bit, 4-bit, or 8-bit
	Transfer modes	Backward compatible mode or high-speed SDR mode selectable
	Other functions	eMMC device access supported

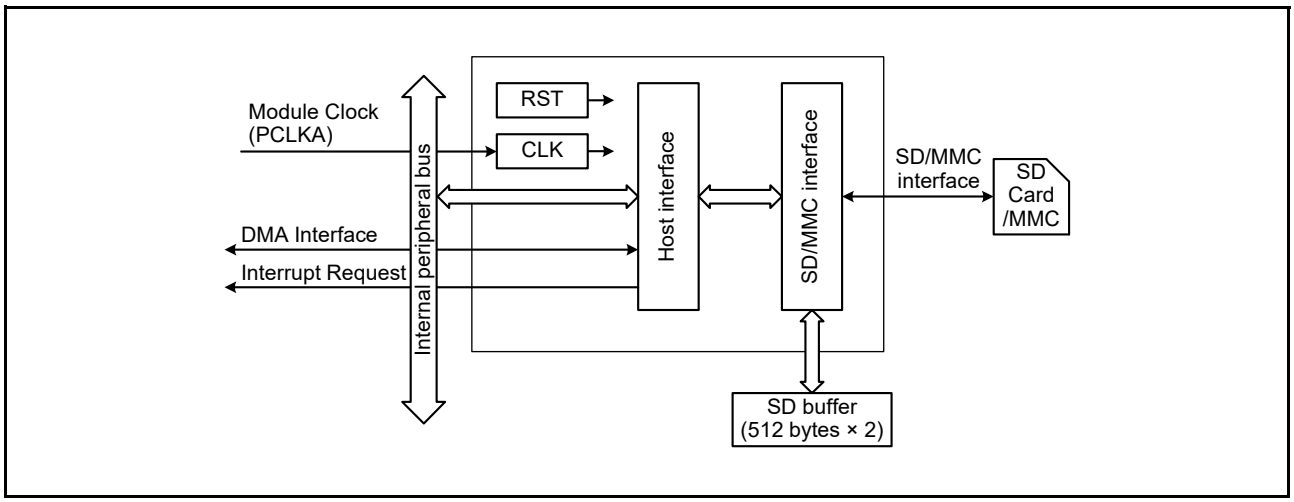


Figure 37.1 SD/MMC host interface block diagram

Table 37.2 Pin configuration of SDHI

Channel	Pin name	I/O	Description
Ch 0	SD0CLK	Output	SDHI clock
	SD0CMD	I/O	Command output, response input
	SD0DAT0	I/O	Data 0 (DAT0)
	SD0DAT1	I/O	Data 1 (DAT1), SDIO interrupt
	SD0DAT2	I/O	Data 2 (DAT2), SDIO Read wait
	SD0DAT3	I/O	Data 3 (DAT3), SD Card detect
	SD0DAT4	I/O	MMC Data 4 (DAT4)
	SD0DAT5	I/O	MMC Data 5 (DAT5)
	SD0DAT6	I/O	MMC Data 6 (DAT6)
	SD0DAT7	I/O	MMC Data 7 (DAT7)
	SD0WP	Input	SD card write protection

37.2 Register Descriptions

37.2.1 Command Type Register (SD_CMD)

Address(es): SDHI0.SD_CMD 4006 2000h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMD12AT[1:0]		TRSTP	CMDRW	CMDTP	RSPTP[2:0]			ACMD[1:0]		CMDIDX[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	CMDIDX[5:0]	Command Index Field Value Select	These bits configure the command index field value. The following examples include the bit values for the ACMD[1:0] bits: b7 b0 0 0 0 0 0 1 1 0: CMD6 0 0 0 1 0 0 1 0: CMD18 0 1 0 0 1 1 0 1: ACMD13.	R/W
b7, b6	ACMD[1:0]	Command Type Select	b7 b6 0 0: CMD 0 1: ACMD. Other settings are prohibited.	R/W
b10 to b8	RSPTP[2:0]	Response Type Select*1	b10 b8 0 0 0: Normal Mode. Depending on the command, the response type and transfer method are selected by setting the ACMD[1:0] bits and CMDIDX[5:0] bits. The values of b15 to b11 in this register are invalid. 0 1 1: Extended mode and no response 1 0 0: Extended mode and R1, R5, R6, or R7 response 1 0 1: Extended mode and R1b response 1 1 0: Extended mode and R2 response 1 1 1: Extended mode and R3 or R4 response. Other settings are prohibited.	R/W
b11	CMDTP	Data Transfer Select*2	0: Command does not include data transfer (bc, bcr, or ac) 1: Command includes data transfer (adtc).	R/W
b12	CMDRW	Data Transfer Direction Select*3	0: Write (SD/MMC host interface → SD card/MMC) 1: Read (SD/MMC host interface ← SD card/MMC).	R/W
b13	TRSTP	Block Transfer Select*3	0: Single block transfer 1: Multiple block transfer.	R/W
b15, b14	CMD12AT[1:0]	CMD12 Automatic Issue Select*4	b15 b14 0 0: CMD12 is automatically issued during multi-block transfer 0 1: CMD12 is not automatically issued during multi-block transfer. Other settings are prohibited.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Some commands cannot be used in normal mode. See Table 37.3 and set the RSPTP[2:0] bits.
Note 2. The CMDTP bit is valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b.
Note 3. The CMDRW and TRSTP bits are valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the CMDTP bit is 1.
Note 4. The CMD12AT[1:0] bits are valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the TRSTP bit is 1.

The command type and response type are set in the SD_CMD register. The command type and transfer mode must be set when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b. The sequence starts when a value is written to this register. See Table 37.8 and Table 37.9 for setting examples. Do not write to the SD_CMD register when the CBSY flag in the SD_INFO2 register is 1.

37.2.2 SD Command Argument Register (SD_ARG)

Address(es): SDHI0.SD_ARG 4006 2008h

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

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

Bit	Symbol	Bit name	Description	R/W
b31 to b0	—	—	Set command format[39:8] (argument)	R/W

The SD_ARG register is used for setting the argument field value. Set the SD_ARG register before setting the SD_CMD register. The automatically issued CMD12 has an argument field of 0000_0000h regardless of the SD_ARG register value.

37.2.3 SD Command Argument Register 1 (SD_ARG1)

Address(es): SDHI0.SD_ARG1 4006 200Ch

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

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

Bit	Symbol	Bit name	Description	R/W
b15 to b0	—	—	Set command format[39:24] (argument)	R/W
b31 to b16	—	Reserved	These bits are read as 0	R

The SD_ARG1 register is used for setting the argument field value. Set the SD_ARG1 register before setting the SD_CMD register. The argument field value of the automatically issued CMD12 is 0000_0000h regardless of the SD_ARG1 register value.

37.2.4 Data Stop Register (SD_STOP)

Address(es): SDHI0.SD_STOP 4006 2010h

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

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

Bit	Symbol	Bit name	Description	R/W
b0	STP	Transfer Stop	Data transfer stops when this bit is set to 1	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	SEC	Block Count Register Value Select *1	0: SD_SECCNT register value is invalid 1: SD_SECCNT register value is valid.	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not rewrite this bit when the CBSY flag in the SD_INFO2 register is 1.

The SD_STOP register stops data transfer. During a multi-block transfer sequence, the SD_SECCNT register value (number of blocks to be transferred) can be set to valid or invalid by setting the SD_STOP register.

STP bit (Transfer Stop)

- When STP is set to 1 during a multi-block transfer, CMD12 is issued to halt the transfer through the SD host interface. However, if a command sequence is halted because of a communication error or timeout, CMD12 is not issued. Although continued buffer access is possible even after STP is set to 1, the buffer access error bit (ILR or ILW) in the SD_INFO2 register is set accordingly.
- When STP is set to 1 during transfer for a single block write, the access end flag is set when SD_BUF becomes empty, and CMD12 is not issued. If SD_BUF does contain data, the access end flag is set on completion of reception of the busy state without CMD12 being issued.
- When STP is set to 1 during transfer for a single block read, the access end flag is set immediately after setting of the STP bit and CMD12 is not issued.
- When STP is set to 1 during reception of a busy state after an R1b response, the access end flag is set on completion of reception of the busy state without CMD12 being issued.
- When STP is set to 1 after a command sequence is complete, CMD12 is not issued and the access end flag is not set.
- Set STP to 1 after the response end flag sets.
- Set STP to 0 after the access end flag sets.

SEC bit (Block Count Register Value Select)

When the SD_CMD register is set to start the command sequence while SEC is set to 1, CMD12 is automatically issued to stop a multi-block transfer with the number of blocks set in the SD_SECCNT register.

SD_CMD[10:8] = 000b for CMD18 or CMD25 in Normal Mode

SD_CMD[15:13] = 001b in extended mode (CMD12 is automatically issued, multiple block transfer)

When the command sequence is halted because of a communication error or timeout, CMD12 is not automatically issued.

37.2.5 Block Count Register (SD_SECCNT)

Address(es): SDHI0.SD_SECCNT 4006 2014h

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

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

The SD_SECCNT is a read/write register that sets the number of blocks to be transferred when performing a multi-block transfer. For example, when the register value is 0000_0001h, 1 block is transferred. When the register value is 0000_FFFFh, 65,535 blocks are transferred and when the register value is FFFF_FFFFh, 4,294,967,295 blocks are transferred.

Do not set this register to 0000_0000h. Do not rewrite the SD_SECCNT register when the SD_INFO2.CBSY flag is 1.

37.2.6 SD Card Response Register 10 (SD_RSP10), SD Card Response Register 32 (SD_RSP32), SD Card Response Register 54 (SD_RSP54)

Address(es): SDHI0.SD_RSP10 4006 2018h, SDHI0.SD_RSP32 4006 2020h, SDHI0.SD_RSP54 4006 2028h

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

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

37.2.7 SD Card Response Register 1 (SD_RSP1), SD Card Response Register 3 (SD_RSP3), SD Card Response Register 5 (SD_RSP5)

Address(es): SDHI0.SD_RSP1 4006 201Ch, SDHI0.SD_RSP3 4006 2024h, SDHI0.SD_RSP5 4006 202Ch

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

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

Bit	Symbol	Bit name	Description	R/W
b15 to b0	—	—	This register stores the response from the SD card/MMC	R
b31 to b16	—	Reserved	These bits are read as 0	R

37.2.8 SD Card Response Register 76 (SD_RSP76)

Address(es): SDHI0.SD_RSP76 4006 2030h

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

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

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	This register stores the response from the SD card/MMC	R
b31 to b24	—	Reserved	These bits are read as 0	R

37.2.9 SD Card Response Register 7 (SD_RSP7)

Address(es): SDHI0.SD_RSP7 4006 2034h

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

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

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	—	This register stores the response from the SD card/MMC	R
b31 to b8	—	Reserved	These bits are read as 0	R

The registers SD_RSP10, SD_RSP32, SD_RSP54, SD_RSP1, SD_RSP3, SD_RSP5, SD_RSP76, and SD_RSP7 are read-only registers that store the response from the SD card/MMC. Depending on the type of response from the SD card/MMC, the SD/MMC host interface divides and stores the response among the four registers. Table 37.3 lists the correspondence between the response type and its storage destination.

Table 37.3 Correspondence between response type and storage destination

Response type	SD_RSP10 register	SD_RSP32 register	SD_RSP54 register	SD_RSP1 register	SD_RSP3 register	SD_RSP5 register	SD_RSP76 register	SD_RSP7 register
R1	[39:8]	-	[39:8]*1	-	-	-	-	-
R1b	[39:8]	-	[39:8]*1	-	-	-	-	-
R2	[39:8]	[71:40]	[103:72]	-	-	-	[127:104]	-
R3	[39:8]	-	-	-	-	-	-	-
R4	[39:8]	-	-	-	-	-	-	-
R5	[39:8]	-	-	-	-	-	-	-
R6	[39:8]	-	-	-	-	-	-	-
R7	[39:8]	-	-	-	-	-	-	-

Note 1. The responses for CMD18 and CMD25 are stored in the registers SD_RSP10 and SD_RSP54. Therefore, even if the SD_RSP10 register is overwritten with the response for the automatically issued CMD12, the response for CMD18 or CMD25 can be confirmed by reading the SD_RSP54 register.

37.2.10 SD Card Interrupt Flag Register 1 (SD_INFO1)

Address(es): SDHI0.SD_INFO1 4006 2038h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SDD3MON	SDD3IN	SDD3RM	SDWPMON	—	—	—	—	ACEND	—	RSPEND
Value after reset:	0	0	0	0	0	x	0	0	x	0	1	0	0	0*1	0	0*1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	RSPEND	Response End Detection Flag	0: Response end is not detected 1: Response end is detected.	R(/W)*2
b1	—	Reserved	This bit is read as 0	R
b2	ACEND	Access End Detection Flag	0: Access end is not detected 1: Access end is detected.	R(/W)*2
b4, b3	—	Reserved	These bits are read as 0	R
b5	—	Reserved	This bit is read as 1	R
b6	—	Reserved	This bit is read as 0	R
b7	SDWPMON	SD0WP Pin Monitor Flag	0: SD0WP pin level is high 1: SD0WP pin level is low.	R
b8	SDD3RM	SD0DAT3 Removal Flag	0: SD card/MMC removal not detected by the SD0DAT3 pin 1: SD card/MMC removal detected by the SD0DAT3 pin.	R(/W)*2
b9	SDD3IN	SD0DAT3 Insertion Flag	0: SD card/MMC insertion not detected by the SD0DAT3 pin 1: SD card/MMC insertion detected by the SD0DAT3 pin.	R(/W)*2
b10	SDD3MON	SD0DAT3 Pin Monitor Flag	0: SD0DAT3 pin level is low 1: SD0DAT3 pin level is high.	R
b31 to b11	—	Reserved	These bits are read as 0	R

Note 1. The value is initialized by a reset and also on a reset triggered by the SDRST flag in the SOFT_RST register.
Note 2. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.

The SD_INFO1 register indicates the detection of a response end or access end for a command sequence. The SD_INFO1 register also indicates the detection of SD card/MMC insertion or removal and the write protection status.

During a multi-block transfer sequence, if CMD12 or CMD52 (SDIO abort) is issued, the ACEND flag becomes 1, but the RSPEND flag remains set to 0.

If the command sequence is stopped due to a communication error or timeout, the ACEND flag or RSPEND flag becomes 1.

After a reset is canceled, the SDD3MON bit, SDD3IN, and SDD3RM flag values are changed according to the status of the SD0DAT3 pin, and their values are changed when data is being transferred in a wide bus mode. These 3 bits are used only for the SD card.

Set the flags to be set to 0 and the flags that are not to be cleared to 1.

RSPEND flag (Response End Detection Flag)

[Setting conditions]

- When reception of the response is complete
- When transmission of a command without response is complete
- When reception of the busy state after R1b response is complete
- When reception of the response to CMD52 that is issued by setting the C52PUB bit to1 is complete for transfer of multiple block read

- When reception of the response to CMD52 that is issued by setting the C52PUB bit to1 is complete for transfer of multiple block write
- When a command sequence is halted because of a communication error or timeout.

[Clearing conditions]

- When 0 is written to the RSPEND flag
- When a command without data is issued.

Note: When a command is issued without data transfer, the RSPEND flag becomes 1 after the command sequence is terminated.

ACEND flag (Access End Detection Flag)

[Setting conditions]

- When read access to the buffer is complete for transfer of a single block read
- When read access to the buffer for the last block of data is complete for transfer of a multiple block read
- When read access to the buffer and reception of the response to CMD12 are complete for transfer of a multiple block read with automatic issuing of CMD12
- When reception of the busy state after reception of the CRC status is complete for transfer of a single block write
- When reception of the busy state after reception of the CRC status of the last block of data is complete for transfer of a multiple block write
- When reception of the response busy state for CMD12 is complete for transfer of a multiple block write with automatic issuing of CMD12
- When reception of the response to CMD12 that was issued by setting the STP bit to 1 is complete for transfer of a multiple block read
- When reception of the response busy state for CMD12 that was issued by setting the STP bit to 1 is complete for transfer of a multiple block write
- When reception of the response to CMD52 that was issued by setting the IOABT bit to 1 is complete in the case of transfer for a multiple block read
- When reception of the response to CMD52 that was issued by setting the IOABT bit to 1 is complete for transfer of a multiple block write
- This bit is set when a command sequence is halted because of a communication error or timeout.

[Clearing conditions]

- When 0 is written to ACEND.
- When the access end bit is set to 1.

Note: The ACEND flag becomes 1 after the command sequence ends.

SDD3RM flag (SD0DAT3 Removal Flag)

[Setting condition]

- After a change in SD0DAT3 from 1 to 0, 2 cycles of PCLKA elapsed with SD0DAT3 held at 0.

[Clearing condition]

- When 0 is written to SDD3RM.

SDD3IN flag (SD0DAT3 Insertion Flag)

[Setting condition]

- After a change in SD0DAT3 from 0 to 1, 2 cycles of PCLKA elapsed with SD0DAT3 held at 1.

[Clearing condition]

- When 0 is written to SDD3IN.

37.2.11 SD Card Interrupt Flag Register 2 (SD_INFO2)

Address(es): SDHI0.SD_INFO2 4006 203Ch

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ILA	CBSY	SD_CLK_CTRLEN	—	—	—	BWE	BRE	SDD0MON	RSPTO	ILR	ILW	DTO	ENDE	CRCE	CMDE
Value after reset:	0*1	0*1	1*1	0	0	0	0*1	0*1	x	0*1	0*1	0*1	0*1	0*1	0*1	0*1

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	CMDE	Command Error Detection Flag	0: Command error not detected 1: Command error detected.	R/W*1
b1	CRCE	CRC Error Detection Flag	0: CRC error not detected 1: CRC error detected.	R/W*1
b2	ENDE	End Bit Error Detection Flag	0: End bit error not detected 1: End bit error detected.	R/W*1
b3	DTO	Data Timeout Detection Flag	0: Data timeout not detected 1: Data timeout detected.	R/W*1
b4	ILW	SD_BUF0 Illegal Write Access Detection Flag	0: Illegal write access to the SD_BUF0 register not detected 1: Illegal write access to the SD_BUF0 register detected.	R/W*1
b5	ILR	SD_BUF0 Illegal Read Access Detection Flag	0: Illegal read access to the SD_BUF0 register not detected 1: Illegal read access to the SD_BUF0 register detected.	R/W*1
b6	RSPTO	Response Timeout Detection Flag	0: Response timeout not detected 1: Response timeout detected.	R/W*1
b7	SDD0MON	SDHI_D0 Pin Status Flag	0: SD0DAT0 pin is low 1: SD0DAT0 pin is high.	R
b8	BRE	SD_BUF0 Read Enable Flag	0: Read access to the SD_BUF0 register disabled 1: Read access to the SD_BUF0 register enabled.	R/W*1
b9	BWE	SD_BUF0 Write Enable Flag	0: Write access to the SD_BUF0 register disabled 1: Write access to the SD_BUF0 register enabled.	R/W*1
b10	—	Reserved	This bit is read as 0	R
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	—	Reserved	This bit is read as 0	R
b13	SD_CLK_CTRLEN	SD_CLK_CTRL Write Enable Flag	0: SD/MMC bus (CMD and DAT lines) is busy, so write access to the SD_CLK_CTRL.CLKEN bit and CLKSEL[7:0] bits is disabled 1: SD/MMC bus (CMD and DAT lines) is not busy, so write access to the SD_CLK_CTRL.CLKEN bit and CLKSEL[7:0] bits is enabled.	R
b14	CBSY	Command Sequence Status Flag	0: Command sequence completed 1: Command sequence in progress (busy).	R
b15	ILA	Illegal Access Error Detection Flag	0: Illegal access error not detected 1: Illegal access error detected.	R/W*1
b31 to b16	—	Reserved	These bits are read as 0	R

Note 1. Only 0 can be written to clear the bit.

The SD_INFO2 register indicates the status of the SD buffer and the status of the SD card/MMC. Set the flags to be set to 0 and the flags that are not to be cleared to 1.

CMDE flag (Command Error Detection Flag)

The CDME flag indicates that a command error was detected. The command sequence is stopped when a command error occurs. When the SDIO_MODE.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence is not completed. Perform the error processing shown in section 37.3.12, IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) or section 37.3.13, IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write), and complete the command sequence.

[Setting conditions]

- The command index of the transmitted command differs from the command index of the received response
- The command index of a command issued within a command sequence differs from the command index of the received response.

[Clearing condition]

- When 0 is written to CMDE.

CRCE flag (CRC Error Detection Flag)

The CRCE flag indicates that a CRC error was detected. The command sequence is stopped when a CRC error occurs. When the SCIO_MODE.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence is not completed. Perform the error processing shown in section 37.3.12, IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) or section 37.3.13, IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write), and complete the command sequence.

[Setting conditions]

- When an error occurs in the CRC status
- When a CRC error occurs in the read data
- When a CRC error occurs in the response
- When a CRC error occurs in response to a command issued within a command sequence.

[Clearing condition]

- When 0 is written to CRCE.

ENDE flag (End Bit Error Detection Flag)

The ENDE flag indicates that an end bit error was detected. The command sequence is stopped when an end bit error occurs. When the SDIO_MODE.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence is not completed. Perform the error processing shown in section 37.3.12, IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) and section 37.3.13, IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write), and complete the command sequence.

[Setting conditions]

- When an error occurs in the response length (and the end bit is not detected)
- When an error occurs in the read data length (and the end bit is not detected among the valid bits)
- When an error occurs in the CRC status length (and the end bit is not detected)
- When a CRC error in the length of a response to a command issued within a command sequence, for example, the end bit is not detected.

[Clearing condition]

- When 0 is written to ENDE.

DTO flag (Data Timeout Detection Flag)

The DTO flag indicates that a data timeout was detected. The command sequence stops when a data timeout occurs.

[Setting conditions]

- After an R1b response, the busy state (SD0DAT0 = 0) continues for longer than the Ncycle time
- After a CRC status, the busy state (SD0DAT0 = 0) continues for longer than the Ncycle time
- After a write data operation, the CRC status is not received before the Ncycle time elapses
- After a read command, read data is not received before the Ncycle time elapses
- After CMD12 is issued within a command sequence, the busy state (SD0DAT0 = 0) continues for longer than the Ncycle time
- After the reception of read data, read data for the next block is not received before the Ncycle time elapses.
- After release of the read wait state, read data for the next block is not received before the Ncycle time elapses.

Note: Ncycle is set in bits [7:4] in SD_OPTION.

[Clearing condition]

- When 0 is written to DTO.

ILW flag (SD_BUF0 Illegal Write Access Detection Flag)

The ILW flag indicates that an SD_BUF0 illegal write access was detected.

[Setting conditions]

- When data is written to the SD_BUF0 register while it is not in the data read/write command state
- When data is written to the SD_BUF0 register while SD_BUF is full
- When data is written to the SD_BUF0 register while an error occurs in the CRC status or CRC status length
- When data is written to the SD_BUF0 register while a busy state after the CRC status continues for longer than Ncycle.

Note: Ncycle is set in bits [7:4] in SD_OPTION.

[Clearing condition]

- When 0 is written to ILW.

ILR flag (SD_BUF0 Illegal Read Access Detection Flag)

The ILR flag indicates that an SD_BUF0 illegal read access was detected.

[Setting conditions]

- When the SD_BUF register is empty while the SD_BUF0 register is read
- When data with a CRC error or END error is read from the SD_BUF0 register.

[Clearing condition]

- When 0 is written to ILR.

RSPTO flag (Response Timeout Detection Flag)

The RSPTO flag indicates that a response timeout was detected. The command sequence is stopped when a response timeout occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence is not completed. Perform the error processing shown in section 37.3.12, IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read) and section 37.3.13, IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write), and complete the command sequence.

[Setting condition]

- When a response is not received even after a time longer than 640 cycles of SD/MMC clock elapses (including a response to a command issued within a command sequence).

[Clearing condition]

- When 0 is written to RSPTO.

SDD0MON flag (SDHI_D0 Pin Status Flag)

If the data timeout (DTO) is set but the response timeout (RSPTO) is not set after the Erase command is issued, the end of the erase sequence (SDD0MON = 1) is confirmed by polling DAT0.

If a communication error or timeout occurs during a write sequence, the DAT0 bit might retain the value 0.

When the SD/MMC clock is stopped, the DAT0 bit retains the value before the clock is stopped.

BRE flag (SD_BUF0 Read Enable Flag)

[Setting conditions]

- When data set in the SD_SIZE register is stored in SD_BUF0 for a single block transfer
- When data set in the SD_SIZE register is stored in either bank 1 or bank 2 of the SD_BUF0 register for a multiple block transfer.

[Clearing conditions]

- When 0 is written to BRE
- Reading of a block of data from the SD_BUF0 register by DMA transfer.

When data is read from the SD_BUF0 register by the CPU, clear BRE then read the amount of data specified in SD_SIZE.

Even if a CRC error or an END error occurs while block data is read, data is stored in the SD_BUF0 register and BRE is set.

BWE flag (SD_BUF0 Write Enable Flag)

[Setting conditions]

- When the SD_BUF0 register is empty for a single block transfer
- When either bank 1 or bank 2 of the SD_BUF0 register is empty for a multiple block transfer.

[Clearing conditions]

- When 0 is written to BWE
- Writing of a block of data to the SD_BUF0 register by DMA transfer.

When data is written to the SD_BUF0 register by the CPU, clear BWE and then write the amount of data specified in SD_SIZE.

SD_CLK_CTRLEN flag (SD_CLK_CTRL Write Enable Flag)

When a command sequence is started by writing to SD_CMD, the CBSY bit is set to 1 and, at the same time, the SD_CLK_CTRLEN bit is set to 0. The SD_CLK_CTRLEN bit is set to 1 after 8 cycles of SDCLK elapse after setting of the CBSY bit to 0 because of the completion of the command sequence.

ILA flag (Illegal Access Error Detection Flag)

[Setting conditions]

- Writing of data to SD_CMD within a command sequence (CBSY = 1)
- When SD_CMD[11] = 1 (command with data transfer) and SD_CMD[7:0] = 0000_1100b (CMD12) are set in SD_CMD.

[Clearing condition]

- When 0 is written to ILA.

37.2.12 SD INFO1 Interrupt Mask Register (SD_INFO1_MASK)

Address(es): SDHI0.SD_INFO1_MASK 4006 2040h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SDD3INM	SDD3RMM	—	—	—	—	—	ACENDM	—	RSPENDM
Value after reset:	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	1

Bit	Symbol	Bit name	Description	R/W
b0	RSPENDM	Response End Interrupt Request Mask	0: Response end interrupt request is not masked 1: Response end interrupt request is masked.	R/W
b1	—	Reserved	This bit is read as 0	R
b2	ACENDM	Access End Interrupt Request Mask	0: Access end interrupt request is not masked 1: Access end interrupt request is masked.	R/W
b4, b3	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	SDD3RMM	SD0DAT3 Removal Interrupt Request Mask	0: SD card/MMC removal interrupt request by the SD0DAT3 pin is not masked 1: SD card/MMC removal interrupt request by the SD0DAT3 pin is masked.	R/W
b9	SDD3INM	SD0DAT3 Insertion Interrupt Request Mask	0: SD card/MMC insertion interrupt request by the SD0DAT3 pin is not masked 1: SD card/MMC insertion interrupt request by the SD0DAT3 pin is masked.	R/W
b31 to b10	—	Reserved	These bits are read as 0	R

The SD_INFO1_MASK register enables or disables the interrupt requests from the status flags in the SD_INFO1 register. See Table 37.5, Interrupt sources for details of the relationship between the status flags and the requested interrupt source.

37.2.13 SD INFO2 Interrupt Mask Register (SD_INFO2_MASK)

Address(es): SDHI0.SD_INFO2_MASK 4006 2044h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ILAM	—	—	—	—	—	BWEM	BREM	—	RSPTOM	ILRM	ILWM	DTOM	ENDEM	CRCEM	CMDEM
Value after reset:	1	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b0	CMDEM	Command Error Interrupt Request Mask	0: Command error interrupt request is not masked 1: Command error interrupt request is masked.	R/W
b1	CRCEM	CRC Error Interrupt Request Mask	0: CRC error interrupt request is not masked 1: CRC error interrupt request is masked.	R/W
b2	ENDEM	End Bit Error Interrupt Request Mask	0: End bit detection error interrupt request is not masked 1: End bit detection error interrupt request is masked.	R/W
b3	DTOM	Data Timeout Interrupt Request Mask	0: Data timeout interrupt request is not masked 1: Data timeout interrupt request is masked.	R/W
b4	ILWM	SD_BUF0 Register Illegal Write Interrupt Request Mask	0: Illegal write detection interrupt request for the SD_BUF0 register is not masked 1: Illegal write detection interrupt request for the SD_BUF0 register is masked.	R/W
b5	ILRM	SD_BUF0 Register Illegal Read Interrupt Request Mask	0: Illegal read detection interrupt request for the SD_BUF0 register is not masked 1: Illegal read detection interrupt request for the SD_BUF0 register is masked.	R/W
b6	RSPTOM	Response Timeout Interrupt Request Mask	0: Response timeout interrupt request is not masked 1: Response timeout interrupt request is masked.	R/W
b7	—	Reserved	This bit is 0 when read.	R
b8	BREM	BRE Interrupt Request Mask	0: Read enable interrupt request for the SD buffer is not masked 1: Read enable interrupt request for the SD buffer is masked.	R/W
b9	BWEM	BWE Interrupt Request Mask	0: Write enable interrupt request for the SD_BUF0 register is not masked 1: Write enable interrupt request for the SD_BUF0 register is masked.	R/W
b10	—	Reserved	This bit is read as 0.	R
b11	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b14 to b12	—	Reserved	These bits are read as 0.	R
b15	ILAM	Illegal Access Error Interrupt Request Mask	0: Illegal access error interrupt request is not masked 1: Illegal access error interrupt request is masked.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. When the SD_INFO2_MASK.BWEM bit is 0 or the SD_INFO2_MASK.BREM bit is 0, set the SD_DMAEN.DMAEN bit to 0. When the SD_DMAEN.DMAEN bit is 1, set the SD_INFO2_MASK.BWEM bit to 1 and the SD_INFO2_MASK.BREM bit to 1.

The SD_INFO2_MASK register enables or disables the interrupt requests from the status flags in the SD_INFO2 register. See Table 37.5 for details of the relationship between the status flags and the requested interrupt source.

37.2.14 SD Clock Control Register (SD_CLK_CTRL)

Address(es): SDHI0.SD_CLK_CTRL 4006 2048h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CLKCTRLEN	CLKEN	CLKSEL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CLKSEL[7:0]	SDHI Clock Frequency Select*1	b7 b0 0 0 0 0 0 0 0 0: PCLKA divided by 2 0 0 0 0 0 0 0 1: PCLKA divided by 4 0 0 0 0 0 0 1 0: PCLKA divided by 8 0 0 0 0 0 1 0 0: PCLKA divided by 16 0 0 0 0 1 0 0 0: PCLKA divided by 32 0 0 0 1 0 0 0 0: PCLKA divided by 64 0 0 1 0 0 0 0 0: PCLKA divided by 128 0 1 0 0 0 0 0 0: PCLKA divided by 256 1 0 0 0 0 0 0 0: PCLKA divided by 512 Other settings are prohibited.	R/W
b8	CLKEN	SD/MMC Clock Output Control*1	0: SD/MMC clock output is disabled (SD0CLK signal fixed low) 1: SD/MMC clock output enabled.	R/W
b9	CLKCTRLEN	SD/MMC Clock Output Automatic Control Select	0: Automatic control of SD/MMC clock output is disabled 1: Automatic control of SD/MMC clock output is enabled.	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Bits CLKSEL[7:0] and CLKEN cannot be write accessed when the SD_INFO2.SD_CLK_CTRLEN flag is 0.

The SDCLKCTRL register controls the SD/MMC clock frequency settings and output. Set the CLKEN bit to 1 before writing to the SD_CMD register to start a command sequence. Do not write to the SDCLKCTRL register when the SD_INFO2.SD_CLK_CTRLEN flag is 0.

CLKCTRLEN bit (SD/MMC Clock Output Automatic Control Select)

This automatic control function for SD/MMC clock output causes the SD/MMC clock output only within a command sequence.

The timing with which SD/MMC clock output starts and stops is as follows:

- SD/MMC clock output starts after writing to SD_CMD
- SD/MMC clock output stops when 8 cycles of SD/MMC clock elapse after the end of the command sequence.

In addition, SD/MMC clock is fixed to 0 while the CLKEN bit in the SD_CLK_CTRL register is 0, regardless of the value of this bit.

37.2.15 Transfer Data Length Register (SD_SIZE)

Address(es): SDHI0.SD_SIZE 4006 204Ch

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	LEN[9:0]									
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	LEN[9:0]	Transfer Data Size Setting	Set the transfer data size*1	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Do not rewrite these bits when the SD_INFO2.CBSY flag is 1.

The SD_SIZE register is used to set the transfer data size.

LEN[9:0] bits (Transfer Data Size Setting)

When using single block transfer, the transfer data size can be set from 1 byte to 512 bytes. When CMD12 is automatically issued during a multi-block transfer sequence (CMD18 and CMD25), the transfer data size can only be set to 512 bytes. When CMD12 is not automatically issued during a multi-block transfer sequence, the transfer data size can be set to 32, 64, 128, 256, or 512 bytes. However, a 32-, 64-, 128-, or 256-byte multi-block read transfer can only be performed during an SDIO multi-block transfer (CMD53). Do not set these bits to 0 when using a command that includes data transfer.

37.2.16 SD Card Access Control Option Register (SD_OPTION)

Address(es): SDHI0.SD_OPTION 4006 2050h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	WIDTH	—	WIDTH8	—	—	—	—	TOUTMASK	TOP[3:0]				—	—	—	—
Value after reset:	0*2	1	0*2	0	0	0	0	0*2	1*2	1*2	1*2	0*2	1	1	1	0

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3 to b1	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7 to b4	TOP[3:0]	Timeout Counter*1	b7 b4 0 0 0 0: SDHI clock × 2^{13} 0 0 0 1: SDHI clock × 2^{14} 0 0 1 0: SDHI clock × 2^{15} 0 0 1 1: SDHI clock × 2^{16} 0 1 0 0: SDHI clock × 2^{17} 0 1 0 1: SDHI clock × 2^{18} 0 1 1 0: SDHI clock × 2^{19} 0 1 1 1: SDHI clock × 2^{20} b7 b4 1 0 0 0: SDHI clock × 2^{21} 1 0 0 1: SDHI clock × 2^{22} 1 0 1 0: SDHI clock × 2^{23} 1 0 1 1: SDHI clock × 2^{24} 1 1 0 0: SDHI clock × 2^{25} 1 1 0 1: SDHI clock × 2^{26} 1 1 1 0: SDHI clock × 2^{27} 1 1 1 1: Setting prohibited	R/W
b8	TOUTMASK	Timeout Mask	0: Activate timeout 1: Deactivate timeout. (RSPTO bit and DTO bit of SD_INFO2 and E6 to E0 bits of SDERRSTS2 are not set.) When timeout occurs because of an inactivated timeout, execute a software reset to terminate the command sequence.	R/W
b12 to b9	—	Reserved	These bits are read as 0	R
b13	WIDTH8*2	BUS WIDTH	See b15, WIDTH bit	R/W
b14	—	Reserved	This bit is read as 1	R
b15	WIDTH	BUS WIDTH*2	b15 b13 0 1: 8-bit width 0 0: 4-bit width 1 0: 1-bit width 1 1: 1-bit width. In case of 1-byte write transfer, set 4-bit width or 1-bit width. Do not set 8-bit width.	R/W
b31 to b16	—	Reserved	These bits are read as 0	R

Note 1. Do not rewrite these bits when the SD_INFO2.CBSY flag is 1.
Note 2. The initial value is applied at a reset and when the SOFT_RST.SDRST flag is 0.

The SD bus width and timeout counter are set in the SD_OPTION register.

37.2.17 SD Error Status Register 1 (SD_ERR_STS1)

Address(es): SDHI0.SD_ERR_STS1 4006 2058h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	CRCTK[2:0]			CRCTKE	RDCRCE	RSPCRCE1	RSPCRCE0	—	—	CRCLENE	RDLENE	RSPLENE1	RSPLENE0	CMDE1	CMDE0
Value after reset:	0	0*3	1*3	0*3	0*3	0*3	0*3	0*3	0	0	0*3	0*3	0*3	0*3	0*3	0*3

x: Undefined

Bit	Symbol	Bit name	Description	R/W
b0	CMDE0	Command Error Flag 0	0: Command index field value for a command*1 response is error free 1: Command index field value for a command*1 response is in error.	R
b1	CMDE1	Command Error Flag 1	0: Command index field value for a command*2 response is error free 1: Command index field value for a command*2 response is in error (by setting the SD_CMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the CMDE0 flag).	R
b2	RSPLENE0	Response Length Error Flag 0	0: Command*1 response length is error free 1: Command*1 response length is in error.	R
b3	RSPLENE1	Response Length Error Flag 1	0: Command*2 response length is error free 1: Command*2 response length is in error (by setting the SD_CMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the RSPLENE0 flag).	R
b4	RDLENE	Read Data Length Error Flag	0: Read data length error did not occur 1: Read data length error occurred.	R
b5	CRCLENE	CRC Status Token Length Error Flag	0: CRC status token length error did not occur 1: CRC status token length error occurred.	R
b7, b6	—	Reserved	These bits are read as 0	R
b8	RSPCRCE0	Response CRC Error Flag 0	0: No CRC error detected in command*1 response 1: CRC error detected in command*1 response.	R
b9	RSPCRCE1	Response CRC Error Flag 1	0: No CRC error detected in command*2 response (by setting the SD_CMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the RSPCRCE0 flag) 1: CRC error detected in command*2 response.	R
b10	RDCRCE	Read Data CRC Error Flag	0: No CRC error detected in read data 1: CRC error detected in read data.	R
b11	CRCTKE	CRC Status Token Error Flag	0: No error detected in CRC status token 1: Error detected in CRC status token.	R
b14 to b12	CRCTK[2:0]	CRC Status Token	Store the CRC status token value (normal value is 010b)	R
b15	—	Reserved	This bit is read as 0	R
b31 to b16	—	Reserved	These bits are read as undefined	R

Note 1. CMD other than CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Note 2. CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Note 3. The initial value is applied at a reset and when the SOFT_RST.SDRST flag is 0.

The SD_ERR_STS1 register indicates the CRC status token, CRC error, end bit error, and command error.

37.2.18 SD Error Status Register 2 (SD_ERR_STS2)

Address(es): SDHI0.SD_ERR_STS2 4006 205Ch

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	CRCBSYTO	CRCTO	RDTO	BSYTO1	BSYTO0	RSPTO1	RSPTO0
Value after reset:	0	0	0	0	0	0	0	0	0	0*4	0*4	0*4	0*4	0*4	0*4	0*4

Bit	Symbol	Bit name	Description	R/W
b0	RSPTO0	Response Timeout Flag 0	0: After a command*1 was issued, a response was received in less than 640 cycles of the SD/MMC clock 1: After a command*1 was issued, a response was not received even after 640 cycles or more of the SD/MMC clock elapsed.	R
b1	RSPTO1	Response Timeout Flag 1	0: After a command*2 was issued, a response was received in less than 640 cycles of the SD/MMC clock 1: After a command*2 was issued, a response was not received even after 640 cycles or more of the SD/MMC clock elapsed (by setting the SD_CMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the RSPTO0 flag).	R
b2	BSYTO0	Busy Timeout Flag 0	0: After the R1b response was received, the SD/MMC was released from the busy state during the specified period*3 1: After the R1b response was received, the SD/MMC was in the busy state even after the specified period*3 elapsed.	R
b3	BSYTO1	Busy Timeout Flag 1	0: After CMD12 was automatically issued, the SD/MMC was released from the busy state during the specified period*3 1: After CMD12 was automatically issued, the SD/MMC was in the busy state even after the specified period*3 elapsed (by setting the SD_CMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the BSYTO0 flag).	R
b4	RDTO	Read Data Timeout Flag	After a read command is issued, this flag becomes 1 when read data is not received even after the specified period*3 elapses. After read data is received, this flag becomes 1 when the next block of read data is not received even after the specified period*3 elapses. After the SD/MMC exits the read wait state, this flag becomes 1 when the next block of read data is not received even after the specified period*3 elapses.	R
b5	CRCTO	CRC Status Token Timeout Flag	0: After data was written to the SD card/MMC, a CRC status token was received during the specified period*3 1: After CRC data was written to the SD card/MMC, a CRC status token was not received even after the specified period*3 elapsed.	R
b6	CRCBSYTO	CRC Status Token Busy Timeout Flag	0: After a CRC status token was received, the SD/MMC was released from the busy state during the specified period*3 1: After a CRC status token was received, the SD/MMC is in the busy state even after the specified period*3 elapsed.	R
b31 to b7	—	Reserved	These bits are read as 0.	R

Note 1. CMD other than CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Note 2. CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Note 3. Set the SD_OPTION.TOP[3:0] bits to select the number of *n* cycles.

Note 4. The initial value is applied at a reset and when the SOFT_RST.SDRST flag is 0.

The SD_ERR_STS2 register indicates the timeout status.

37.2.19 SD Buffer Register (SD_BUF0)

Address(es): SDHI0.SD_BUF0 4006 2060h

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

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

x: Undefined

When writing to the SD card, the write data is written to this register. When reading from the SD card, the read data is read from this register. This register is internally connected to two 512-byte buffers.

If both buffers are not empty when executing multiple block read, the SD card/MMC clock is stopped to suspend receiving data. When one of the buffers is empty, the SD card/MMC clock is supplied to resume receiving data.

37.2.20 SDIO Mode Control Register (SDIO_MODE)

Address(es): SDHI0.SDIO_MODE 4006 2068h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	C52PUB	IOABT	—	—	—	—	—	RWREQ	—	INTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	INTEN	SDIO Interrupt Acceptance Enable*1	0: SDIO interrupt accept disabled 1: SDIO interrupt accept enabled.	R/W
b1	—	Reserved	This bit is read as 0.	R
b2	RWREQ	Read Wait Request	0: SD/MMC exits read wait state 1: Request for SD/MMC to enter read wait state.	R/W
b7 to b3	—	Reserved	These bits are read as 0	R
b8	IOABT	SDIO Abort	If this bit is set to 1 during multi-block transfer triggered by CMD53, CMD52 is immediately issued, and the command sequence is aborted	R/W
b9	C52PUB	SDIO None Abort	If this bit is set to 1 during multi-block transfer triggered by CMD53, CMD52 is issued after the transfer process is complete, and the command sequence is completed	R/W
b31 to b10	—	Reserved	These bits are read as 0	R

Note 1. Do not rewrite this bit when the SD_INFO2.CBSY flag is 1.

The SDIO_MODE register controls reception of the SDIO interrupt, CMD52 issuance during multi-block transfer, and read wait request. Do not set bits C52PUB and IOABT to 1 at the same time.

RWREQ bit (Read Wait Request)

When RWREQ is set to 1 in the CMD53 (multiple block) read sequence, the block transfer enters the read wait state between blocks.

[Read wait state releasing]

- The read wait state is released when RWREQ is set to 0 in the read wait state
- When IOABT is set to 1 in the read wait state, RWREQ is automatically set to 0 after CMD52 is issued, and then the read wait state is released
- When C52PUB and RWREQ are set to 1 simultaneously in the CMD53 (multiple block) read sequence, the read wait state is not automatically released. Therefore, after the CMD52 response is received, clear RWREQ. Be sure to set RWREQ and C52PUB simultaneously.

When RWREQ is set to 1 while the last block in the CMD53 (multiple block) read sequence is transferred, the read wait state is not entered and RWREQ is automatically set to 0 by setting access end. Set RWREQ to 1 after the response end flag is set.

IOABT bit (SDIO Abort)

- When IOABT is set to 1 in the CMD53 (multiple block) sequence, the CMD53 sequence is halted and CMD52 is issued. However, if a command sequence is halted because of a communication error or timeout, CMD52 is not issued. Although continued buffer access is possible even after IOABT is set to 1, the buffer access error bit (ILR or ILW) in SD_INFO2 will be set accordingly. Set SD_ARG before setting IOABT to 1.
- When IOABT is set to 1 during transfer for single block write, the access end flag is set when the SD_BUF0 register becomes empty, and CMD52 is not issued. If the SD_BUF0 register contains data, the access end flag is set on completion of reception of the busy state without CMD52 having been issued.
- When IOABT is set to 1 during transfer for single block read, the access end flag is set immediately after setting IOABT, and CMD52 is not issued
- When IOABT is set to 1 during reception of the busy state after an R1b response, the access end flag is set on completion of reception of the busy state without CMD52 being issued
- When IOABT is set to 1 after a command sequence completes, CMD52 is not issued and the access end flag is not set.
- Set IOABT to 1 after the response end flag is set
- Set IOABT to 0 after the access end flag is set.

C52PUB bit (SDIO None Abort)

- When C52PUB is set to 1 in the CMD53 (multiple block) write sequence, CMD52 is automatically issued between blocks if the SD_BUF0 register becomes empty. C52PUB is automatically set to 0 after reception of the response to CMD52 is completed. Additionally, if C52PUB is set to 1 while the last block is being transferred, CMD52 is not issued. In this case, C52PUB is automatically set to 0 after the access end flag is set to 1.
- When C52PUB and RWREQ are set to 1 in the CMD53 (multiple block) read sequence, the block transfer enters the read wait state between blocks and CMD52 is automatically issued. C52PUB is automatically set to 0 after reception of the response to CMD52 is completed. Additionally, if C52PUB is set to 1 while the last block is being transferred, CMD52 is not issued. In this case, C52PUB is automatically set to 0 after the access end flag is set to 1.
- If C52PUB is set to 1 in the CMD53 (multiple block) read sequence, be sure to set RWREQ to 1 in addition to C52PUB
- Set SD_ARG before setting C52PUB to 1
- Set C52PUB to 1 after the response end flag is set.

37.2.21 SDIO Interrupt Flag Register (SDIO_INFO1)

Address(es): SDHI0.SDIO_INFO1 4006 206Ch

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EXWT	EXPUB52	—	—	—	—	—	—	—	—	—	—	—	—	—	IOIRQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	0

Bit	Symbol	Bit name	Description	R/W
b0	IOIRQ	SDIO Interrupt Status Flag	0: SDIO interrupt is not detected 1: SDIO interrupt is detected.	R(/W)*1
b2, b1	—	Reserved	The read value is undefined. The write value should be 1.	R/W
b13 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	EXPUB52	EXPUB52 Status Flag	Indicates the status of the EXPUB52	R(/W)*1
b15	EXWT	EXWT Status Flag	Indicates the status of the EXWT	R(/W)*1
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to clear the bit.

The SDIO_INFO1 register indicates the status of the SDIO card access. When clearing a flag, set the bits to be set to 0 and set the bits that are not to be cleared to 1.

IOIRQ flag (SDIO Interrupt Status Flag)

[Setting condition]

- When SDIO interrupt from an SDIO card is received while INTEN in SDIO_MODE is set to 1.

[Clearing condition]

- When 0 is written to IOIRQ.*1

Note 1. Before clearing this bit, access the SDIO card to negate the SDIO interrupt signal from the SDIO card. If the interrupt signal is not negated, this bit can be set again.

EXPUB52 flag (EXPUB52 Status Flag)

[Setting conditions]

- While the last block in the CMD53 (multiple block) sequence is transferred, C52PUB in SDIO_MODE is set to 1
- While C52PUB is set to 1 in the CMD53 (multiple block) write sequence, the last block is transferred.

[Clearing condition]

- When 0 is written to EXPUB52.

EXWT flag (EXWT Status Flag)

[Setting condition]

- While the last block in the CMD53 (multiple block) read sequence is transferred, RWREQ in SDIO_MODE is set to 1

[Clearing condition]

- When 0 is written to EXWT.

37.2.22 SDIO INFO1 Interrupt Mask Register (SDIO_INFO1_MASK)

Address(es): SDHI0.SDIO_INFO1_MASK 4006 2070h

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

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EXWTM	EXPUB52M	—	—	—	—	—	—	—	—	—	—	—	—	—	IOIRQM
Value after reset:	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Bit name	Description	R/W
b0	IOIRQM	IOIRQ Interrupt Mask Control	0: IOIRQ interrupt not masked 1: IOIRQ interrupt masked.	R/W
b2, b1	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b13 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	EXPUB52M	EXPUB52 Interrupt Request Mask Control	0: EXPUB52 interrupt request not masked 1: EXPUB52 interrupt request masked.	R/W
b15	EXWTM	EXWT Interrupt Request Mask Control	0: EXWT interrupt request not masked 1: EXWT interrupt request masked.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SDIO_INFO1_MASK register enables or disables the interrupt requests from the status flags in the SDIO_INFO1 register. See Table 37.5, Interrupt sources for details of the relationship between the status flags and the requested interrupt source.

37.2.23 DMA Mode Enable Register (SD_DMAEN)

Address(es): SDHI0.SD_DMAEN 4006 21B0h

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

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

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	This bit is read as 0	R
b1	DMAEN	DMA Transfer Enable*1, *2	0: Using DMA transfer to access the SD_BUF0 register is disabled 1: Using DMA transfer to access the SD_BUF0 register is enabled.	R/W
b3, b2	—	Reserved	These bits are read as 0	R
b4	—	Reserved	This bit is read as 1	R
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	—	Reserved	These bits are read as 0	R
b9, b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11, b10	—	Reserved	These bits are read as 0	R

Bit	Symbol	Bit name	Description	R/W
b12	—	Reserved	This bit is read as 1	R
b31 to b13	—	Reserved	These bits are read as 0	R

Note 1. Do not rewrite this bit when the SD_INFO2.CBSY bit is 1.
Note 2. When the SD_INFO2_MASK.BWEM bit is 0 or the SD_INFO2_MASK.BREM bit is 0, set the SD_DMAEN.DMAEN bit to 0. When the SD_DMAEN.DMAEN bit is 1, set the SD_INFO2_MASK.BWEM bit to 1 and the SD_INFO2_MASK.BREM bit to 1.

The SD_DMAEN register enables or disables DMA transfer.

DMAEN bit (DMA Transfer Enable)

When using DMA transfer to access the SD buffer, set the DMAEN bit to 1 before setting the SD_CMD register.

37.2.24 Software Reset Register (SOFT_RST)

Address(es): SDHI0.SOFT_RST 4006 21C0h

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

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

Bit	Symbol	Bit name	Description	R/W
b0	SDRST	Software Reset Control	0: SD/MMC host interface software reset 1: SD/MMC host interface software reset canceled.	R/W
b2, b1	—	Reserved	These bits are read as 1	R
b31 to b3	—	Reserved	These bits are read as 0	R

Table 37.4 lists the bits and flags initialized by SD/MMC host interface software reset.

Table 37.4 Bits and flags initialized by SD/MMC host interface software reset

Register	Bit/Flag
SD_STOP	SEC
SD_INFO1	RSPEND, ACEND
SD_INFO2	CMDE, CRCE, ENDE, DTO, ILW, ILR, RSPTO, SDD0MON, BRE, BWE, SD_CLK_CTRLEN, ILA
SD_CLK_CTRL	CLKEN
SD_OPTION	TOP[3:0], WIDTH Bits b8 and b13 in the SD_OPTION register are also initialized by the SDHI software reset
SD_ERR_STS1	CMDE0, CMDE1, RSPLENE0, RSPLENE1, RDLENE, CRCLENE, RSPCRCE0, RSPCRCE1, RDCRCE, CRCTKE, CRCTK[2:0]
SD_ERR_STS2	RSPTO0, RSPTO1, BSYTO0, BSYTO1, RDTO, CRCTO, CRCBSYTO
SDIO_INFO1	IOIRQ, EXPUB52, EXWT

37.2.25 SD Interface Mode Setting Register (SDIF_MODE)

Address(es): SDHI0.SDIF_MODE 4006 21CCh

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

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

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b1	—	Reserved	These bits are read as 0.	R
b8	NOCHKCR	CRC Check Mask (for MMC test commands)	Set when CRC16 or CRC status value check is not executed: 0: CRC Check is valid 1: CRC Check is invalid. CRC16 value is ignored for read and CRC Status value is ignored for write.	R/W
b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b10	—	Reserved	These bits are read as 0.	R

37.2.26 Swap Control Register (EXT_SWAP)

Address(es): SDHI0.EXT_SWAP 4006 21E0h

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

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

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	This bit is read as 0	R
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	—	Reserved	This bit is read as 0	R
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	—	Reserved	This bit is read as 0	R
b6	BWSWP	SD_BUF0 Swap Write*1	0: Normal write operation 1: Swap the byte endianness before writing to the SD_BUF0 register.	R/W
b7	BRSWP	SD_BUF0 Swap Read*1	0: Normal read operation 1: Swap the byte endianness before reading the SD_BUF0 register.	R/W
b10 to b8	—	Reserved	These bits are read as 0	R
b12, b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14, b13	—	Reserved	These bits are read as 0	R
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b31 to b16	—	Reserved	These bits are read as 0	R

Note 1. Do not rewrite this bit when the SD_INFO2.CBSY flag is 1.

The EXT_SWAP register is used to select whether the byte endianness is swapped when accessing the SD_BUF0 register. See section 37.3.1 for details on the differences in accessing the SD_BUF0 register based on the EXT_SWAP register value.

37.3 Operation

37.3.1 SD/MMC Interface

When data is read from the SD card/MMC, the process is as follows:

1. The SD/MMC host interface receives data from the SD card/MMC through the SD0DAT signal, see Figure 37.2 and Figure 37.3.
2. The received data is stored in the SD_BUF register of the MMC host interface, see Figure 37.4.
3. The data stored in the SD_BUF register is read from the SD_BUF0 register, see Figure 37.5.

When data is written to the SD card/MMC, the specified procedure is reversed.

When accessing the SD_BUF0 register, pay attention to the transfer order in SD0DAT and the store order in the SD_BUF register. The byte endianness of the data read from or written to the SD_BUF0 register can be swapped using the SDSWAP register. See Figure 37.6.

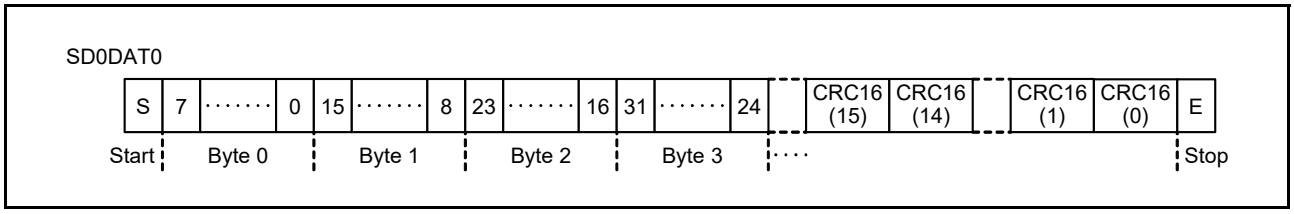


Figure 37.2 SD0DAT in 1-bit width mode

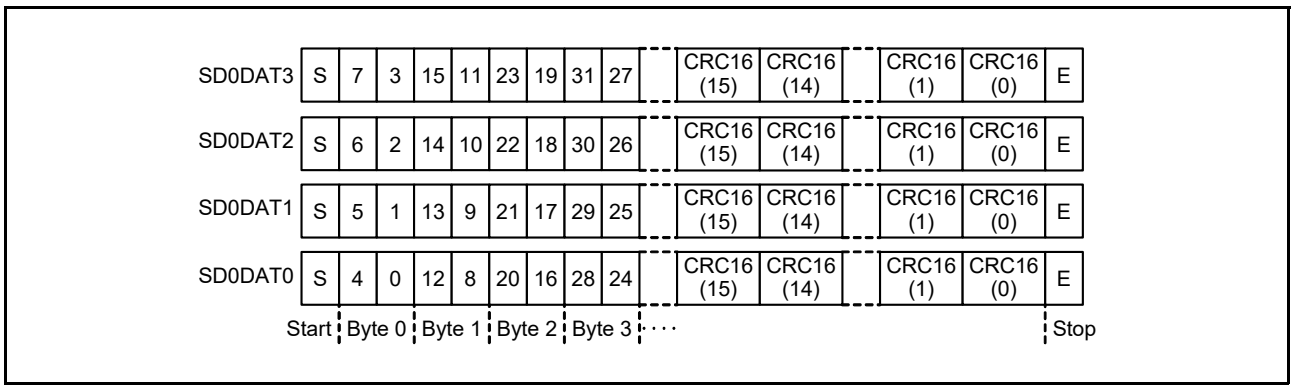


Figure 37.3 SD0DAT in 4-bit width mode

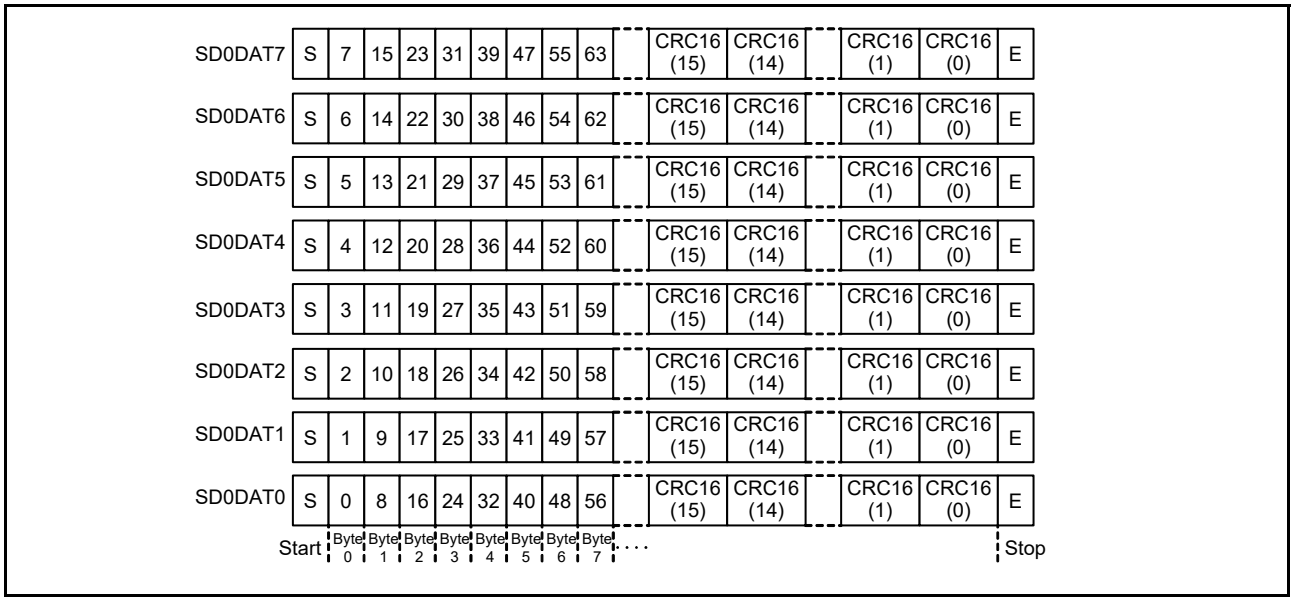


Figure 37.4 SD0DAT in 8-bit width mode

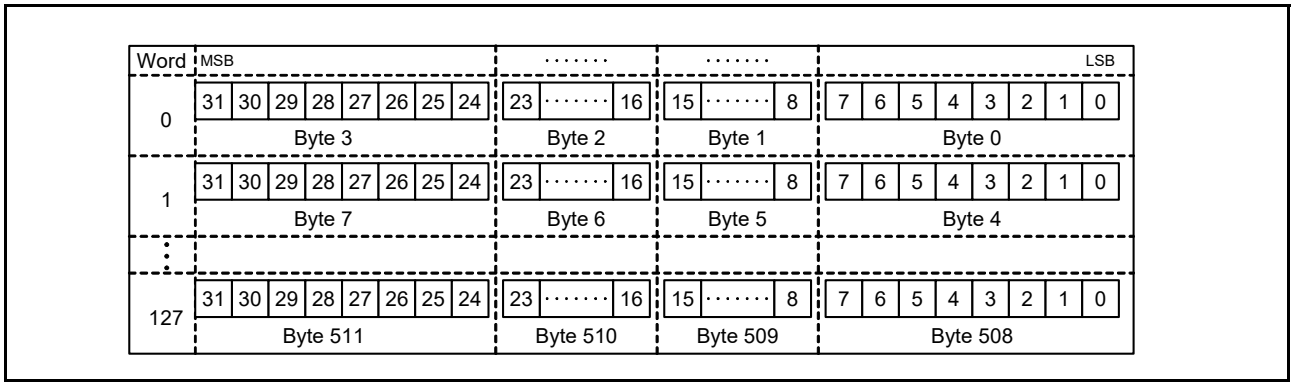


Figure 37.5 SD_BUF register data storage

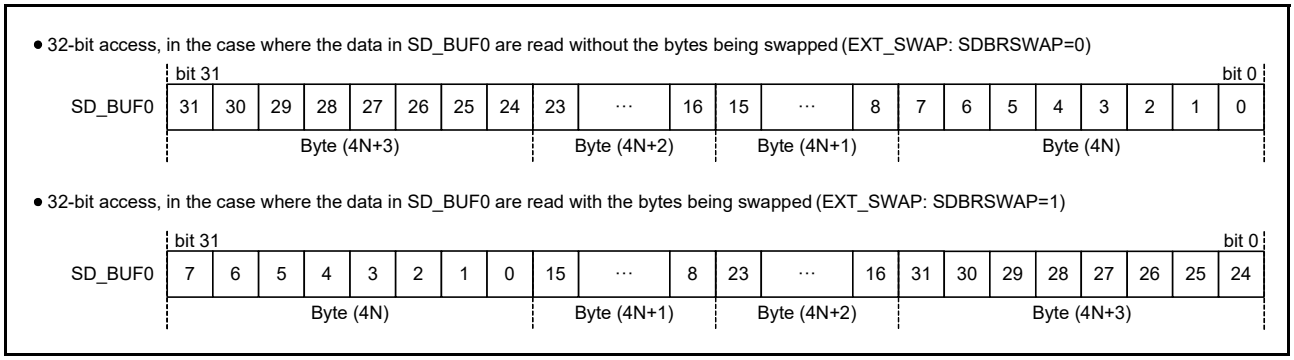


Figure 37.6 Reading from the SD_BUF0 register

37.3.2 Card Detect/Write Protect

37.3.2.1 Card Detect

The SD/MMC host interface has two types of card detect functions.

(1) SD card detect with SD0DAT3

Figure 37.7 shows the timing chart when the SD card is detected with SD0DAT3. In addition, SD0DAT3 is pulled down by the host device, and the resistance value for pulling down is determined by the specification of the SD host device.

(2) Card insertion

When an SD card is inserted, SD0DAT3 is pulled up and SDD3IN in SD_INFO1 is set to 1. It is cleared by writing 0.

(3) Card removal

When an SD card is removed, SD0DAT3 is pulled down and SDD3RM in SD_INFO1 is set to 1. It is cleared by writing 0.

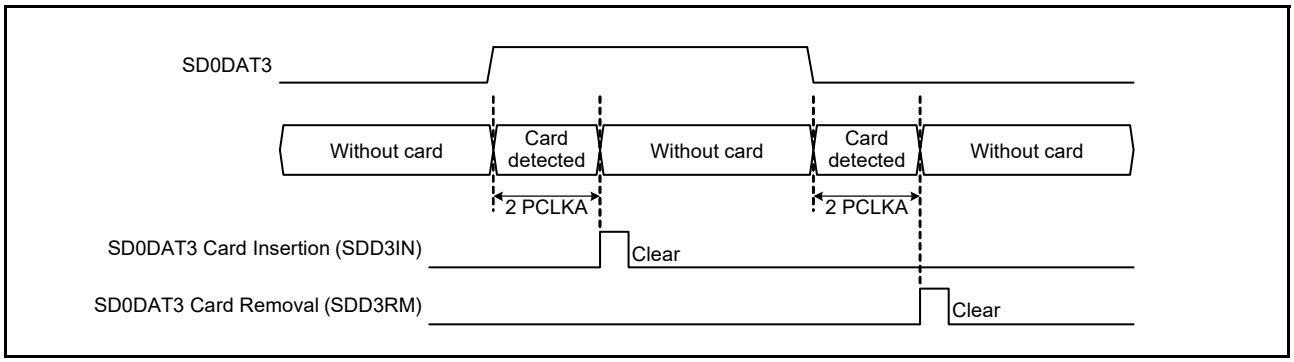


Figure 37.7 SD card detect with SD0DAT3

37.3.2.2 Write Protect

The SD/MMC host interface has two types of write protect functions.

(1) Write protect with SD0WP

SD0WP is connected to the card socket and pulled up or pulled down by the card insertion. The selection of pulling up or pulling down and the resistance value is determined by the specification of the SD host device. When the SD0WP state is reflected to SDWPMON in SD_INFO1, the write protect state is set after the SD card is inserted.

(2) Write protect with command

The internal write protection of the card and the lock/unlock operation of the card are realized by the command.

37.3.3 Interrupt Request and DMA Transfer Request

37.3.3.1 Interrupts

Table 37.5 lists the SDHI interrupt sources. The SDHI requests an interrupt when:

- Status flags in registers SD_INFO1, SD_INFO2, or SDIO_INFO1 become 1
- The associated bits in registers SD_INFO1_MASK, SD_INFO2_MASK, and SDIO_INFO1_MASK are 0.

When clearing the status flags in registers SD_INFO1, SD_INFO2, and SDIO_INFO1, write 0 to the status flags to be cleared and write 1 to the status flags that are not being cleared.

Table 37.5 Interrupt sources

Interrupt sources	Status flag register		Interrupt mask register		Interrupt name
	Register symbol	Bit symbol	Register symbol	Bit symbol	
Card Access Interrupt	SD_INFO1	ACEND	SD_INFO1_MASK	ACENDM	SDHI_MMC0_ACCS
		RSPEND		RSPENDM	
	SD_INFO2	ILA	SD_INFO2_MASK	ILAM	
		BWE		BWEM	
		BRE		BREM	
		RSPTO		RSPTOM	
		ILR		ILRM	
		ILW		ILWM	
		DTO		DTOM	
		ENDE		ENDEM	
		CRCE		CRCEM	
		CMDE		CMDEM	
SDIO Access Interrupt	SDIO_INFO1	EXWT	SDIO_INFO1_MASK	EXWTM	SDHI_MMC0_SDIO
		EXPUB52		EXPUB52M	
		IOIRQ		IOIRQM	
Card Detect Interrupt	SD_INFO1	SDD3IN	SD_INFO1_MASK	SDD3INM	SDHI_MMC0_CARD
		SDD3RM		SDD3RMM	

37.3.3.2 DMA Transfer Requests (SDHI_MMC0_ODMSDBREQ)

The SD/MMC host interface has two types of DMA transfer request: SD_BUF write DMA transfer request and SD_BUF read DMA transfer request. SD/MMC host interface uses SDHI_MMC0_ODMSDBREQ for DMA transfer request. For details on event number of SDHI_MMC0_ODMSDBREQ, see Table 14.4 in section 14, Interrupt Controller Unit (ICU).

(1) SD_BUF write DMA transfer request

- When the BWE bit in SD_INFO2 is set to 1 while the DMAEN bit in SD_DMAEN is set to 1, the SD_BUF write DMA transfer request is asserted
- The SD_BUF write DMA transfer request is negated when the last data in one block (according to the transfer data size set in SD_SIZE) is transferred. The SD_BUF write DMA transfer request is also negated by clearing the SDRST bit in SOFT_RST to 0 or setting the STP bit in SD_STOP to 1. However, if a communication error or timeout occurs at the DMA transfer, the SD_BUF write DMA transfer request is not negated.
- The BWE bit in SD_INFO2 is cleared after transfer of the last data in one block following a request for writing to SD_BUF by DMA transfer
- The number of DMA transfers should be n x one block. (n = integer, one block = the transfer data size set in SD_SIZE)
- When the IOABT bit in SDIO_MODE is set to 1, the SD_BUF write DMA transfer request is negated.
- The DMA transfer request is also negated by clearing the DMAEN bit to 0. However, the DMA transfer request is asserted again when the DMAEN bit is set to 1 before writing to SD_CMD.
- Since the BWE bit in SD_INFO2 is not cleared in response to setting the STP/IOABT bit, or to a communication error or timeout, clear the bit to 0 before issuing the next command. The next request to write to SD_BUF by DMA transfer will not be issued while the BWE bit is set.

(2) SD_BUF read DMA transfer request

- When the BRE bit in SD_INFO2 is set to 1 while the DMAEN bit in the SD_DMAEN register is set to 1, the SD_BUF read DMA transfer request is asserted
- The SD_BUF read DMA transfer request is negated when the last data in one block (according to the transfer data size set in SD_SIZE) is transferred. The SD_BUF read DMA transfer request is also negated by clearing the SDRST bit in SOFT_RST to 0 or setting the STP bit in SD_STOP to 1. If a communication error or timeout occurs at the DMA transfer, the SD_BUF read DMA transfer request is not negated.
- The BRE bit in SD_INFO2 is cleared after transfer of the last data in one block following a request to write to SD_BUF by DMA transfer
- The number of DMA transfers should be n x one block. (n = integer, one block = the transfer data size set in SD_SIZE)
- When the IOABT bit in SDIO_MODE is set to 1, the SD_BUF read DMA transfer request is negated
- The DMA transfer request is also negated by clearing the DMAEN bit to 0. However, the DMA transfer request is asserted again when the DMAEN bit is set to 1 before writing to SD_CMD.
- Since the BRE bit in SD_INFO2 is not cleared in response to setting the STP/IOABT bit or in response to a communication error or timeout, clear the bit to 0 before issuing the next command. The next request to write to SD_BUF by DMA transfer is not issued while the BRE bit is set.

37.3.4 Communication Errors and Timeouts

When a communication error or timeout error occurs, depending on the type of error, the corresponding status flag in the SD_INFO2 register becomes 1. Also, depending on the source of the error, the associated flag in the SD_ERR_STS1 or SD_ERR_STS2 register becomes 1.

The status flags in registers SD_ERR_STS1 and SD_ERR_STS2 become 0 by writing to the SD_CMD register, or by setting the SOFT_RST.SDRST bit to 0.

Table 37.6 Communication errors

Communication error	Interrupt flag register		Error status register		This occurs when...
	Register symbol	Bit symbol	Register symbol	Bit symbol	
End bit error	SD_INFO2	ENDE	SD_ERR_STS1	CRCLENE	The CRC status token length is in error
				RDLENE	The read data length is in error
				RSPLENE1	The response length is in error*1
				RSPLENE0	The response length is in error*2
CRC error		CRCE		CRCTKE	The CRC status token is in error
				RDCRCE	There is a CRC error in the read data
				RSPCRCE1	There is a CRC error in the response*1
				RSPCRCE0	There is a CRC error in the response*2
Command error		CMDE		CMDE1	The command index field value for the transmitted command and received response do not match*1
				CMDE0	The command index field value for the transmitted command and received response do not match*2

Note 1. CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Note 2. CMD other than CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Table 37.7 Timeouts

Timeout	Interrupt flag register		Error status register		This occurs when...
	Register symbol	Bit symbol	Register symbol	Bit symbol	
Response timeout	SD_INFO2	RSPTO	SD_ERR_STS2	RSPTO1	A response is not received even after a minimum of 640 SDHI clock cycles elapse*1
				RSPTO0	A response is not received even after a minimum of 640 SDHI clock cycles elapse*2
Data timeout (excluding response timeout)		DTO		CRCBSYTO	After the CRC status token is received, the SDHI is busy for at least the period set*3
				CRCTO	After the write data is transmitted, the CRC status token is not received even after at least the period set*3 elapses
				RDTO	After the read command is issued, the read data is not received even after at least the period set*3 elapses
					After the read data is received, the next block read data is not received even after at least the period set*3 elapses
					After the SDHI exits the read wait state, the next block read data is not received even after at least the period set*3 elapses
				BSYTO1	After CMD12 is issued during the command sequence, the SDHI is busy for at least the period set*3
				BSYTO0	After the R1b response is received, the SDHI is busy for at least the period set*3 (a command other than CMD12 is issued during the command sequence)

Note 1. CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Note 2. CMD other than CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

Note 3. The period is set in the SD_OPTION.TOP[3:0] bits.

37.3.5 Command without Data Transfer [SD/MMC]

Figure 37.8 and Figure 37.9 show example flows.

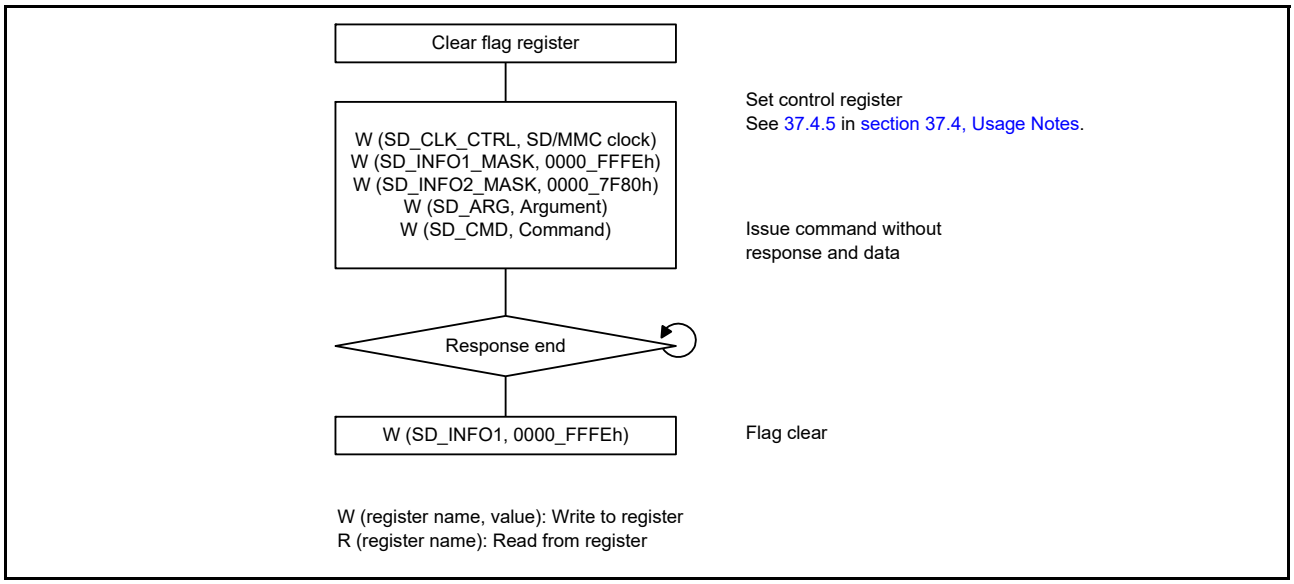


Figure 37.8 Example flow of command without response and data

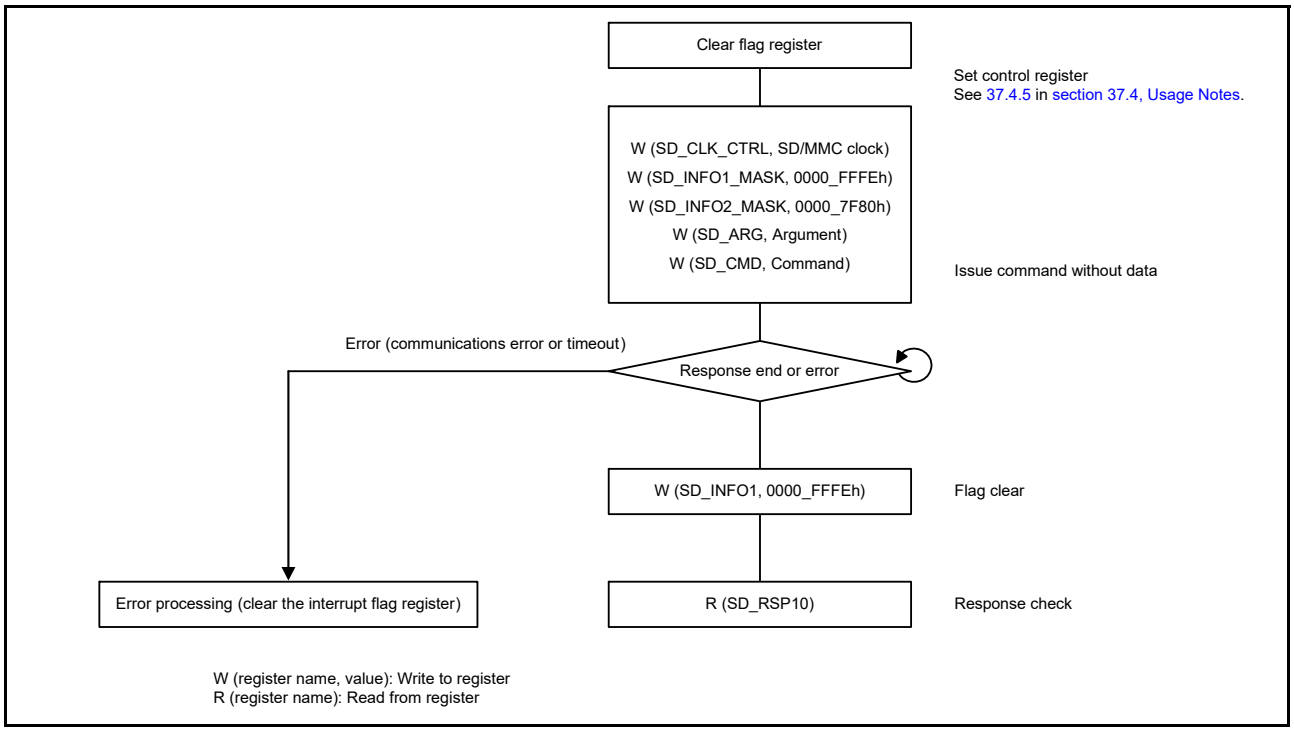


Figure 37.9 Example flow of command without data

37.3.5.1 Operation for Command without Data Transfer

The following legend is used for description of register read/write.

W (register name, value): Write to register

R (register name): Read from register

The operation is described in the following section.

(1) Command without response and data

a. Flag register clear
First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)

b. Control register set
Set the SD/MMC clock, interrupt mask, and so on. (SD_CLK_CTRL, SD_INFO1_MASK, and SD_INFO2_MASK)

c. Command issue
Set CMD argument in SD_ARG and write to SD_CMD.
Accordingly, CMD is issued, and the operation is started.

d. Flag clear
When transmission of a command is completed, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0.

(2) Command without data

a. Flag register clear
First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)

b. Control register set
Set the SD/MMC clock, interrupt mask, and so on. (SD_CLK_CTRL, SD_INFO1_MASK, and SD_INFO2_MASK)

c. Command issue
Set CMD argument in SD_ARG and write to the SD_CMD.
Accordingly, CMD is issued, and the operation is started.

d. Flag clear
When a response is received, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0.

e. Read a response from SD_RSP10. Additionally, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

37.3.6 Single Block Read [SD/MMC]

Figure 37.10 shows an example flow of a single block read operation.

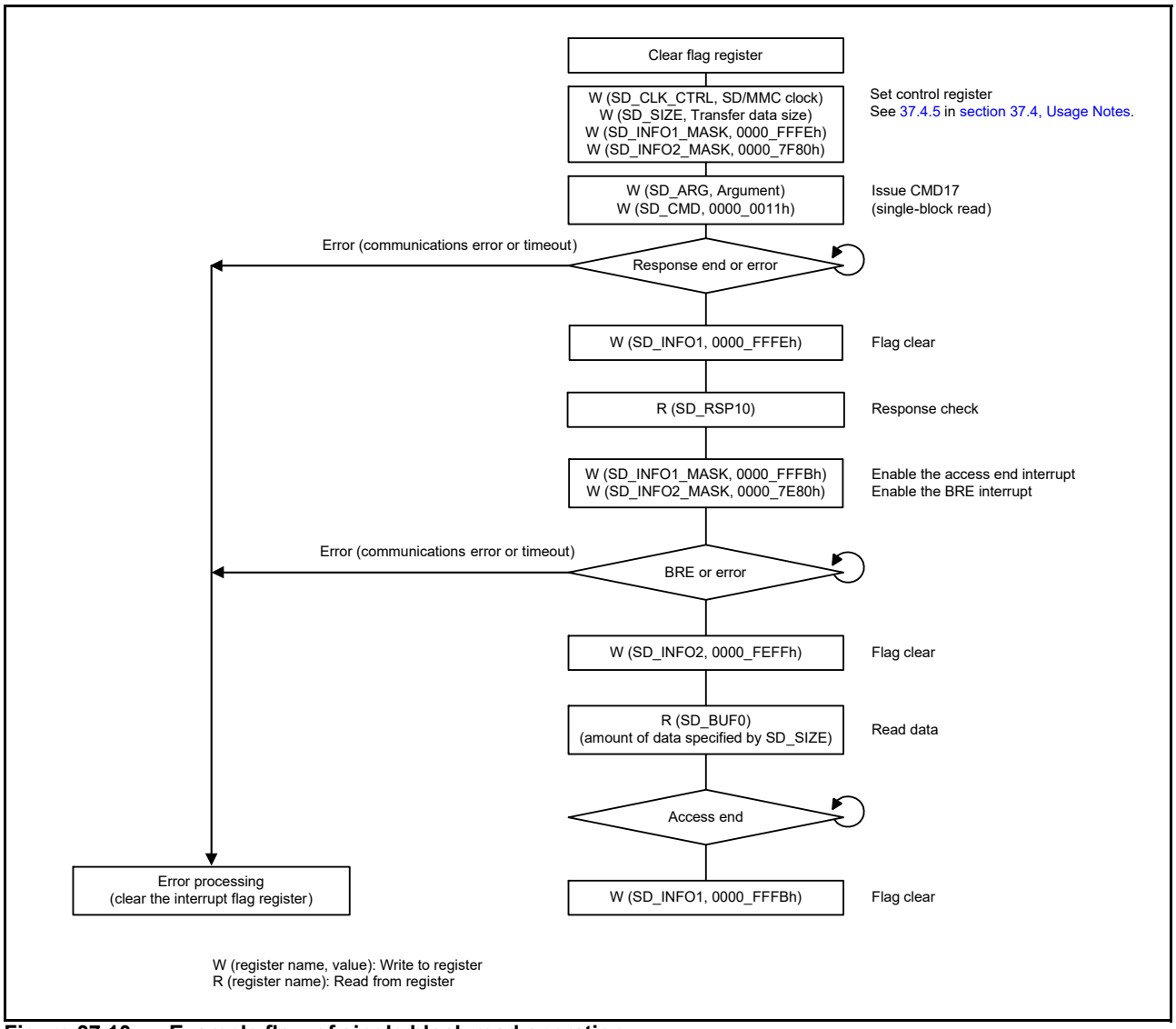


Figure 37.10 Example flow of single block read operation

37.3.6.1 Single Block Read Operation

The operation of the single block read is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the SD/MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).

c. Command issue (CMD17)
Set CMD17 argument in SD_ARG and write 0000_0011h to SD_CMD. CMD17 is issued and the single block read operation is started.

d. Response check
On receiving the response, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0 and read the response from SD_RSP10. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP or the IOABT bit in SDIO_MODE to 1. In addition, this causes CMD12 and CMD52 to not be issued. If the ACEND bit (access end) in SD_INFO is set, halting the command sequence also leads to the generation of an interrupt.

e. Data receive from SD card/MMC and data read
Write 0000_FFFBh to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0000_7E80h to SD_INFO2_MASK to enable the BRE interrupt. When the data received from the SD card/MMC is completed, the BRE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BRE bit to 0 and read the amount of data specified by SD_SIZE from the SD_BUF0 register.
A communication error or timeout might be generated if data are being received while reading the SD_BUF0 register.

f. Operation complete
When the data read from the SD_BUF0 register is completed, ACEND (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear ACEND to 0 to end the single block read operation.
Additionally, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

37.3.7 Single Block Write [SD/MMC]

Figure 37.11 shows an example flow of a single block write operation.

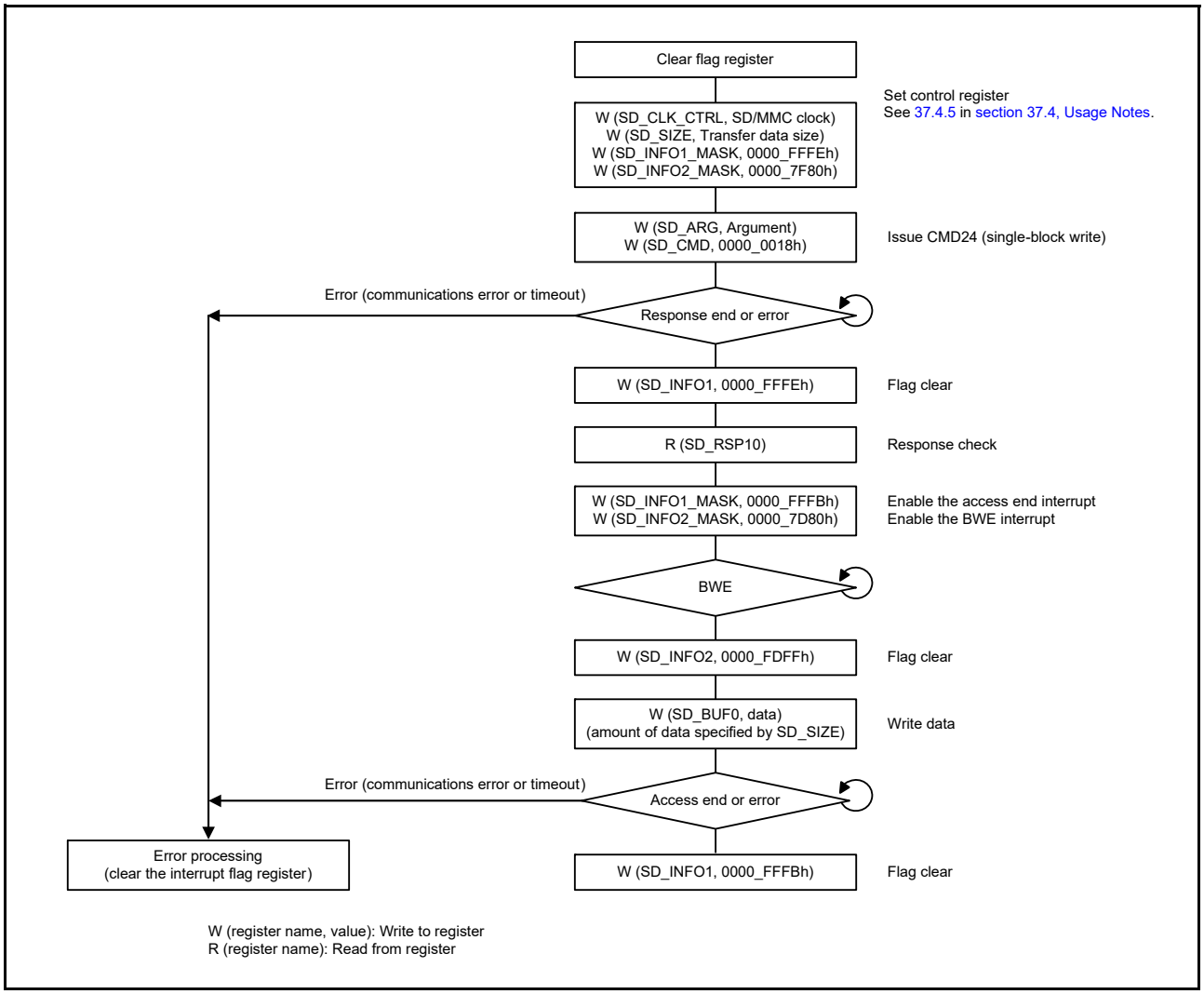


Figure 37.11 Example flow of single block write operation

37.3.7.1 Single Block Write Operation

The operation of the single block write is:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the SD/MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).

c. Command issue (CMD24)
Set CMD24 argument in SD_ARG and write 0000_0018h to SD_CMD. CMD24 is issued and the single block write operation is started.

d. Response check
On receiving the response, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0 and read the response from SD_RSP10. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP or the IOABT bit in SDIO_MODE to 1. In addition, this causes CMD12 and CMD52 to not be issued. If the ACEND bit (access end) in SD_INFO is set, halting the command sequence also leads to the generation of an interrupt.

e. Data write and data transmit to SD card/MMC
Write 0000_FFFBh to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0000_7D80h to SD_INFO2_MASK to enable the BWE interrupt. When the SD_BUF0 register is ready for the data to be written, the BWE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to the SD_BUF0 register. When the data write to the SD_BUF0 register is completed, data is transmitted to the SD card. Then, the CRC status and busy state are received from the SD card/MMC.
However, a communication error or timeout may be generated if data are being transmitted after writing to the SD_BUF0 register.

f. Operation complete
When the CRC status and busy state are received from the SD card/MMC, ACEND (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the ACEND bit to 0 to end the single block write operation.
In addition, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

37.3.8 Multiple Block Read [SD/MMC]

Figure 37.12 shows an example flow of a multiple block read operation.

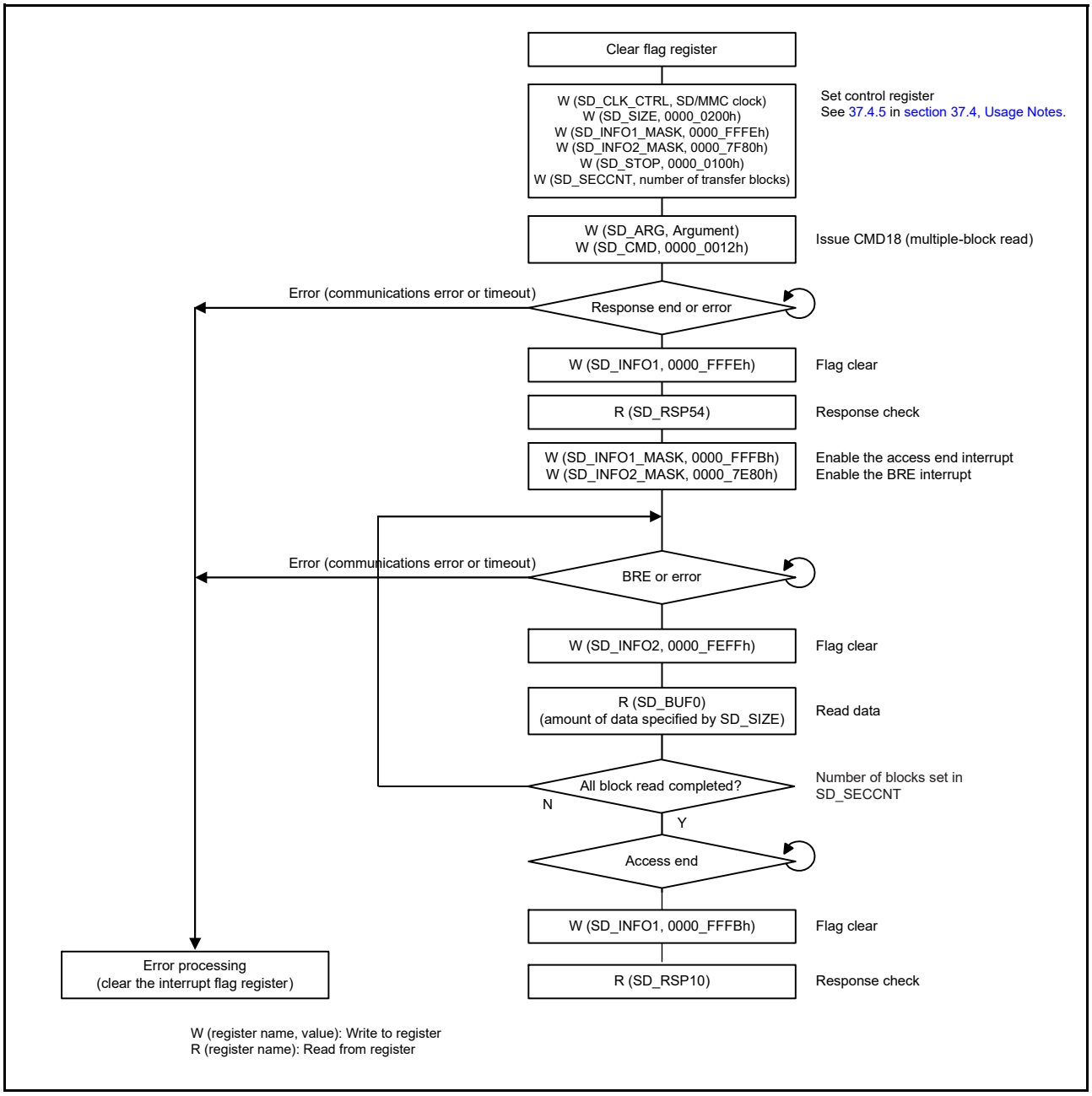


Figure 37.12 Example flow of multiple block read operation

37.3.8.1 Multiple Block Read Operation

The operation of the multiple block read is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the SD/MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).
Set SEC in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

c. Command issue (CMD18)
Set CMD18 argument in SD_ARG and write 0000_0012h to SD_CMD. CMD18 is issued and the multiple block read operation is started.

d. Response check
On receiving the response, RSPEND (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear RSPEND to 0 and read the response from SD_RSP54. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt is enabled, an interrupt is generated by setting of the ACEND bit (access end) bit in SD_INFO1 to 1 when reception of the response is complete. Clear the ACEND bit to 0 and read the response.

e. Data receive from SD card/MMC and data read
Write 0000_FFFBh to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0000_7E80h to SD_INFO2_MASK to enable the BRE interrupt. When one-block data received from the SD card/MMC is completed, the BRE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BRE bit to 0 and read the amount of data specified by SD_SIZE from the SD_BUF0 register. Doing this repeats transfer of the number of blocks set in SD_SECCNT. However, a communication error or timeout might be generated if data are being received while reading of the SD_BUF0 register is in progress. CMD12 is automatically issued to stop multi-block transfer with the number of blocks that is set to SD_SECCNT and the response is received. At this point, CMD12 argument is automatically set to 0000_0000h.

f. Operation complete
When all-block data read and the CMD12 response received are completed, ACEND (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear ACEND to 0 to read the response. This is the end of multiple block read operation. In addition, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

37.3.9 Multiple Block Write (SD/MMC using internal timer)

Figure 37.13 shows an example flow of a multiple block write using internal timer.

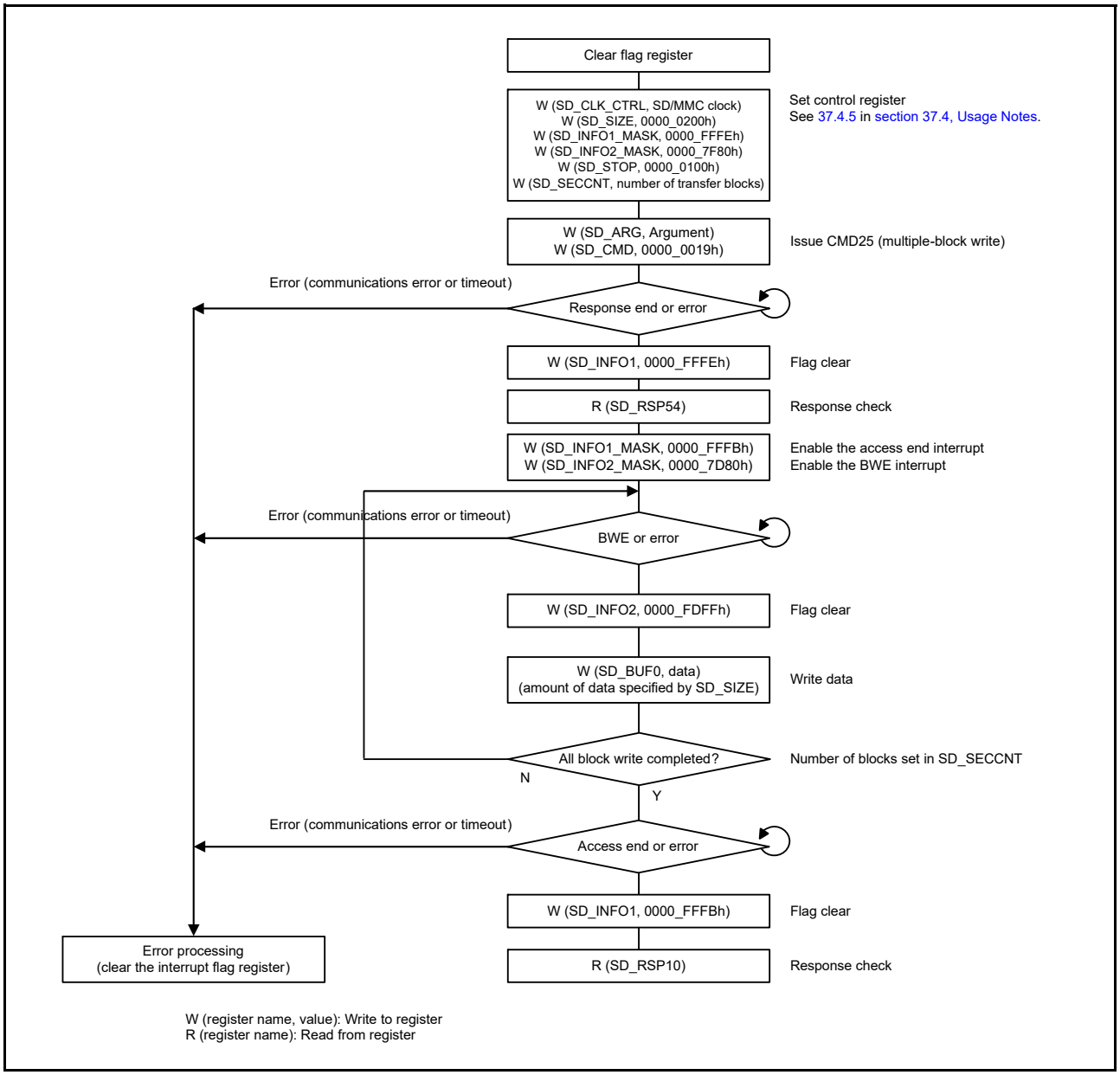


Figure 37.13 Example flow of multiple block write operation using internal timer

37.3.9.1 Multiple Block Write Operation using internal timer

The operation of the multiple block write is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the SD/MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).
Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

c. Command issue (CMD25)
Set CMD25 argument in SD_ARG and write 0000_0019h to SD_CMD. CMD25 is issued and the multiple block write operation is started.

d. Response check
On receiving the response, the RSPEND bit (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the RSPEND bit to 0 and read the response from SD_RSP54. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt is enabled, an interrupt is generated by setting of the ACEND bit (access end) bit in SD_INFO1 to 1 when reception of the response is complete. Clear the ACEND bit to 0 and read the response.

e. Data write and data transmit to SD card/MMC
Write 0000_FFFBh to SD_INFO1_MASK to enable the access end interrupt. In addition, write 0000_7D80h to SD_INFO2_MASK to enable the BWE interrupt. When the SD_BUF0 register is ready for the data to be written, the BWE bit in the SD_INFO2 resister is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to the SD_BUF0 register. When the data write to the SD_BUF0 register is completed, data is transmitted to the SD card/MMC. The CRC status and busy state are received from the SD card/MMC. This repeats transfer of the number of blocks set in SD_SECCNT. However, a communication error or timeout might be generated if data are being received while writing to the SD_BUF0 register is in progress. CMD12 is automatically issued to stop multi-block transfer with the number of blocks which is set to SD_SECCNT and the response is received. At this point, CMD12 argument is automatically set to 0000_0000h.

f. Operation complete
When all-block data transmit and the CRC status receive are completed, the ACEND bit (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the ACEND bit to 0 to read the response. This is the end of multiple block write operation. Additionally, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs.

37.3.10 Multiple Block Write (MMC using external timer)

Figure 37.14 shows an example flow of a multiple block write using an external timer.

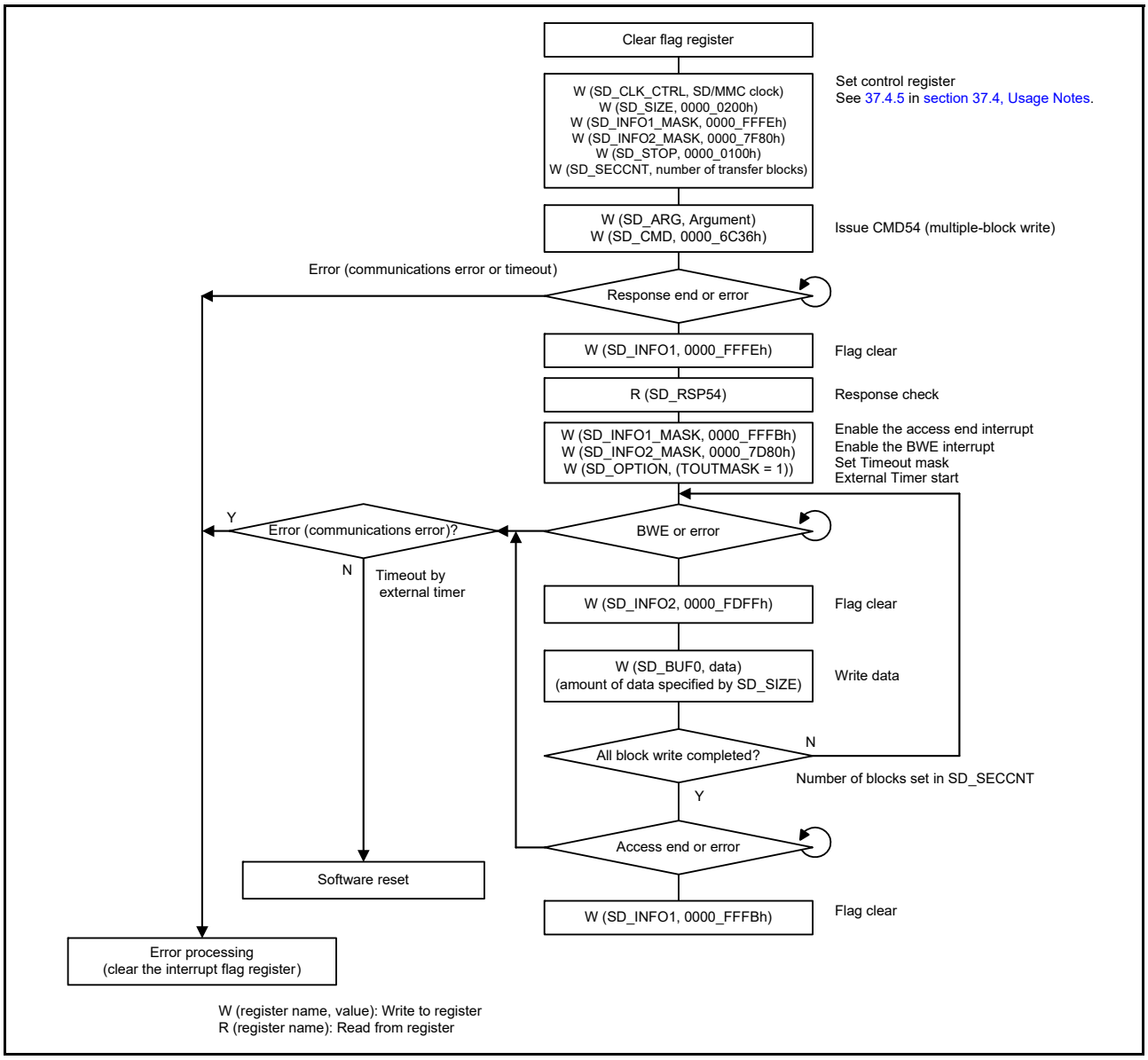


Figure 37.14 Example flow of multiple block write operation using external timer

37.3.10.1 Multiple Block Write Operation using external timer

The operation of the multiple block write is described as follows:

a. Flag register clear
First, clear the bits in the flag register (SD_INFO1 and SD_INFO2).

b. Control register set
Set the MMC clock, transfer data size, interrupt mask (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK).
Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.

c. Command issue (CMD54)
Set CMD54 Argument in SD_ARG and write 0000_6C36h to SD_CMD. CMD54 is issued and the multiple block write operation is started.

d. Response check
On receiving the response, the RSPEND bit (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the RSPEND bit to 0 and read the response from SD_RSP54. If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt is enabled, an interrupt is generated by setting of the ACEND bit (access end) bit in SD_INFO1 to 1 when reception of the response is complete. Clear the ACEND bit to 0 and read the response.

e. Data write and data transmit to MMC
Write 0000_FFFBh to SD_INFO1_MASK to enable the access end interrupt, write 0000_7D80h to SD_INFO2_MASK to enable the BWE interrupt and set 1 to TOUTMASK of SD_OPTION to deactivate timeout. In addition, start the external timer. When the SD_BUF0 register is ready for the data to be written, the BWE bit in the SD_INFO2 resister is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to the SD_BUF0 register. When the data write to the SD_BUF0 register is completed, data is transmitted to the MMC. The CRC status and busy state are received from the MMC. Doing this repeats transfer of the number of blocks set in SD_SECCNT. However, a communication error or timeout might be generated if data are being received while writing to the SD_BUF0 register is in progress.

f. Operation complete
When all-block data transmit and the CRC status receive are completed, the ACEND bit (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the ACEND bit to 0 to read the response. This is the end of multiple block write operation. Additionally, perform error processing (clear the interrupt flag register) if a communication error or timeout occurs when receiving response. Execute software reset if a timeout by external timer occurs when transmitting data.

37.3.11 IO_RW_DIRECT Command (SD: CMD52)

Figure 37.15 shows an example flow of an IO_DIRECT command (CMD52) operation.

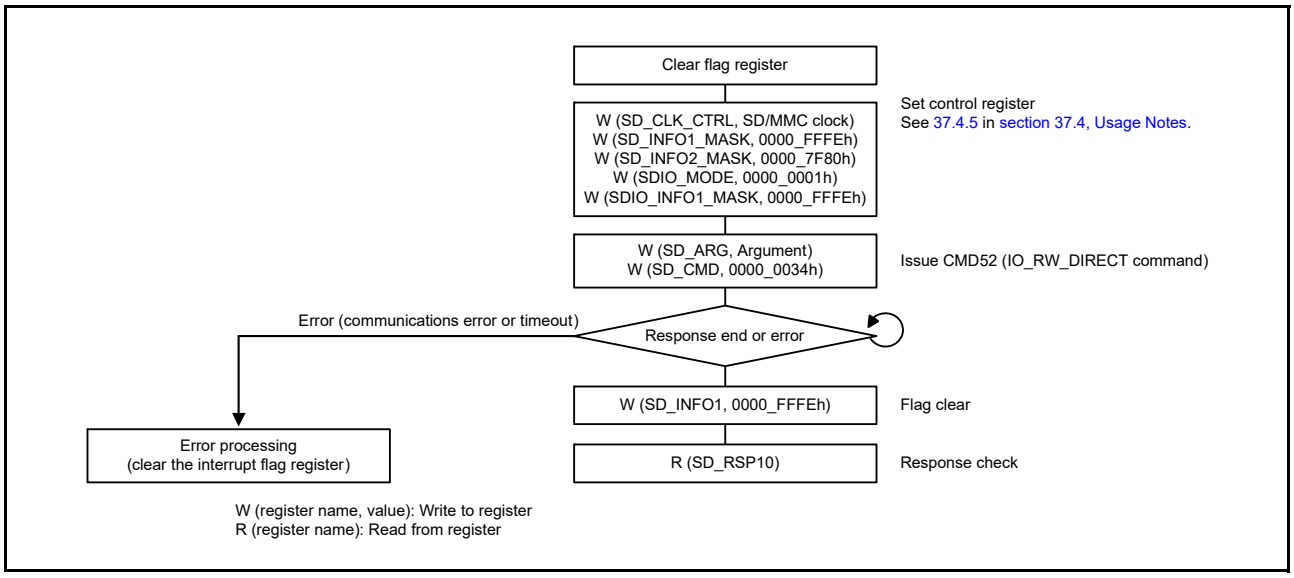


Figure 37.15 Example of IO_RW_DIRECT command (CMD52) operation

37.3.12 IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Read)

Figure 37.16 shows an example flow for a CMD53 multiple block read operation.

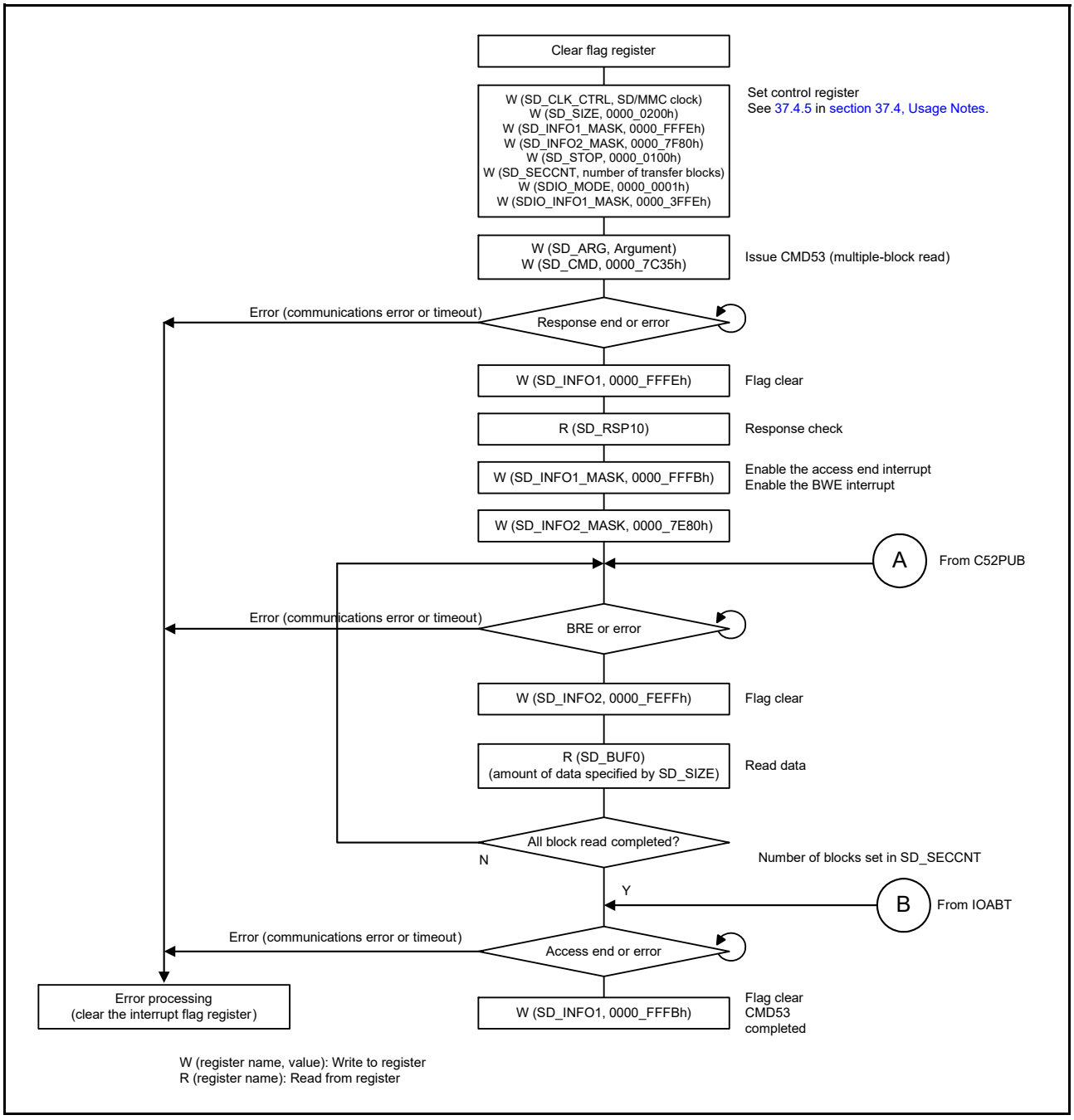


Figure 37.16 Example of IO_RW_EXTENDED command (CMD53) for multiple block read operation

Figure 37.17 shows an example flow when CMD52 (SDIO abort) is issued at a CMD53 multiple block read.

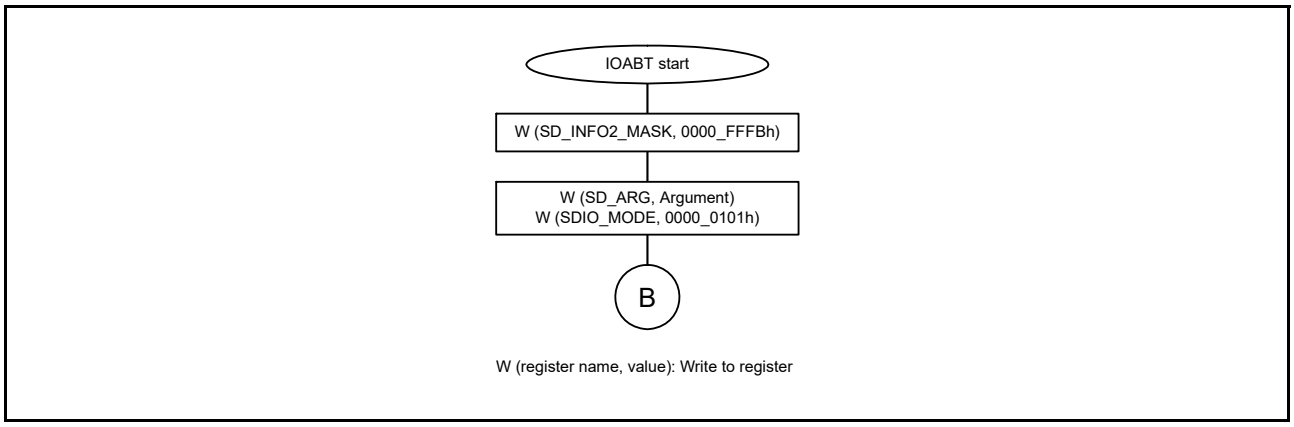


Figure 37.17 CMD52 (SDIO Abort) is issued at CMD53 multiple block read

Figure 37.18 shows an example flow when CMD52 (SDIO none abort) is issued at a CMD53 multiple block read while the SD host interface is in the read wait state.

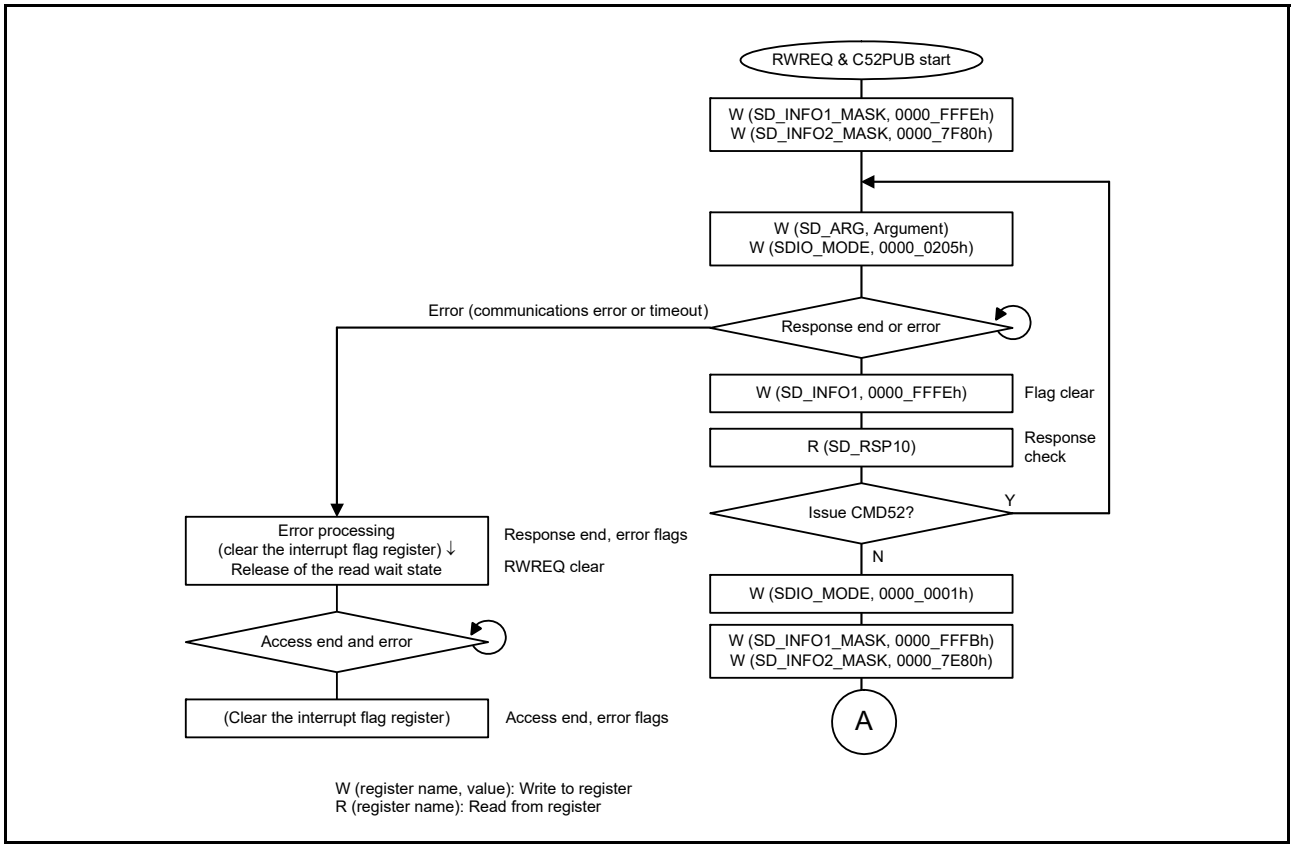


Figure 37.18 CMD52 (SDIO None Abort) is issued at CMD53 multiple block read while SD Host Interface is in read wait state

37.3.13 IO_RW_EXTENDED Command (SD: CMD53/Multiple Block Write)

Figure 37.19 shows an example flow for a CMD53 multiple block write.

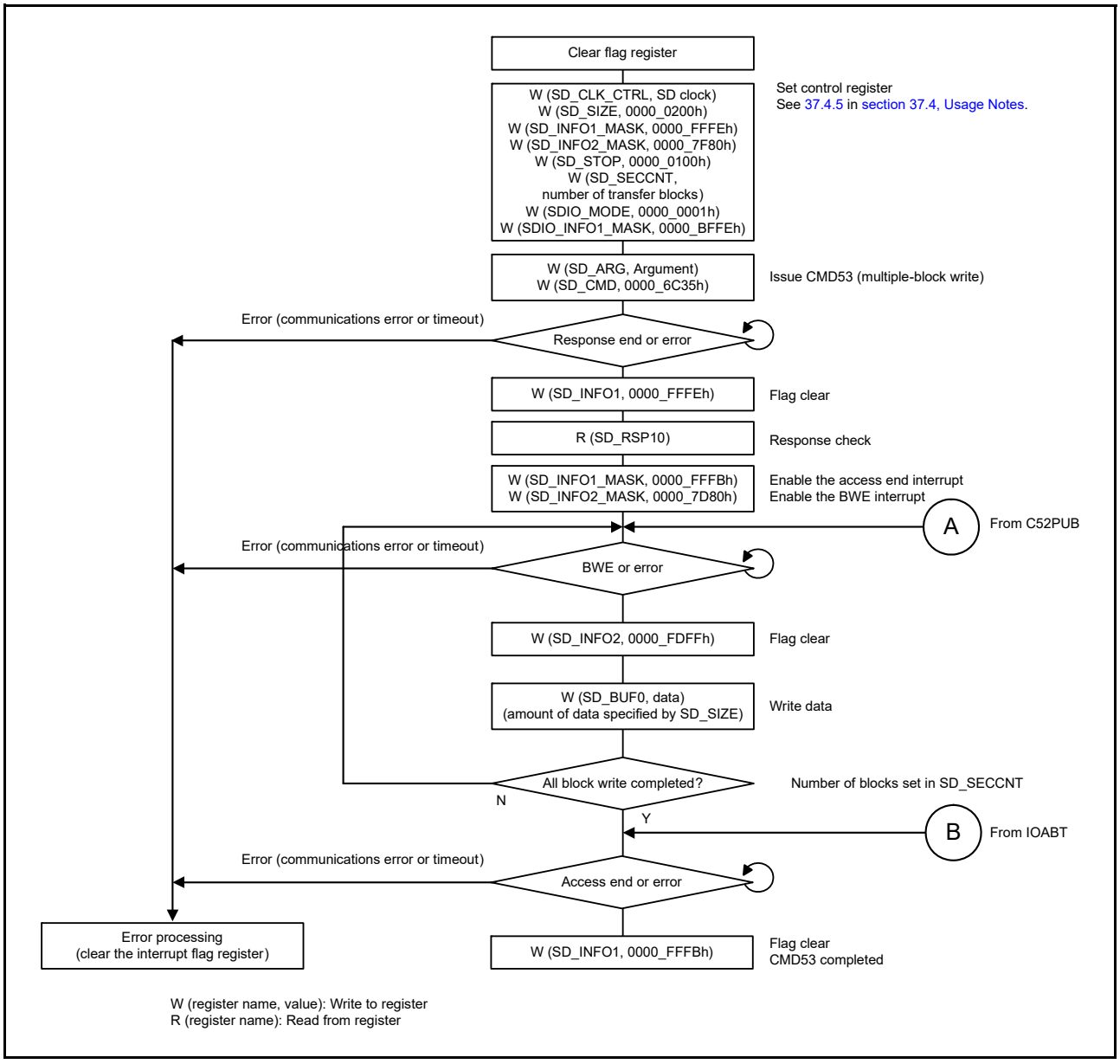


Figure 37.19 Example of IO_RW_EXTENDED command for a CMD53 multiple block write operation

Figure 37.20 shows an example flow when CMD52 (SDIO abort) is issued for a CMD53 multiple block write.

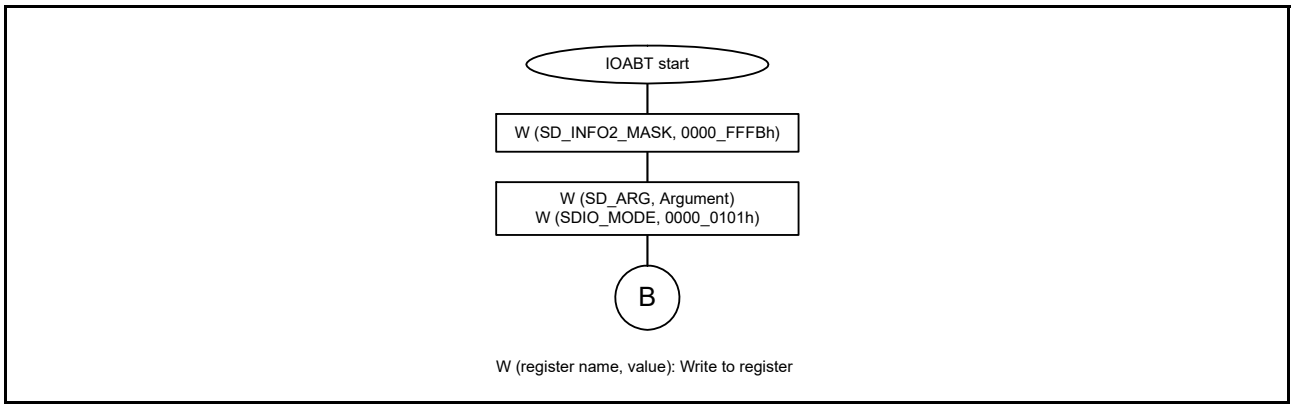


Figure 37.20 CMD52 (SDIO Abort) is issued for a CMD53 multiple block write

Figure 37.21 shows an example flow when CMD52 (SDIO none abort) is issued at CMD53 multiple block write.

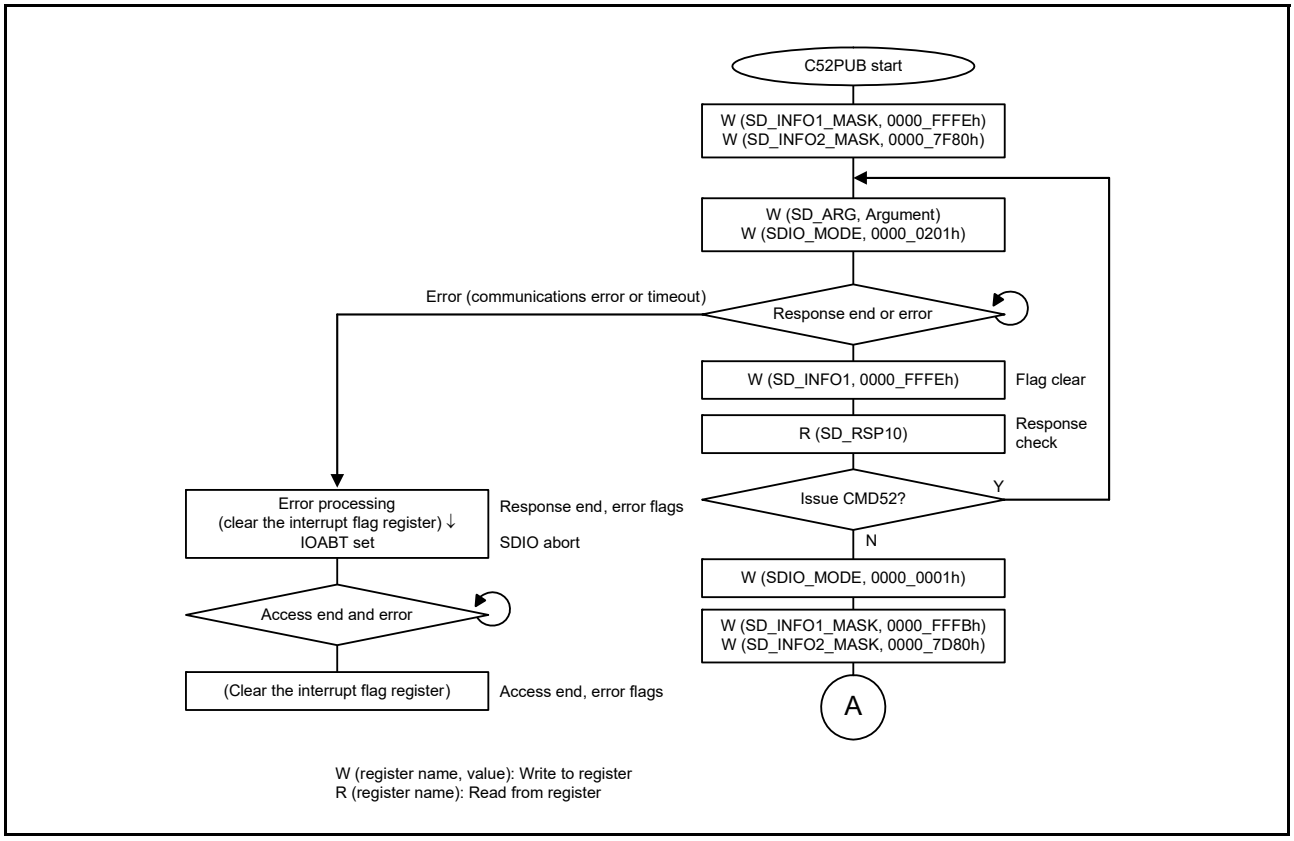


Figure 37.21 CMD52 (SDIO None Abort) is issued for a CMD53 multiple block write

37.3.14 DMA Transfer [SD/MMC]

37.3.14.1 SD_BUF DMA Transfer

Figure 37.22 shows an example flow for SD_BUF DMA read when CMD18 multiple block read is issued.

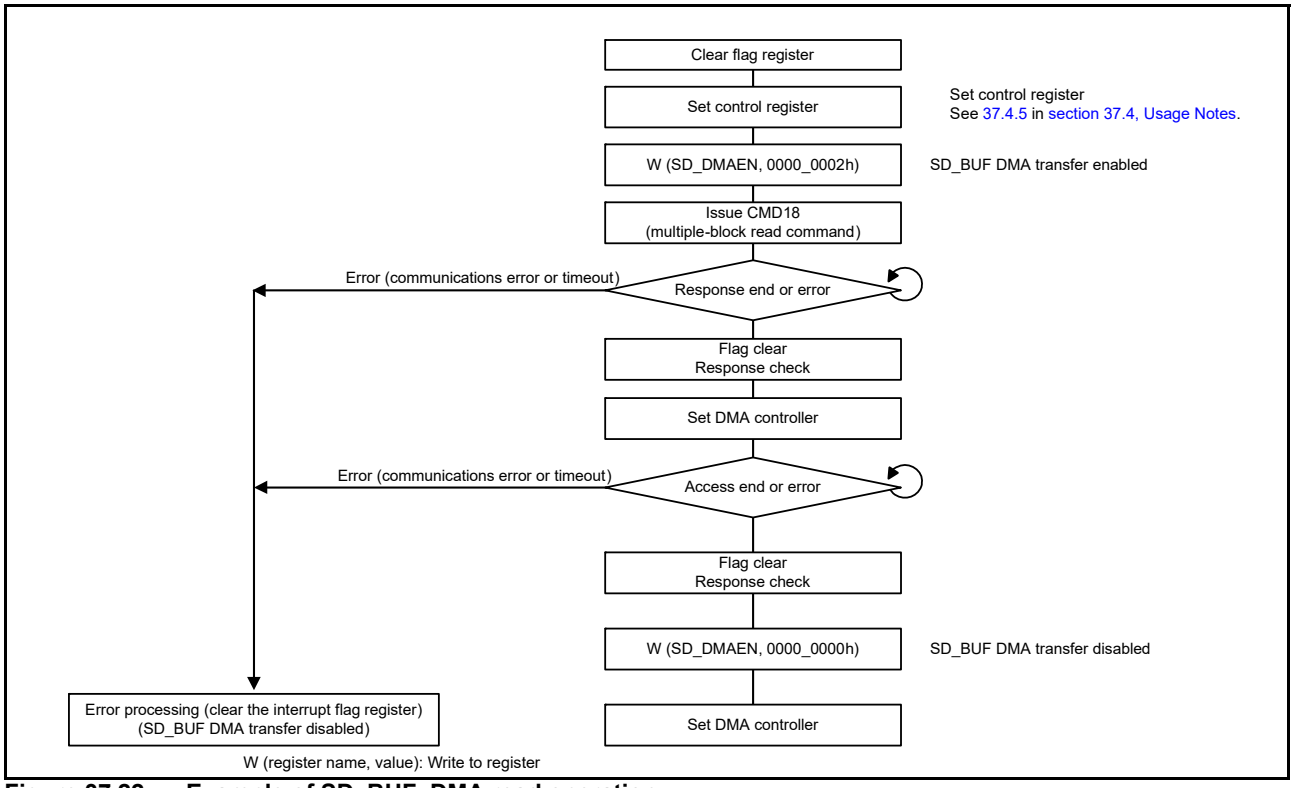


Figure 37.22 Example of SD_BUF_DMA read operation

Figure 37.23 shows an example flow for SD_BUF DMA write when CMD25 multiple block write is issued.

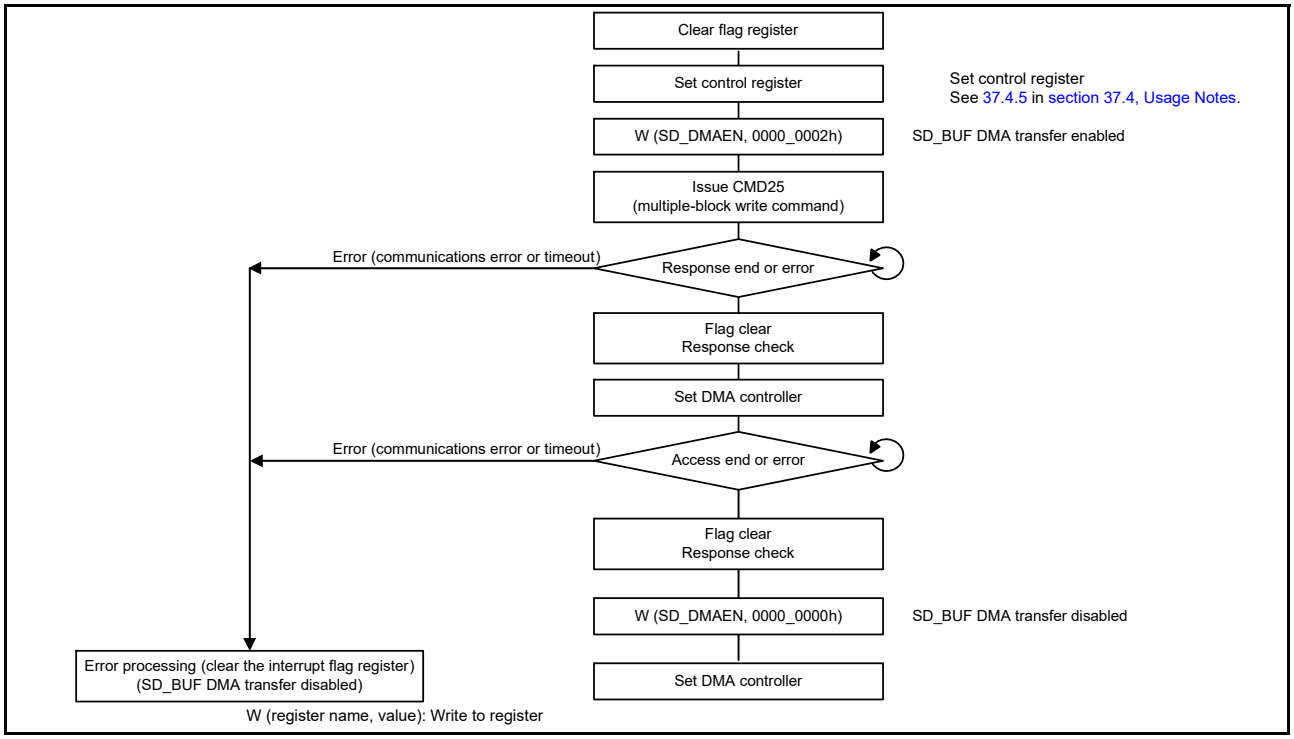


Figure 37.23 Example of SD_BUF_DMA write operation

37.3.15 Example of SD_CMD Register Setting

Table 37.8 and Table 37.9 list the SD_CMD register setting.

Table 37.8 Example of SD_CMD register setting [SD] (1 of 2)

Type	Commamd	Example of SD_CMD Register Setting	Remark
CMD	CMD0	0000_0000h	-
	CMD2	0000_0002h	-
	CMD3	0000_0003h	-
	CMD4	0000_0004h	-
	CMD5	0000_0705h or 0000_0005h	-
	CMD6	0000_1C06h or 0000_0006h	-
	CMD7	0000_0007h	When the card is placed in the deselected state, the response timeout flag is set since there is no response
	CMD8	0000_0408h or 0000_0008h	-
	CMD9	0000_0009h	-
	CMD10	0000_000Ah	-
	CMD11	0000_040Bh or 0000_000Bh	-
	CMD12	0000_000Ch	-
	CMD13	0000_000Dh	-
	CMD15	0000_000Fh	-
	CMD16	0000_0010h	-
	CMD17	0000_0011h	-
	CMD18	0000_0012h	With automatic CMD12
	CMD20	0000_0514h or 0000_0014h	-
	CMD24	0000_0018h	-
	CMD25	0000_0019h	With automatic CMD12
	CMD27	0000_001Bh	-
	CMD28	0000_001Ch	-
	CMD29	0000_001Dh	-
	CMD30	0000_001Eh	-
	CMD32	0000_0020h	-
	CMD33	0000_0021h	-
	CMD38	0000_0026h	-
	CMD42	0000_002Ah	-
	CMD52	0000_0434h or 0000_0034h	-
	CMD53	0000_1C35h	Single read
		0000_0C35h	Single write
		0000_7C35h	Multiple read
		0000_6C35h	Multiple write
		0000_0035h	The value on the left can be set irrespective of whether single or multi. However, the CF39 bit in SD_ARG must be set as follows: Read = 0 Write = 1
	CMD55	0000_0037h	-
	CMD56	0000_0038h	-

Table 37.8 Example of SD_CMD register setting [SD] (2 of 2)

Type	Commamd	Example of SD_CMD Register Setting	Remark
ACMD	ACMD6	0000_0046h	-
	ACMD13	0000_004Dh	-
	ACMD22	0000_0056h	-
	ACMD23	0000_0057h	-
	ACMD41	0000_0069h	-
	ACMD42	0000_006Ah	-
	ACMD51	0000_0073h	-

Table 37.9 Example of SD_CMD Rregister setting (MMC) (1 of 2)

Type	Command	Example of SD_CMD Register Setting	Remark
CMD	CMD0	0000_0000h	-
	CMD1	0000_0701h	-
	CMD2	0000_0002h	-
	CMD3	0000_0003h	-
	CMD4	0000_0004h	-
	CMD5	0000_0505h	-
	CMD6	0000_0506h	With response busy
		0000_0406h	Without response busy.
	CMD7	0000_0007h	When the card is placed in the deselected state, the response timeout flag is set since there is no response
	CMD8	0000_1C08h	-
	CMD9	0000_0009h	-
	CMD10	0000_000Ah	-
	CMD12	0000_000Ch	-
	CMD13	0000_000Dh	-
	CMD14	0000_1C0Eh	Must set SD_IFMODE = 0000_0100h (CRC check is invalid)
	CMD15	0000_000Fh	-
	CMD16	0000_0010h	-
	CMD17	0000_0011h	-
	CMD18	0000_7C12h	Pre-defined
	CMD19	0000_0C13h	Must set SD_IFMODE = 0000_0100h (CRC check is invalid)
	CMD21	0000_1C15h	DDR mode is inhibited
	CMD23	0000_0017h	-
	CMD24	0000_0018h	-
	CMD25	0000_6C19h	Pre-defined
	CMD26	0000_0C1Ah	-
	CMD27	0000_001Bh	-
	CMD28	0000_001Ch	-
	CMD29	0000_001Dh	-
	CMD30	0000_001Eh	-
	CMD31	0000_1C1Fh	-

Table 37.9 Example of SD_CMD Rregister setting (MMC) (2 of 2)

Type	Command	Example of SD_CMD Register Setting	Remark
CMD	CMD35	0000_0423h	-
	CMD36	0000_0424h	-
	CMD38	0000_0026h	-
	CMD39	0000_0427h	-
	CMD40	0000_0428h	-
	CMD42	0000_002Ah	-
	CMD49	0000_0C31h	-
	CMD53	0000_7C35h	-
	CMD54	0000_6C36h	-
	CMD55	0000_0037h	-
	CMD56	0000_0038h	-

37.4 Usage Notes

37.4.1 SD_BUF Illegal Write Access [SD/MMC]

When writing data to the SD_BUF0 register after the single block write or multi-block write command is issued, the data of the size specified by SD_SIZE must be written.

If the data exceeds the size specified by SD_SIZE is written, the ERR4 bit in SD_INFO2 is set to 1. In addition, the data written to the SD_BUF0 register might not be transmitted and the SD_CLK_CTRLEN bit in SD_INFO2 is held at the value of 0. If this occurs, clearing the SDRST bit in SOFT_RST to 0 and then restoring its value to 1 clears the SD_CLK_CTRLEN bit to 1.

However, this does not apply to the single byte or three bytes when the SD_SIZE setting is odd, or to the fraction of bytes when the SD_SIZE setting is even (the 2 bytes that are not in a 4-byte unit), because the portion of dummy data writing is regarded as excess data and ignored.

37.4.2 Block Number Constraint for Multiple Block Read [SD]

When performing a multiple block read of one or two blocks, depending on the timing with which the SD card response register is read, the response value might not be read properly. This must be avoided by either of the following countermeasures:

- When receiving one or two blocks of data, use single block reading
- Read the response to CMD18 from SD_RSP54.

37.4.2.1 Mechanism of incorrect reading

Figure 37.24 shows the processing flows of SD host interface (hardware) operation and software operation when a multiple block read is performed on two blocks. As shown in the incorrect operation in Figure 37.24, when an interrupt is generated on reception of the CMD18 response and the timing with which the SD card response register (SD_RSP10) is read by the interrupt is delayed, the data during the CMD12 response reception or the CMD12 response might be read. The problem does not occur for multiple block reads of three or more blocks, because CMD12 is not issued until the block of data is read. The problem also does not occur for multiple block writes, because the CMD25 response is read before the block of data is sent.

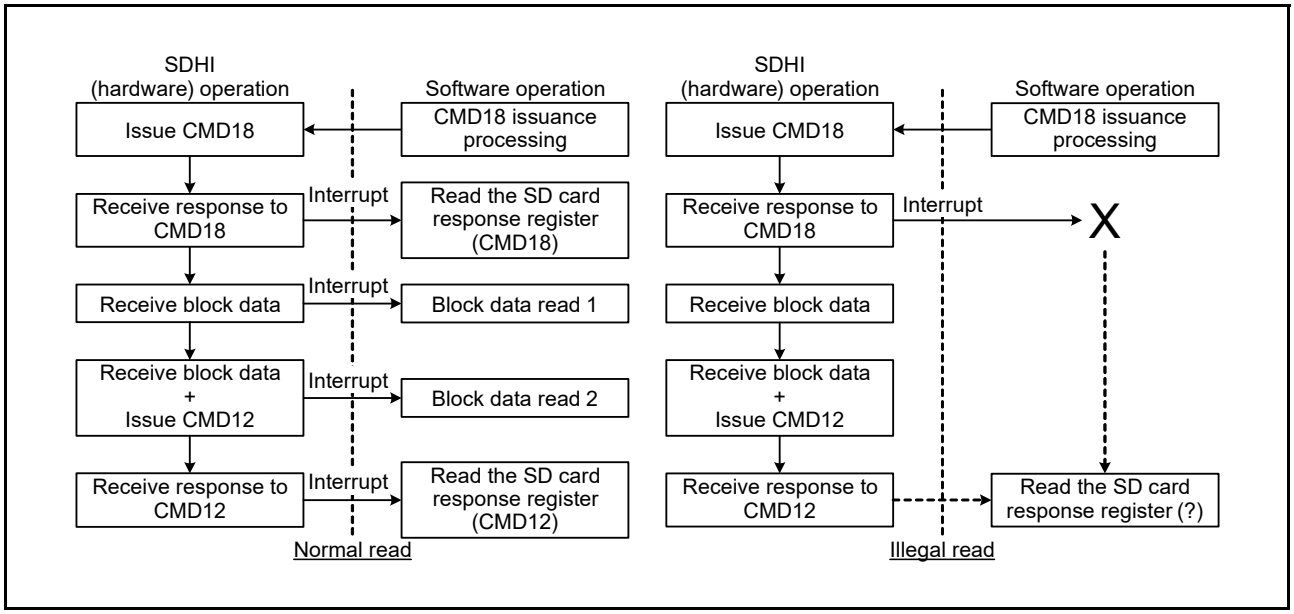


Figure 37.24 Multiple block read operation flow chart (two blocks)

37.4.3 Automatic Control of SD/MMC Clock Output [SD/MMC]

In the SD Card/MMC standard, 74 cycles of SD/MMC clock must be output before initialization of the card. For this reason, use automatic control of SD/MMC clock output after 74 cycles of SD/MMC clock are output. In addition, if automatic control of SD/MMC clock output was in use, SD/MMC clock output is stopped on completion of the sequence for a communication error or timeout. When state transitions within the SD card/MMC are required after completion of the sequence, release automatic control of SD/MMC clock output and restart supply of the SD/MMC clock to the SD card/MMC.

37.4.4 Control of the C52PUB Setting for Multiple Block Write [SD]

If the C52PUB bit in SDIO_MODE is set to 1 during a sequence of multiple block write because of CMD53, CMD52 is not issued until SD_BUF becomes empty. For this reason, set the C52PUB bit after suspending writing to SD_BUF by using one of the following procedures, as appropriate:

(a) When DMA transfer is not in use

1. Before setting the C52PUB bit, suspend writing to SD_BUF by making the setting in SD_INFO2 to disable BWE interrupts.
2. Set the C52PUB bit in SDIO_MODE to 1 (so that CMD52 is issued when SD_BUF becomes empty).
3. After the RSPEND interrupt processing in SD_INFO1 due to the issuing of CMD52 is complete, restart writing to SD_BUF by making the setting in SD_INFO2 to enable BWE interrupts.

(b) When DMA transfer is in use

1. Every time DMA transfer of the value set in SD_SIZE × n blocks (where n = 1, 2, ...) proceeds, suspend writing to SD_BUF by DMA transfer before the C52PUB bit is set.
2. Set the C52PUB bit in SDIO_MODE to 1 (so that CMD52 is issued when SD_BUF becomes empty).
3. After the RSPEND interrupt processing in SD_INFO1 because the issuing of CMD52 is complete, restart writing to SD_BUF by DMA transfer.

37.4.5 Notes on SD_CLK_CTRL Register Settings [SD/MMC]

When the SD_CLK_CTRLEN bit in SD_INFO2 is 0, SD_CLK_CTRL cannot be written to. Before writing to SD_CLK_CTRL, be sure to check that the SD_CLK_CTRLEN bit in SD_INFO2 is 1.

37.4.6 Specification Limitations

1. The Suspend/Resume operation of the SDIO is not supported.
2. The SPI bus is not supported (SD/MMC).
3. The shared bus and 8-bit SD bus of the embedded SDIO are not supported.
4. Stream transfer of MMC is not supported.
5. High Priority Interrupt (HPI) of MMC is not supported.
6. Boot Operation/Alternative Boot Operation of MMC is not supported.
7. Open-ended multiple block transfer of MMC is not supported.

37.4.7 STP Bit Setting during Multiple Block Read [SD/MMC]

During execution of multiple block read with automatic CMD12 execution by setting the SEC bit in SD_STOP to 1, even if the STP bit in SD_STOP is set to 1 to forcibly stop the execution, the command sequence might not stop depending on the timing of setting the STP bit.

To avoid this, when setting the STP bit in SD_STOP to 1 during multiple block transfer, clear the SEC bit in SD_STOP to 0 at the same time. Even when the SD_CLK_CTRLEN bit in SD_INFO2 is 0, change the SEC bit from 1 to 0.

When the command sequence is not stopped because the SEC bit is not set to 0, the command sequence can be stopped by clearing the SDRST bit in SOFT_RST to 0.

When forcibly terminating the CMD53 multiple block transfer through the IOABT bit in SDIO_MODE, be sure to leave the SEC bit in SD_STOP as 1.

37.4.8 Register Setting Notes

1. All registers in section 37.2, Register Descriptions are accessed in 32-bit access-only.
2. When setting registers, set them after the I/O Port Register setting.

38. Boundary Scan

38.1 Overview

The boundary scan function provides a serial I/O interface based on the JTAG (Joint Test Action Group), IEEE Std.1149.1, and IEEE Standard Test Access Port and Boundary Scan Architecture. Table 38.1 lists the boundary scan specifications, Figure 38.1 shows a block diagram, and Table 38.2 lists the I/O pins.

Table 38.1 Specifications of boundary scan

Parameter	Description
Execution condition	Boundary scan must be executed when the RES pin is driven low
Six test modes	• BYPASS mode • EXTEST mode • SAMPLE/PRELOAD mode • CLAMP mode • HIGHZ mode • IDCODE mode.

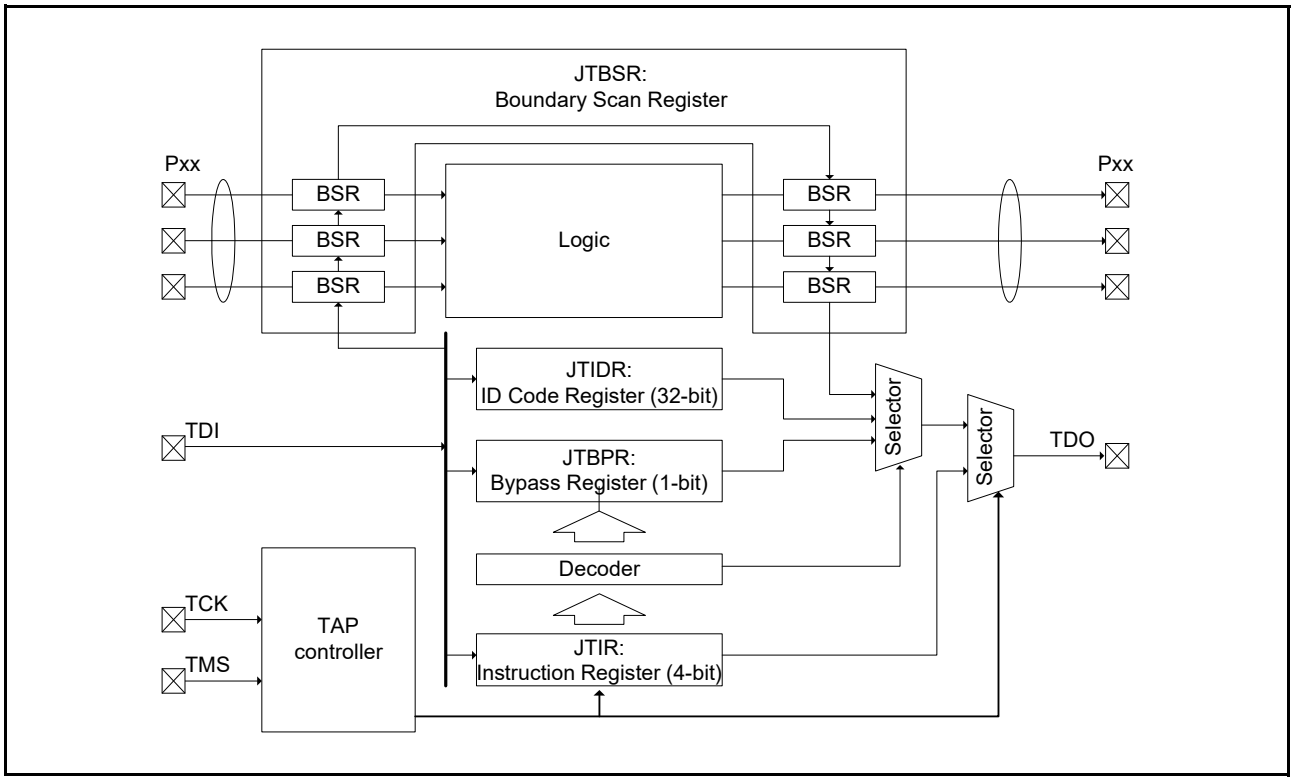


Figure 38.1 Boundary scan function block diagram

Table 38.2 Pin configuration

Pin Name	I/O	Description
TCK	Input	Test clock input pin Clock signal for boundary scan. The input clock duty cycle is 50% when the boundary scan function is used.
TMS	Input	Test mode select pin
TDI	Input	Test data input pin
TDO	Output	Test data output pin

Note: The MCU does not support the TRST pin for the JTAG interface.

38.2 Register Descriptions

Table 38.3 lists the Boundary Scan Registers.

Table 38.3 Boundary scan registers

Register name	Symbol	Value after reset
Instruction Register	JTIR	Eh
ID Code Register	JTIDR	082D B447h
Bypass Register	JTBPR	Undefined
Boundary scan Register	JTBSR	Undefined

Usage notes for the Boundary Scan Registers:

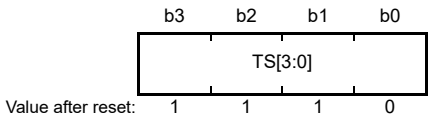
- Instructions can be input to the Instruction Register (JTIR) through the TDI pin by serial transfer.
- The Bypass Register (JTBPR), which is a 1-bit register, is connected between the TDI and TDO pins in BYPASS mode.
- The Boundary Scan Register (JTBSR), which is configured according to the BSDL description, is connected between the TDI and TDO pins when test data is being shifted in.

Table 38.4 shows the availability of serial transfer for the registers.

Table 38.4 Serial transfer for registers

Register name	Serial input	Serial output
Instruction Register (JTIR)	Available	Available
ID Code Register (JTIDR)	Available	Available
Bypass Register (JTBPR)	Available	Available
Boundary Scan Register (JTBSR)	Available	Available

38.2.1 Instruction Register (JTIR)



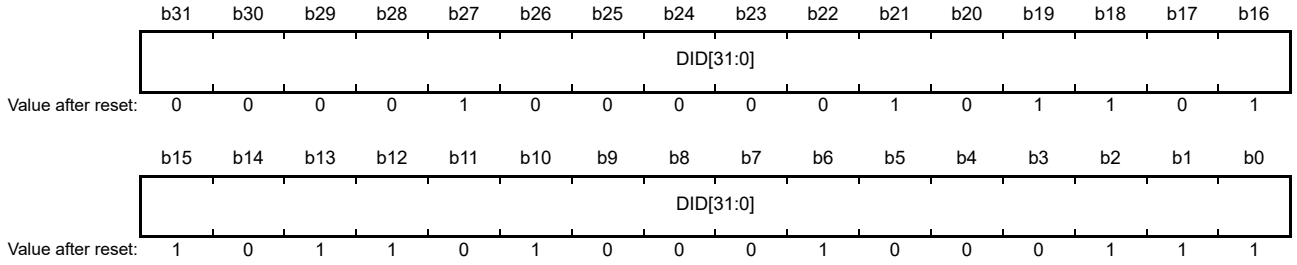
Bit	Symbol	Bit name	Description	R/W
b3 to b0	TS[3:0]	Test Bit Set	The command configuration is shown in Table 38.5	—

Table 38.5 Command configuration

TS3	TS2	TS1	TS0	Instruction
0	0	0	0	EXTEST
0	0	0	1	SAMPLE/PRELOAD
0	0	1	1	IDCODE (Renesas code)
0	1	0	1	CLAMP
0	1	1	0	HIGHZ
1	1	1	1	BYPASS
Other settings				Reserved

JTAG instructions can be transferred to the JTIR register by serial input from the TDI pin. The JTIR register is initialized when a power-on reset occurs, or when the TAP controller is in the Test-Logic-Reset state.

38.2.2 ID Code Register (JTIDR)



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	DID[31:0]	Device ID	These bits store the fixed value that indicates the device IDCODE.	—

JTIDR data is output from the TDO pin when the IDCODE instruction is executed. After a reset release, the IDCODE of JTIDR changes into the Arm® debug code. See *ARM® CoreSight™ SoC-400 Technical Reference Manual (ARM DDI 0480F)*.

38.2.3 Bypass Register (JTBPR)

JTBPR is a 1-bit register connected between the TDI and TDO pins when the JTIR register is set to BYPASS mode.

Note: The JTBPR register cannot be read from or written to by the CPU.

38.2.4 Boundary Scan Register (JTBSR)

JTBSR is a shift register for controlling the external input and output pins of the MCU, and is distributed across the pads. For the JTBSR register in boundary-scan testing, issue the EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ instructions. The BSDL files describe the association between the JTBSR bits and the pins of the MCU. The value after reset is undefined.

38.3 Operations

During a reset, the JTAG ports, TCK, TMS, TDI, and TDO, are assigned as default pin functions. The TCK, TMS, and TDI pins are pulled up by the pull-up resistors. Boundary scan testing can be executed after the setup time elapses when POR is negated and RES is driven low.

38.3.1 TAP Controller

Figure 38.2 shows the state transition diagram of the TAP controller. All transitions are controlled by the TMS signal.

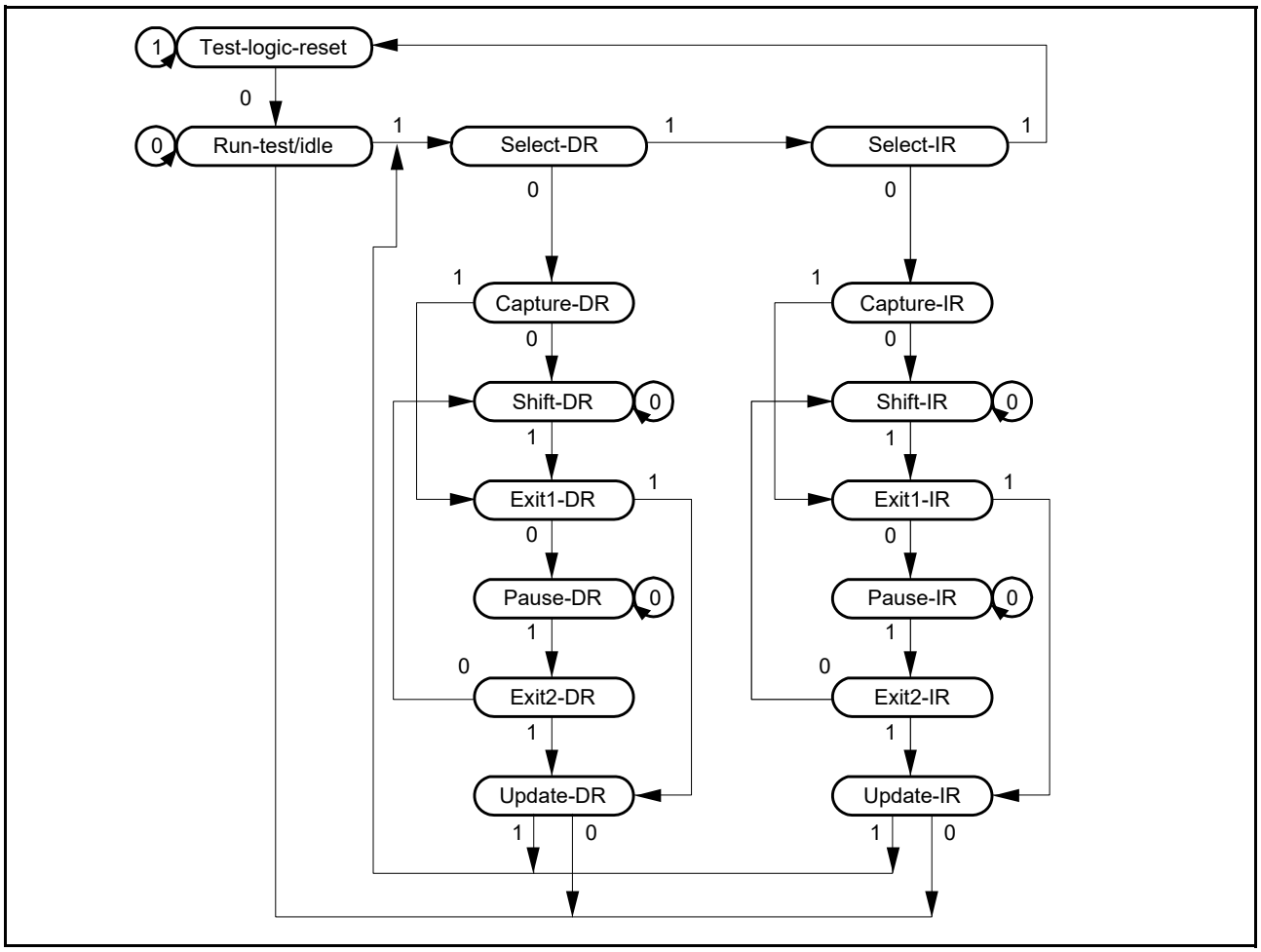


Figure 38.2 State transition diagram of TAP controller

38.3.2 List of Commands

(1) BYPASS

The BYPASS instruction drives the bypass register (JTBPR). This instruction shortens the shift path, facilitating the transfer of serial data to the other LSIs on a printed-circuit board at higher speeds. While this instruction is being executed, the test circuit has no effect on the system circuits.

The Bypass Register (JTBPR) is connected between the TDI and TDO pins. Bypass operation is initiated from the shift-DR operation. The TDO is low in the 1st clock cycle in the shift-DR state. In the subsequent clock cycles, the TDI signal is output on the TDO pin.

(2) EXTEST

The EXTEST instruction is used to test external circuits when the MCU is installed on the printed circuit board. When this instruction is executed, output pins are used to output test data (specified in the SAMPLE/PRELOAD instruction) from the Boundary Scan Register to the printed circuit board, and input pins are used to input the test result.

(3) SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction is used to input data from the internal circuits of the MCU to the Boundary Scan Register, output data from the scan path, and reload the data to the scan path. While this instruction is executed, input signals are directly input to the MCU and output signals are also directly output to the external circuits. The MCU system circuit is not affected by this instruction.

In SAMPLE operation, the Boundary Scan Register latches a snapshot of the data transferred from the input pins to the internal circuit or data transferred from the internal circuit to the output pins. The latched data is read from the scan path. The scan register latches the data snapshot at the rising edge of the TCK pin in the Capture-DR state. The data snapshot is transferred from the internal circuit to the output pins only during a reset.

In PRELOAD operation, the initial value is written from the scan path to the parallel output latch of the Boundary Scan Register prior to the EXTEST instruction execution. If EXTEST is executed without executing the PRELOAD operation, undefined values are output from the beginning to the end (transfer to the output latch) of the EXTEST sequence. In EXTEST instruction, the output parallel latches are always output to the output pins.

(4) IDCODE

When the IDCODE instruction is selected, the ID Code Register value is output to the TDO pin in the Shift-DR state of the TAP controller. The ID Code Register value is output LSB-first. During the instruction execution, the test circuit does not affect the system circuit.

(5) CLAMP

When the CLAMP instruction is selected, output pins output the Boundary Scan Register value that was specified in the SAMPLE/PRELOAD instruction in advance. While the CLAMP instruction is selected, the status of the Boundary Scan Register is maintained regardless of the TAP controller state.

The Bypass Register is connected between the TDI and TDO pins, leading to the same operation as when the BYPASS instruction is selected.

(6) HIGHZ

When the HIGHZ instruction is selected, all output pins enter a high-impedance state and the status of the Boundary Scan Register is maintained regardless of the state of the TAP controller.

The Bypass Register is connected between the TDI and TDO pins, leading to the same operation as when the BYPASS instruction is selected.

38.4 Usage Note

The boundary scan function is subject to the following constraints:

- The boundary scan must be executed when the RES pin is driven low.
- Serial data input or output is in LSB order, as shown in Figure 38.3.

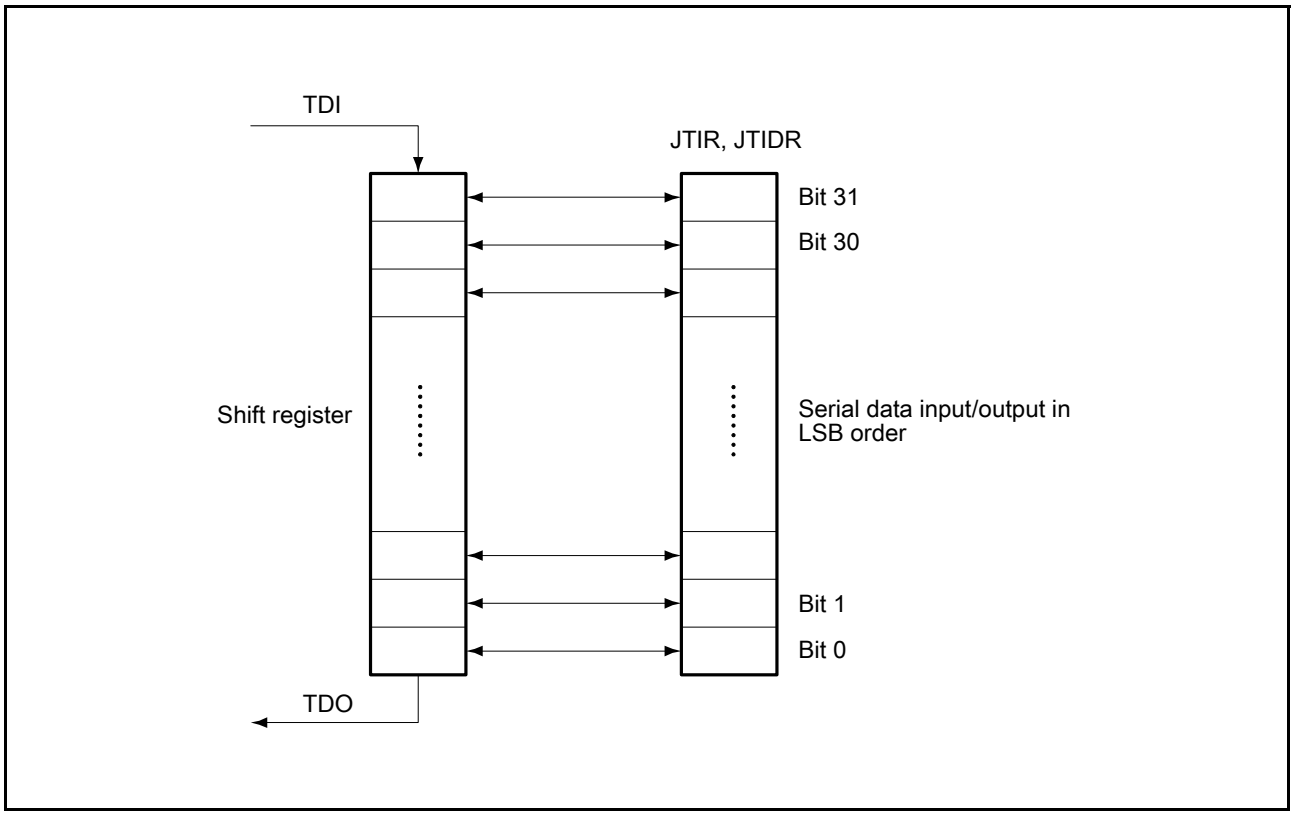


Figure 38.3 Serial data input/output

The following pins cannot be boundary-scanned:

- Power supply pins (VCC, VCL, VSS, VBATT, AVCC0, AVSS0, VCC_USB, and VSS_USB)
- Clock pins (EXTAL, XTAL, XCIN, and XCOUT)
- Reset signal (RES)
- USBFS dedicated pins (USB_DP and USB_DM)
- The boundary-scan pins (TCK, TMS, TDI, and TDO)
- The Mode signal (MD).

39. 14-Bit A/D Converter (ADC14)

39.1 Overview

The MCU incorporates up to one unit of a 14-bit successive approximation A/D converter. Up to 28 analog input channels are selectable. Temperature sensor output and internal reference voltage are selectable for conversion.
The A/D conversion accuracy is selectable from 14-bit conversion making it possible to optimize the trade-off between speed and resolution in generating a digital value.

The A/D converter function supports three operating modes: (1) single-scan mode to convert the analog inputs of arbitrarily selected channels in ascending order of channel number, (2) continuous-scan mode to sequentially convert the analog inputs of arbitrarily selected channels continuously in ascending order of channel number, and (3) group-scan mode to arbitrarily divide the analog inputs of channels into two groups (Group A and Group B) and convert the analog input of the selected channel for each group in ascending order of channel number.

In group-scan mode, Group A and Group B can start A/D conversion at different times by independently selecting their scan start conditions. When priority control operation of Group A is set, the scan start of Group A is accepted during the A/D conversion operation of Group B and the A/D conversion of Group B is suspended. Thus the A/D conversion operation of Group A is preferentially started.

In double-trigger mode, the analog input of an arbitrarily selected channel is converted in single-scan mode or group-scan mode (Group A), and the data converted by the 1st A/D conversion start trigger and the data converted by the second A/D conversion start trigger are stored in different registers (duplexing of A/D converted data).

Self-diagnosis is performed once at the beginning of each scan, and one of the three voltage values generated in the 14-bit ADC is A/D converted.

It is prohibited to simultaneously select both temperature sensor output and internal reference voltage. Perform A/D conversion independently for the temperature sensor output or the 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. The reference power supply pin (VREFH0) or the analog block power supply pin (AVCC0) or the internal reference voltage is selectable as the reference voltage on the high potential side. The reference power supply ground pin (VREFL0) or the analog block power supply ground pin (AVSS0) is selectable as the reference voltage on the low potential side.

The A/D converter provides a compare function (Window A and Window B). This compare function specifies the upper-side reference value and lower-side reference value for Window A and Window B respectively and outputs an interrupt when the A/D converted value of the selected channel meets the comparison conditions.

Table 39.1 lists the specifications of the 14-bit ADC. Table 39.2 indicates the functions of the 14-bit ADC. Figure 39.1 shows a block diagram of the 14-bit ADC.

Table 39.1 Specifications of 14-bit ADC (1 of 3)

Parameter	Specifications
Number of units	One unit
Input channels	Up to 28 channels (AN000 to AN027)
Extended analog function	Temperature sensor output, internal reference voltage
A/D conversion method	Successive approximation method
Resolution	14 bits (14-bit or 12-bit conversion selectable)
Conversion time	0.79 µs/channel (when 14-bit A/D conversion clock PCLKC (ADCLK) is operating at 64 MHz)
A/D conversion clock	Peripheral module clock PCLKB*1 and A/D conversion clock PCLKC (ADCLK)*1 can be set with the following division ratios: PCLKB to PCLKC (ADCLK) frequency ratio = 1:1, 2:1, 4:1, 8:1, 1:2, 1:4

Table 39.1 Specifications of 14-bit ADC (2 of 3)

Parameter	Specifications
Data registers	• 28 registers for analog input: 1 for A/D-converted data duplication in double-trigger mode 2 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 • The results of A/D conversion are stored in A/D data registers. • 12- and 14-bit accuracy output for the results of A/D conversion • The value obtained by adding up A/D-converted results is stored as a value in the number of bit for conversion accuracy + 2 bits*4 in the A/D data registers in A/D-converted value addition mode. • Double-trigger mode (selectable in single-scan and group-scan modes): The 1^{st} piece of A/D-converted analog-input data on one selected channel is stored in the data register for the channel, and the second piece 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 that is prepared for each type of trigger.
Operating modes	• Single-scan mode: A/D conversion is performed only once on the analog inputs of channels arbitrarily selected. A/D conversion is performed only once on the temperature sensor output. A/D conversion is performed only once on the internal reference voltage. • Continuous-scan mode: A/D conversion is performed repeatedly on the analog inputs of channels arbitrarily selected, on the temperature sensor output, and on the internal reference voltage. • Group-scan mode: Analog inputs of channels arbitrarily are divided into group A and group B, and A/D conversion of the analog input selected on a group basis is performed only once. The conditions for scanning start of group A and group B can be independently selected, thus allowing A/D conversion of group A and group B to be started independently. • Group-scan mode (when group A is given priority): If a group A trigger is input during A/D conversion on group B, the A/D conversion on group B is stopped and A/D conversion is performed on group A. Restart (rescan) of A/D conversion on group B after completion of A/D conversion on group A can be set.
Conditions for A/D conversion start	• Software trigger • Synchronous trigger Triggers from the event link controller (ELC). • Asynchronous trigger • A/D conversion can be triggered by the external trigger pins, ADTRG0.
Function	• Variable sampling state count • Self-diagnosis of A/D converter • Selectable A/D-converted value addition mode or average mode • Analog input disconnection detection function (discharge function/precharge function) • Double-trigger mode (duplication of A/D conversion data) • Switching function of 12 and 14-bit conversion*2 • Automatic clear function of A/D data registers • Digital comparison (comparison of values in the comparison register and the data register, and comparison between values in the data registers).

Table 39.1 Specifications of 14-bit ADC (3 of 3)

Parameter	Specifications
Interrupt source	• In single-scan mode (double trigger deselected), an A/D scan end interrupt request and ELC event signal (ADC140_ADI) can be generated on completion of single scan. A compare interrupt request (ADC140_CMPAI/ADC140_CMPBI) can be generated in response to matches with a condition for comparison by digital comparison. A window compare ELC event signal (ADC140_WCMPM) can be generated in response to matches with a condition for comparison by digital comparison. A window compare ELC event signal (ADC140_WCMPUM) can be generated in response to mismatches with a condition for comparison by digital comparison. • In single-scan mode (double trigger selected), an A/D scan end interrupt request and ELC event signal (ADC140_ADI) can be generated on completion of two scans. • In continuous-scan mode, an A/D scan end interrupt request and ELC event signal (ADC140_ADI) can be generated on completion of all the selected channels' scans. • In group-scan mode (double trigger deselected), an A/D scan end interrupt request and ELC event signal (ADC140_ADI) can be generated on completion of group A scan, whereas an A/D scan end interrupt request for group B (ADC140_GBADI) can be generated on completion of group B scan. • In group-scan mode (double trigger selected), an A/D scan end interrupt request and ELC event signal (ADC140_ADI) can be generated on completion of two group A scan, whereas an A/D scan end interrupt request for group B (ADC140_GBADI) can be generated on completion of group B scan. • The ADC140_ADI, ADC140_GBADI, ADC140_WCMPM, and ADC140_WCMPUM can activate the DMA controller (DMAC) and the data transfer controller (DTC).
ELC interface	• Scan can be started by a trigger from the ELC.
Reference voltage	• VREFH0, AVCC0, or internal reference voltage is selectable as the high potential-side reference voltage. • VREFL0 or AVSS0 is selectable as the low potential-side reference voltage.
Module stop function	Module stop state can be specified.*3

Note 1. Peripheral module clock PCLKB is set in the SCKDIVCR.PCKB[2:0] bits and A/D conversion clock ADCLK is set in the SCKDIVCR.PCKC[2:0]. The maximum frequency of PCLKB is 32 MHz and the maximum frequency of PCLKC (ADCLK) is 64 MHz.

Note 2. When A/D conversion accuracy is modified, A/D conversion time is also changed. See section 39.3.6, Analog Input Sampling and Scan Conversion Time for details.

Note 3. See section 11, Low Power Modes for details.

Note 4. The number of extended bits for addition varies with the A/D conversion accuracy and the number of addition times. A 2-bit extension is up to 4 times conversion (3 times addition) when the A/D conversion accuracy is 12 or 14 bits.

Note 5. When selecting the temperature sensor output or the internal reference voltage, do not use continuous-scan mode or group-scan mode.

Table 39.2 14-bit ADC functions

Parameter			ADC140
Analog input channel			AN000 to AN027 Internal reference voltage Temperature sensor output
Conditions for A/D conversion start	External trigger	Trigger input pin	ADTRG0
	Software	Software trigger	Enabled
	Synchronous trigger (trigger from ELC)	ELC trigger	ELC_AD00 ELC_AD01
Interrupt			ADC140_ADI ADC140_GBADI ADC140_CMPAI ADC140_CMPBI
Output to ELC			ADC140_ADI ADC140_WCMPM ADC140_WCMPUM
Setting of module stop function*1, *2			MSTPCRD.MSTPD16 bit

Note 1. See section 11, Low Power Modes for details.

Note 2. Wait for 1 µs or longer to start A/D conversion after release from the module-stop state.

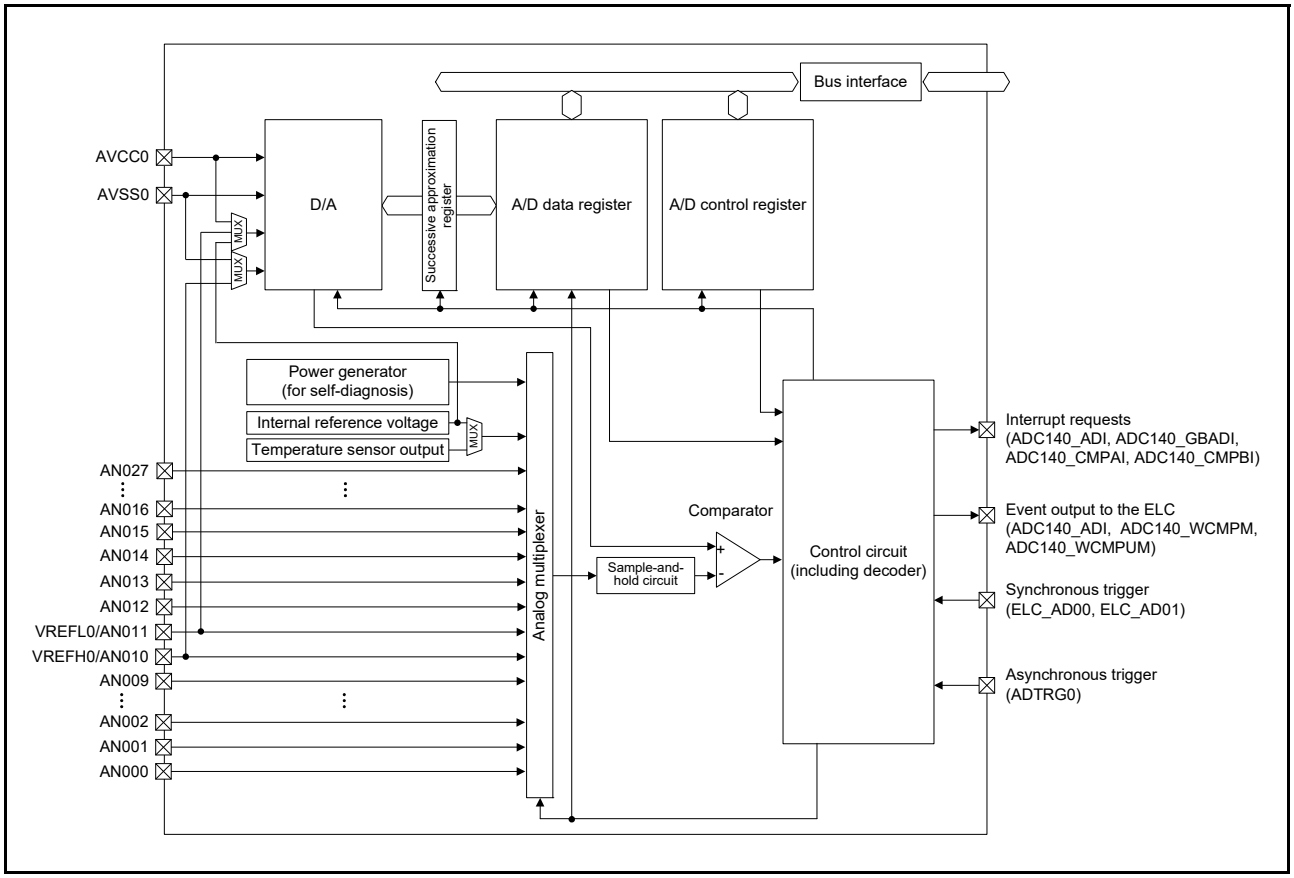


Figure 39.1 14-bit ADC block diagram

Table 39.2 lists the I/O pins of the 14-bit ADC.

Table 39.3 I/O pins of the A/D Converter

Unit	Pin name	I/O	Function
Unit 0	AVCC0	Input	Analog block power supply pin
	AVSS0	Input	Analog block power supply ground pin
	VREFH0	Input	Reference power supply pin
	VREFL0	Input	Reference power supply ground pin
	AN000 to AN027	Input	Analog input pins 0 to 27
	ADTRG0	Input	External trigger input pin for starting A/D conversion

39.2 Register Descriptions

39.2.1 A/D Data Registers y (ADDRy), A/D Data Duplexing Register (ADDBLDR), A/D Data Duplexing Register A (ADDBLDRA), A/D Data Duplexing Register B (ADDBLDRB), A/D Temperature Sensor Data Register (ADTSDR), A/D Internal Reference Voltage Data Register (ADOCDR)

The ADDRy registers (y = 0 to 27) are 16-bit read-only registers for storing the result of A/D conversion. Register ADDBLDR is a 16-bit read-only register for storing the result of A/D conversion in response to the second trigger in double-trigger mode. Registers ADDBLDRA and ADDBLDRB are 16-bit read-only registers for storing the result of A/D conversion in response to the respective triggers during extended operation in double-trigger mode. Register ADTSDR is a 16-bit read-only register for storing the A/D conversion result of temperature sensor output. Register ADOCDR is a 16-bit read-only register for storing the A/D result of internal reference voltage.

The formats for data in the ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, and ADOCDR registers vary according to the following conditions:

- The setting of the A/D data register format select bit (ADCER.ADRFMT) (determining whether the data is flush-left or flush-right in the registers)
- The setting of the A/D Conversion Accuracy Specify bits (ADCER.ADPRC[1:0]) (12 or 14-bit is selectable)
- The setting of the addition/average count select bits (ADADC.ADC[2:0]) (once, twice, three times, four times, or 16 times is selectable)
- The setting of the average mode enable bit (ADADC.AVEE) (addition or average is selectable).

(1) When A/D-converted value addition/average mode is not selected

The settings for flush-right data with 14-bit accuracy

Address(es): ADC140.ADDR0 4005 C020h to ADC140.ADDR27 4005 C056h, ADC140.ADDBLDR 4005 C018h, ADC140.ADDBLDRA 4005 C084h, ADC140.ADDBLDRB 4005 C086h, ADC140.ADTSDR 4005 C01Ah, ADC140.ADOCDR 4005 C01Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AD[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b13 to b0	AD[13:0]	Converted Value 13 to 0	14-bit A/D-converted value	R
b15, b14	—	Reserved	These bits are read as 0	R

The settings for flush-right data with 12-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	AD[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b11 to b0	AD[11:0]	Converted Value 11 to 0	12-bit A/D-converted value	R
b15 to b12	—	Reserved	These bits are read as 0	R

The settings for flush-left data with 14-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[13:0]														—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0	R
b15 to b2	AD[13:0]	Converted Value 13 to 0	14-bit A/D-converted value	R

The settings for flush-left data with 12-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[11:0]												—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0	R
b15 to b4	AD[11:0]	Converted Value 11 to 0	12-bit A/D-converted value	R

(2) When A/D-converted value average mode is selected

In case of selecting 2 or 4 times in A/D-converted value addition mode, A/D-converted value average mode can be selected. When A/D-converted value average mode is selected, this register indicates the mean of A/D-converted values on a specific channel. When A/D-converted value average mode is selected, the value is stored in the A/D data register according to the settings of the A/D data register format select bit in the same way as normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12 or 14-bit A/D data register bit-accuracy, 1, 2, 3, or 4 times can be selected in A/D -converted value addition mode. For 12-bit A/D data register bit-accuracy, 16 times can also be selected in A/D-converted value addition mode.

When A/D-converted value addition mode is selected, this register indicates the value that is obtained by adding up A/D converted values on a specific channel. The value is stored in the A/D data register according to the settings of the A/D data register format select bits.

For 12 or 14-bit A/D data register bit-accuracy, under the setting of 1, 2, 3, or 4 times in A/D-converted value addition mode, the A/D conversion result is retained in the A/D data register as a 2-bit-extended value of the conversion accuracy specified.

For 12-bit A/D data register bit-accuracy, under the setting of 16 times in A/D-converted value addition mode, the A/D conversion result is retained in the A/D data register as a 4-bit-extended value of the conversion accuracy specified.

The data formats for each given condition are shown below.

The settings for flush-right data with 14-bit accuracy

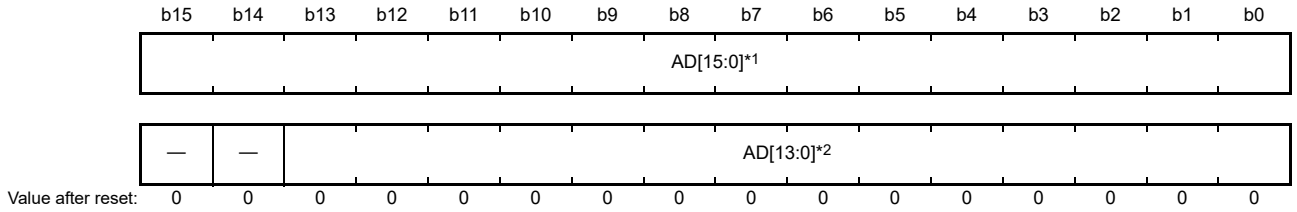
(When A/D-converted value addition mode is selected)

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

Bit	Symbol	Bit name	Description	R/W
b15 to b0	AD[15:0]	Added Value 15 to 0	16-bit value obtained by adding the A/D conversion results	R

The settings for flush-right data with 12-bit accuracy

(When A/D-converted value addition mode is selected)



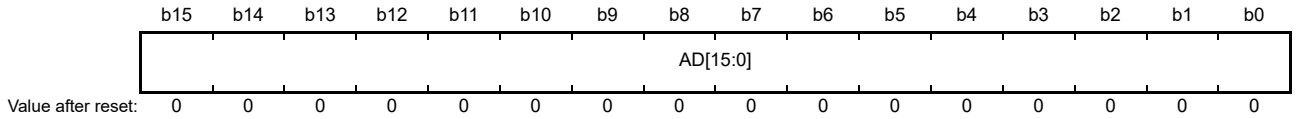
Bit	Symbol	Bit name	Description	R/W
b15 to b0	AD[15:0]*1	Added Value 15 to 0	16-bit value obtained by adding the A/D conversion results	R

Bit	Symbol	Bit name	Description	R/W
b13 to b0	AD[13:0]*2	Added Value 13 to 0	14-bit value obtained by adding the A/D conversion results	R
b15, b14	—	Reserved	These bits are read as 0	R

Note 1. Used when 16 conversion times is specified in A/D-converted value addition mode.
Note 2. Used when 1, 2, 3, or 4 conversion times is specified in A/D-converted value addition mode.

The settings for flush-left data with 14-bit accuracy

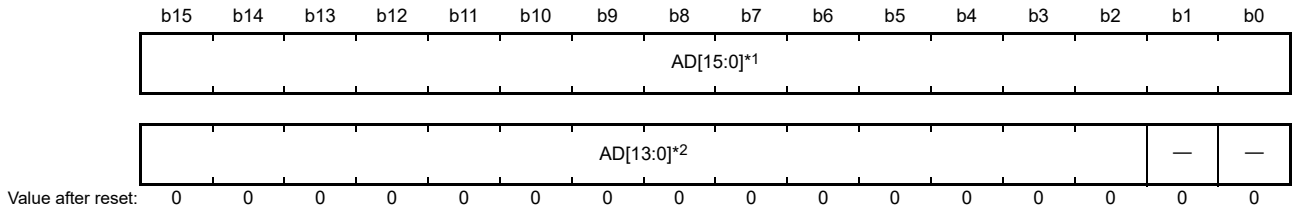
(When A/D-converted value addition mode is selected)



Bit	Symbol	Bit name	Description	R/W
b15 to b0	AD[15:0]	Added Value 15 to 0	16-bit value obtained by adding the A/D conversion results	R

The settings for flush-left data with 12-bit accuracy

(When A/D-converted value addition mode is selected)



Bit	Symbol	Bit name	Description	R/W
b15 to b0	AD[15:0]*1	Added Value 15 to 0	16-bit value obtained by adding the A/D conversion results	R

Bit	Symbol	Bit name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0	R
b15 to b2	AD[13:0]*2	Added Value 13 to 0	14-bit value obtained by adding the A/D conversion results	R

Note 1. Used when 16 conversion times is specified in A/D-converted value addition mode.
Note 2. Used when 1, 2, 3, or 4 conversion times is specified in A/D-converted value addition mode.

39.2.2 A/D Self-Diagnosis Data Register (ADRD)

ADRD is a 16-bit read-only register that holds the A/D conversion results based on the self-diagnosis of 14-bit ADC. In addition to the AD bits indicating A/D-converted value, the self-diagnosis status bit (DIAGST) is included. In the ADRD register, different formats are used depending on the following conditions:

- The setting of the A/D data register format select bit (ADCER.ADRFT) (determines whether the data are flush-left or flush-right in the registers)
- The setting of the A/D data register bit-accuracy specify bits (ADCER.ADPRC[1:0]) (12-bit or 14-bit)

The A/D-converted value addition mode and A/D-converted value average mode cannot be applied to the A/D self-diagnosis function. For details of self-diagnosis, see section 39.2.11, A/D Control Extended Register (ADCER). The data formats for each given condition are shown below.

The settings for flush-right data with 14-bit accuracy

Address(es): ADC140.ADRD 4005 C01Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DIAGST[1:0]		AD[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b13 to b0	AD[13:0]	Converted Value 13 to 0	14-bit A/D-converted value	R
b15, b14	DIAGST[1:0]	Self-Diagnosis Status	b15 b14 0 0: Self-diagnosis has not been executed since power-on 0 1: Self-diagnosis was executed using the voltage of 0 V 1 0: Self-diagnosis was executed using the voltage of reference power supply*1 × 1/2 1 1: Self-diagnosis was executed using the voltage of reference power supply*1. For details of self-diagnosis, see section 39.2.11, A/D Control Extended Register (ADCER).	R

Note 1. "Reference voltage" refers to VREFH0.

The settings for flush-right data with 12-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit name	Description	R/W
b11 to b0	AD[11:0]	Converted Value 11 to 0	12-bit A/D-converted value	R
b13, b12	—	Reserved	These bits are read as 0	R
b15, b14	DIAGST[1:0]	Self-Diagnosis Status	b15 b14 0 0: Self-diagnosis has not been executed since power-on 0 1: Self-diagnosis was executed using the voltage of 0 V 1 0: Self-diagnosis was executed using the voltage of reference power supply*1 × 1/2 1 1: Self-diagnosis was executed using the voltage of reference power supply*1. For details of self-diagnosis, see section 39.2.11, A/D Control Extended Register (ADCER).	R

Note 1. "Reference voltage" refers to VREFH0.

The settings for flush-left data with 14-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[13:0]														DIAGST[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	DIAGST[1:0]	Self-Diagnosis Status	b1 b0 0 0: Self-diagnosis has not been executed since power-on 0 1: Self-diagnosis was executed using the voltage of 0 V 1 0: Self-diagnosis was executed using the voltage of reference power supply*1 × 1/2 1 1: Self-diagnosis was executed using the voltage of reference power supply*1. For details of self-diagnosis, see section 39.2.11, A/D Control Extended Register (ADCER).	R
b15 to b2	AD[13:0]	Converted Value 13 to 0	14-bit A/D-converted value	R

Note 1. "Reference voltage" refers to VREFH0.

The settings for flush-left data with 12-bit accuracy

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AD[11:0]												—	—	DIAGST[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	DIAGST[1:0]	Self-Diagnosis Status	b1 b0 0 0: Self-diagnosis has not been executed since power-on 0 1: Self-diagnosis was executed using the voltage of 0 V 1 0: Self-diagnosis was executed using the voltage of reference power supply*1 × 1/2 1 1: Self-diagnosis was executed using the voltage of reference power supply*1. For details of self-diagnosis, see section 39.2.11, A/D Control Extended Register (ADCER).	R
b3, b2	—	Reserved	These bits are read as 0	R
b15 to b4	AD[11:0]	Converted Value 11 to 0	12-bit A/D-converted value	R

Note 1. "Reference voltage" refers to VREFH0.

39.2.3 A/D Control Register (ADCSR)

Address(es): ADC140.ADCSR 4005 C000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit name	Description	R/W
b4 to b0	DBLANS[4:0]	Double Trigger Channel Select	These bits select one analog input channel for double-triggered operation. The setting is only effective when double-trigger mode is selected.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	GBADIE	Group B Scan End Interrupt Enable	0: Disables ADC140_GBADI interrupt generation upon group B scan completion 1: Enables ADC140_GBADI interrupt generation upon group B scan completion Group B scan works only in group-scan mode.	R/W
b7	DBLE	Double Trigger Mode Select	0: Deselects double-trigger mode 1: Selects double-trigger mode.	R/W
b8	EXTRG	Trigger Select*1	0: A/D conversion is started by a synchronous trigger (ELC) 1: A/D conversion is started by the asynchronous trigger (ADTRG0).	R/W
b9	TRGE	Trigger Start Enable	0: Disables A/D conversion to be started by the synchronous or asynchronous trigger 1: Enables A/D conversion to be started by the synchronous or asynchronous trigger.	R/W
b10	ADHSC	A/D Conversion Mode Select	0: High-speed A/D conversion mode 1: Low power A/D conversion mode.	R/W
b12, b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14, b13	ADCS[1:0]	Scan Mode Select	b14 b13 0 0: Single-scan mode 0 1: Group-scan mode 1 0: Continuous-scan mode 1 1: Setting prohibited.	R/W
b15	ADST	A/D Conversion Start	0: Stops A/D conversion process 1: Starts A/D conversion process.	R/W

Note 1. Starting 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 ADCSR and drive ADTRG0 low. The falling edge of ADTRG0 is detected and the scan conversion process is started. In this case, the pulse width of the low-level input must be at least 1.5 PCLKB clock cycles.

DBLANS[4:0] bits (Double Trigger Channel Select)

The DBLANS[4:0] bits select one of the channels for A/D conversion data duplication in double-trigger mode. The A/D conversion results of the analog input of the channel selected by the DBLANS[4:0] bits are stored into the A/D data register y when conversion is started by the 1st trigger, and into the A/D data duplexing register when started by the second trigger. Table 39.4 shows selection of the channel for double-triggered operation.

A/D-converted value addition/average mode with double-trigger mode can be set by selecting the channel selected by the DBLANS[4:0] bits using the ADADS0/1 register. When double-trigger mode is selected, the channels selected by registers ADANSA0 and ADANSA1 are invalid, and the channel selected by the DBLANS[4:0] bits is subjected to A/D conversion instead.

When double-trigger mode is used in group-scan mode, double-trigger control is applied to only group A and not applied to group B. Therefore, multi-channel analog input can be selected for group B even in double-trigger mode.

The DBLANS[4:0] bits should be set while the ADST bit is 0. These bits should not be set simultaneously when 1 is written to the ADST bit.

To enter A/D-converted value addition/average mode while double-trigger mode is set, the channel selected by the DBLANS[4:0] bits should be selected in the ADANSA0 and ADANSA1 registers.

Table 39.4 Relationship between DBLANS bit settings and double-trigger enabled channels

DBLANS[4:0]	Duplication channel	DBLANS[4:0]	Duplication channel
00000	AN000	10000	AN016
00001	AN001	10001	AN017
00010	AN002	10010	AN018
00011	AN003	10011	AN019
00100	AN004	10100	AN020
00101	AN005	10101	AN021
00110	AN006	10110	AN022
00111	AN007	10111	AN023
01000	AN008	11000	AN024
01001	AN009	11001	AN025
01010	AN010	11010	AN026
01011	AN011	11011	AN027
01100	AN012		
01101	AN013		
01110	AN014		
01111	AN015		

Note 1. A/D-converted data of the self-diagnosis function, temperature sensor output, and internal reference voltage cannot be used in double-trigger mode.

GBADIE bit (Group B Scan End Interrupt Enable)

The GBADIE bit enables or disables group B scan end interrupt (ADC140_GBADI) in group-scan mode.

DBLE bit (Double Trigger Mode Select)

Double-trigger mode can be only operated by synchronous trigger (ELC) selected by the ADSTRGR.TRSA[5:0] bits. In double-trigger mode:

- The ADC140_ADI interrupt is output not upon completion of the 1st conversion but upon completion of the second conversion.
- The A/D conversion results of the duplication channel (selected by DBLANS[4:0] bits) started by the 1st trigger are stored into the A/D data register y and the results started by the second trigger are stored into the A/D data duplication register.

When DBLE is set (double-trigger mode is selected), the channels specified in the ADANSA0 and ADANSA1 registers are invalid. Do not select double-trigger mode in continuous-scan mode.

Software trigger cannot be used in double-trigger mode. The DBLE bit should be set after the ADST bit is set to 0 (it should not be set simultaneously when 1 is written to the ADST bit).

EXTRG bit (Trigger Select)

The EXTRG bit selects the synchronous trigger or the asynchronous trigger as the trigger for starting A/D conversion.

TRGE bit (Trigger Start Enable)

The TRGE bit enables or disables A/D conversion by the synchronous trigger and the asynchronous trigger. This bit should be set to 1 in group-scan mode.

ADHSC bit (A/D Conversion Mode Select)

The ADHSC bit selects either the high-speed mode or low-current mode of A/D conversion.

For details on how to rewrite this bit, see section 39.8.8, ADHSC Bit Rewriting Procedure.

ADCS[1:0] bits (Scan Mode Select)

The ADCS bit selects the scan mode.

In single-scan mode, A/D conversion is performed for the analog inputs of a maximum of 28 channels selected with the ADANSA0 and ADANSA1 registers in ascending order of channel number. When one cycle of A/D conversion is completed for all the selected channels, the scan conversion is stopped.

In continuous-scan mode, while the ADST bit in ADCSR is 1, A/D conversion is performed for the analog inputs selected with the ADANSA0 and ADANSA1 registers in ascending order of channel number. When one cycle of A/D conversion is completed for all the selected channels, A/D conversion is repeated from the 1st channel. If the ADST bit in ADCSR is set to 0 during continuous-scan, A/D conversion is stopped even if scanning is in progress.

In group-can mode, A/D conversion is performed for the analog inputs (group A), of a maximum selected with the ADANSA0 and ADANSA1 registers, in ascending order of channel number. After scanning is started by the synchronous trigger (ELC) selected by the TRSA[5:0] bits in ADSTRGR, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped. A/D conversion is also performed for the analog inputs (group B) of a maximum of 28 channels selected with the ADANSB0 and ADANSB1 registers in ascending order of channel number. After scanning is started by the synchronous trigger (ELC) selected by the TRSB[5:0] bits in ADSTRGR, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped. If the conversion processes in group A and B occur at the same time, those conversion cannot be controlled separately. In this case, set the group A priority control setting bit (ADGSPCR.PGS) in the A/D group scan priority control register (ADGSPCR) to 1 in order to assign priority to conversion of group A. In group-scan mode, different channels and triggers should be selected 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 channels by setting ADANSA0/1 register. When completing A/D conversion of the selected temperature sensor output or internal reference voltage, A/D conversion stops.

The ADCS[1:0] bits should be set while the ADST bit is 0 (it should not be set simultaneously when 1 is written to the ADST bit).

Table 39.5 Selectable targets for A/D conversion depending on settings of scan mode and double-trigger mode

Scan mode setting	Double trigger mode setting	Targets for A/D conversion				
		Self-diagnosis	Analog input (including 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)	✓	×	×

✓: Selectable. ×: Not selectable.

ADST bit (A/D Conversion Start)

The ADST bit starts or stops A/D conversion process.

Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and conversion target analog input.

[Setting conditions]

- When 1 is written by software.
- When the synchronous trigger (ELC) selected by the ADSTRGR.TRSA[5:0] bits is detected with ADCSR.EXTRG and ADCSR.TRGE bits being set to 0 and 1, respectively
- When the synchronous trigger (ELC) selected by the ADSTRGR.TRSB[5:0] bits is detected with the ADCSR.TRGE bit being set to 1 in group-scan mode
- When the asynchronous trigger is detected with the ADCSR.TRGE and ADCSR.EXTRG bits being set to 1 and the ADSTRGR.TRSA[5:0] bits being set to 000000b
- With group A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), a group B trigger is detected and A/D conversion of group B is started

- With group A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1 and A/D conversion of group B is restarted
- With group A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRP bit is set to 1 and each time A/D conversion of group B is started.

[Clearing conditions]

- When 0 is written by software
- When the A/D conversion of all the selected channels, the temperature sensor output or the internal reference voltage is completed in single-scan mode
- When group A scan is completed in group-scan mode
- When group B scan is completed in group-scan mode
- With group A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), a group A trigger is detected during group B A/D conversion and the scanning of group B is stopped
- With group A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1 and the scanning of group B started by a resumption trigger is completed
- With group A priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1) the ADGSPCR.GBRP bit is set to 1 and each time a scanning of group B is completed.

Note: When group A priority control operation mode is enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), do not set the ADST bit to 1.

Note: When group A priority control operation mode is enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1) and ADGSPCR.GBRP bit = 1, do not set the ADST bit to 0. When forcibly terminating A/D conversion, follow the procedure for clearing the ADST bit.

39.2.4 A/D Channel Select Register A0 (ADANSA0)

Address(es): ADC140.ADANSA0 4005 C004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ANSA15	ANSA14	ANSA13	ANSA12	ANSA11	ANSA10	ANSA09	ANSA08	ANSA07	ANSA06	ANSA05	ANSA04	ANSA03	ANSA02	ANSA01	ANSA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	ANSA15 to ANSA00	A/D Conversion Channels Select	0: Do not select associated input channel 1: Select associated input channel. Bit 15 (ANSA15) associated with AN015 and bit 0 (ANSA00) associated with AN000.	R/W

ANSAn bits (n = 00 to 15) (A/D Conversion Channels Select)

The ADANSA0 register selects analog input channels for A/D conversion among AN000 to AN015. The channels to be selected and the number of channels can be arbitrarily set. The ANSA00 bit corresponds to AN000 and the ANSA15 bit corresponds to AN015. When performing A/D conversion of temperature sensor output or internal reference voltage, do not select analog any input channels. (Set this register to 0000h.)

When double-trigger mode is selected, the channel selection in the ADANSA0 register is invalid. Instead, the group A channel specified by the ADCSR.DBLANS[4:0] bits is selected.

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).

The ADANSA0 register should be set while the ADCSR.ADST bit is 0.

39.2.5 A/D Channel Select Register A1 (ADANSA1)

Address(es): ADC140.ADANSA1 4005 C006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ANSA27	ANSA26	ANSA25	ANSA24	ANSA23	ANSA22	ANSA21	ANSA20	ANSA19	ANSA18	ANSA17	ANSA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b11 to b0	ANSA27 to ANSA16	A/D Conversion Channels Select	0: Do not select associated input channel 1: Select associated channel. Bit 11 (ANSA27) corresponds to AN027 and bit 0 (ANSA16) corresponds to AN016.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ANSAn Bits (n = 16 to 27) (A/D Conversion Channels Select)

The ADANSA1 register selects analog input channels for A/D conversion among AN016 to AN027. The channels to be selected and the number of channels can be arbitrarily set. The ANSA16 bit corresponds to AN016 and the ANSA27 bit corresponds to AN027. When double-trigger mode is selected, the channel selection in the ADANSA1 register is invalid. Instead, the group A channel specified by the ADCSR.DBLANS[4:0] bits is selected.

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).

The ADANSA1 register should be set while the ADCSR.ADST bit is 0.

When selecting A/D conversion of temperature sensor output or internal reference voltage, do not select any analog input channels. (Set this register to 0000h.)

39.2.6 A/D Channel Select Register B0 (ADANSB0)

Address(es): ADC140.ADANSB0 4005 C014h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ANSB15	ANSB14	ANSB13	ANSB12	ANSB11	ANSB10	ANSB09	ANSB08	ANSB07	ANSB06	ANSB05	ANSB04	ANSB03	ANSB02	ANSB01	ANSB00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	ANSB15 to ANSB00	A/D Conversion Channels Select	0: Do not select associated input channel. 1: Select associated input channel. Bit [15] (ANSB15) corresponds to AN015 and bit [0] (ANSB00) corresponds to AN000.	R/W

ANSBn bits (n = 00 to 15) (A/D Conversion Channels Select)

The ADANSB0 register selects analog input channels for A/D conversion among AN000 to AN015 in group B when group-scan mode is selected. The ADANSB0 register is used for group-scan mode only and not for any other modes. The channels specified in group A (the channels associated with group A, selected with the ADANSA0 and ADANSA1 registers and the ADCSR.DBLANS[4:0] bits in double-trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

The ANSB00 bit corresponds to AN000, the ANSB07 bit corresponds to AN007, and the ANSB15 bit corresponds to AN015.

The ADANSB register should be set while the ADCSR.ADST bit is 0. When selecting temperature sensor output or internal reference voltage, do not select any analog input channels. (Set this register to 0000h.)

39.2.7 A/D Channel Select Register B1 (ADANSB1)

Address(es): ADC140.ADANSB1 4005 C016h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ANSB27	ANSB26	ANSB25	ANSB24	ANSB23	ANSB22	ANSB21	ANSB20	ANSB19	ANSB18	ANSB17	ANSB16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b11 to b0	ANSB27 to ANSB16	A/D Conversion Channels Select	0: Do not select associated input channel 1: Select associated input channel. Bit 11 (ANSB27) corresponds to AN027 and bit 0 (ANSB16) corresponds to AN016.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ANSBn bits (n = 16 to 27) (A/D Conversion Channels Select)

The ADANSB1 register selects analog input channels for A/D conversion among AN016 to AN027 in group B when group-scan mode is selected. The ADANSB1 register is used for group scan mode only and not for any other modes. The channels specified in group A (the channels corresponding to group A, selected with the ADANSA0 and ADANSA1 registers and the ADCSR.DBLANS[4:0] bits in double-trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

The ANSB16 bit corresponds to AN016, the ANSB20 bit corresponds to AN020, and the ANSB27 bit corresponds to AN027.

The ADANSB1 register bits should be set while the ADST bit is 0. When selecting temperature sensor output or internal reference voltage, do not select analog input channels. Set 0000h to this register.

39.2.8 A/D-Converted Value Addition/Average Channel Select Register 0 (ADADS0)

Address(es): ADC140.ADADS0 4005 C008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADS15	ADS14	ADS13	ADS12	ADS11	ADS10	ADS09	ADS08	ADS07	ADS06	ADS05	ADS04	ADS03	ADS02	ADS01	ADS00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	ADS15 to ADS00	A/D-Converted Value Addition/ Average Channel Select	0: Do not select associated input channel 1: Select associated input channel. Bit 15 (ADS15) corresponds to AN015 and bit 0 (ADS00) corresponds to AN000.	R/W

ADSn bits (n = 00 to 15) (A/D-Converted Value Addition/Average Channel Select)

When the ADSn bit corresponding to the A/D-converted channel selected by the ANSAn bits (n = 00 to 15) in ADANSA0, or DBLANS[4:0] bits in ADCSR and ANSBn bits (n = 00 to 15) in ADANSB0, is set to 1, A/D conversion of the analog input of the selected channels is performed successively 1 to 16 times as determined by the setting of the ADC[2:0] bits in ADADC. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register. For the channel on which the A/D conversion is performed and for which addition/ average mode is not selected, a normal one-time conversion is executed and the conversion result is stored in the A/D data register.

The ADADS0 register bits should be set while the ADCSR.ADST bit is 0.

39.2.9 A/D-Converted Value Addition/Average Channel Select Register 1 (ADADS1)

Address(es): ADC140.ADADS1 4005 C00Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ADS27	ADS26	ADS25	ADS24	ADS23	ADS22	ADS21	ADS20	ADS19	ADS18	ADS17	ADS16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b11 to b0	ADS27 to ADS16	A/D-Converted Value Addition/Average Channel Select	0: Do not select associated input channel 1: Select associated input channel. Bit 11 (ADS27) corresponds to AN027 and bit 0 (ADS16) corresponds to AN016.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSn bits (n = 16 to 27) (A/D-Converted Value Addition/Average Channel Select)

When the ADSn bit corresponding to the A/D-converted channel selected by the ANSAn bits (n = 16 to 27) in ADANSA1, or DBLANS[4:0] bits in ADCSR and ANSBn bits (n = 16 to 27) in ADANSB1, is set to 1, A/D conversion of the analog input of the selected channels is performed successively 1 to 16 times as determined by the setting of the ADC[2:0] bits in ADADC. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register. For the channel on which the A/D conversion is performed and for which addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored in the A/D data register.

The ADADS1 register should be set while the ADCSR.ADST bit is 0. Figure 39.2 shows a scanning operation sequence in which both the ADADS0.ADS02 and ADADS0.ADS06 bits are set to 1.

It is assumed that addition mode is selected (ADADS.AVEE = 0), the time conversion is set to 4 (ADADC.ADC[1:0] = 11b), and channels AN000 to AN007 are selected (ADANSA0.ANSA0[15:0] = 00FFh) in continuous-scan mode (ADCSR.ADCS[1:0] = 10b). The conversion process begins with AN000. The AN002 conversion is performed successively 4 times, and the added (integrated) value is returned to A/ D data register 2 (ADDR2). Next, the AN003 conversion process is started. The AN006 conversion is performed successively 4 times and the added (integrated) value is returned to A/D data register 6 (ADDR6). After conversion of AN007, the conversion operation is once again performed in the same sequence from AN000.

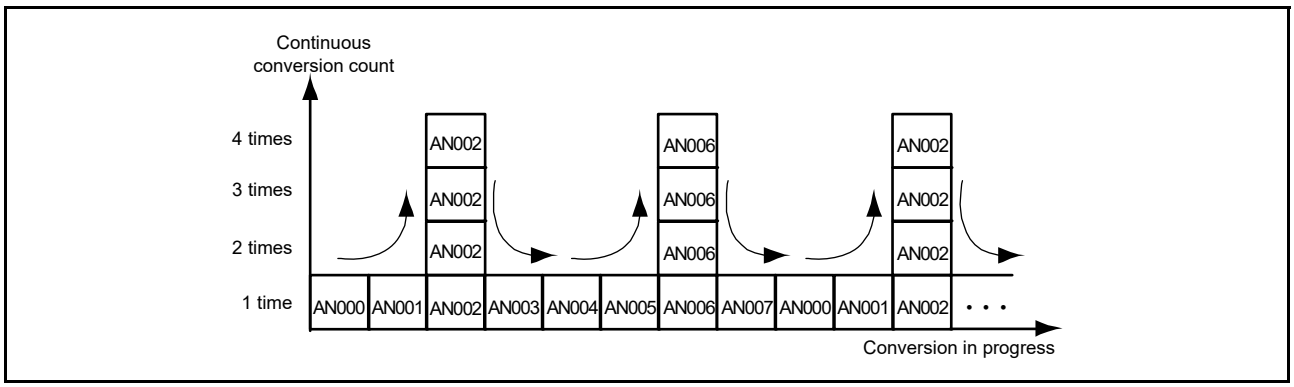


Figure 39.2 Scan conversion sequence with ADADC.ADC[2:0] = 011b, ADADS0.ADS02 = 1, ADADS0.ADS06 = 1

39.2.10 A/D-Converted Value Addition/Average Count Select Register (ADADC)

Address(es): ADC140.ADADC 4005 C00Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	AVEE	—	—	—	—	ADC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	ADC[2:0]	Count Select	b2 b0 0 0 0: 1-time conversion (no addition: same as normal conversion) 0 0 1: 2-time conversion (addition once) 0 1 0: 3-time conversion (addition twice) 0 1 1: 4-time conversion (addition three times) 1 0 1: 16-time conversion (addition 15 times). Other settings are prohibited.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	AVEE	Average Mode Enable	0: Average mode is disabled*1 1: Average mode is enabled.*2	R/W

Note 1. When average mode is not selected (by setting the ADADC.AVEE bit to 0), set addition count to 1, 2, 3, 4, or 16-time conversion. Note that 16-time conversion can be used with 12-bit accuracy only.

Note 2. When average mode is selected (by setting the ADADC.AVEE bit to 1), set addition count to 2-time or 4-time conversion; do not set the addition count to 3-time or 16-time conversion (ADC[2:0] = 010b and 101b).

ADC[2:0] bits (Count Select)

The ADC[2:0] bits set the addition count common to the channels for which A/D conversion and A/D-converted value addition/average mode is selected, including the channels selected in double-trigger mode (by ADCSR.DBLANS[4:0] bits), and to A/D conversion of temperature sensor output and internal reference voltage.

When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to 3-time conversion (ADADC.ADC[2:0] = 010b). The combination of the addition count 16-time conversion (ADADC.ADC[2:0] = 101b) with the conversion accuracy 14 bits (ADCER.ADPRC[1:0] = 11b) is prohibited setting, as described in section 39.2.1.

The ADC[2:0] bits should be set while the ADCSR.ADST bit is 0. When self-diagnosis is executed (ADCER.DIAGM = 1), do not set the ADC[2:0] bits to any value other than 000b. When the conversion accuracy 14 bits (ADCER.ADPRC[1:0] = 11b), do not set the ADC[2:0] bits to 101b.

AVEE bit (Average Mode Enable)

The AVEE bit selects addition or average mode for A/D conversion of the channels for which A/D conversion and A/D-converted value addition/average mode is selected, including the channels selected in double-trigger mode (by ADCSR.DBLANS[4:0] bits), temperature sensor output, and internal reference voltage. When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to 3-time conversion (ADADC.ADC[2:0] = 010b).

The AVEE bits should be set while the ADCSR.ADST bit is 0.

39.2.11 A/D Control Extended Register (ADCER)

Address(es): ADC140.ADCER 4005 C00Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADRFMT	—	—	—	DIAGM	DIAGLD	DIAGVAL[1:0]		—	—	ACE	—	—	ADPRC[1:0]		—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2, b1	ADPRC[1:0]	A/D Conversion Accuracy Specify	b2 b1 0 0: A/D conversion is performed with 12-bit accuracy 0 1: Setting prohibited 1 0: Setting prohibited 1 1: A/D conversion is performed with 14-bit accuracy.	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	ACE	A/D Data Register Automatic Clearing Enable	0: Disables automatic clearing 1: Enables automatic clearing.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	DIAGVAL[1:0]	Self-Diagnosis Conversion Voltage Select	b9 b8 0 0: Setting prohibited when self-diagnosis is enabled 0 1: Uses the voltage of 0 V for self-diagnosis 1 0: Uses the voltage of reference power supply*1 × 1/2 for self-diagnosis 1 1: Uses the voltage of reference power supply*1 for self-diagnosis.	R/W
b10	DIAGLD	Self-Diagnosis Mode Select	0: Rotation mode for self-diagnosis voltage 1: Fixed mode for self-diagnosis voltage.	R/W
b11	DIAGM	Self-Diagnosis Enable	0: Disables self-diagnosis of 14-bit ADC 1: Enables self-diagnosis of 14-bit ADC.	R/W
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	ADRFMT	A/D Data Register Format Select	0: Flush-right is selected for the A/D data register format 1: Flush-left is selected for the A/D data register format.	R/W

Note 1. Reference voltage refers to VREFH0.

ADPRC[1:0] bits (A/D Conversion Accuracy Specify)

These bits select the A/D conversion accuracy, either 12 or 14-bit. When the A/D conversion accuracy is changed, the bit width of effective data stored in the result register and A/D conversion time are also changed.

See section 39.3.6, Analog Input Sampling and Scan Conversion Time for details. The ADPRC[1:0] bits should be set while the ADCSR.ADST bit is 0.

ACE bit (A/D Data Register Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all 0) of ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, or ADOCDR after any of these registers are read by the CPU, DTC, or DMAC.

DIAGVAL[1:0] bits (Self-Diagnosis Conversion Voltage Select)

These bits select the voltage value used in self-diagnosis voltage fixed mode. For details, see the descriptions of the ADCER.DIAGLD bit.

Self-diagnosis should not be executed when the ADCER.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 this bit (ADCER.DIAGLD) to 0 allows conversion of the voltages in rotation mode where 0, the reference power supply × 1/2, and the reference power supply are converted in that order. After reset, when the self-diagnosis voltage rotation mode is selected, the self-diagnosis is executed from 0V. The fixed voltage specified by the ADCER.DIAGVAL[1:0] bits is converted when self-diagnosis voltage fixed mode is selected. In self-diagnosis voltage rotation mode, the self-diagnosis voltage value does not return to 0 when scan conversion is completed. When scan conversion is restarted, therefore, rotation starts at the voltage value following the previous value. If fixed mode is switched to rotation mode, rotation starts at the fixed voltage value.

The DIAGLD bit should be set while the ADCSR.ADST bit is 0.

DIAGM bit (Self-Diagnosis Enable)

The DIAGM bit enables or disables self-diagnosis.

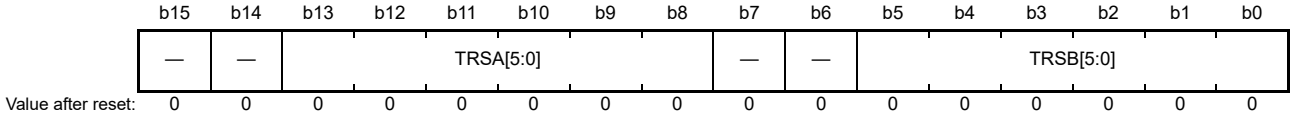
Self-diagnosis is used to detect a failure of the 14-bit A/D converter. In self-diagnosis mode, one of the internally generated voltage values (0, the reference power supply × 1/2, or the reference power supply) is converted. When conversion is completed, information on the converted voltage and the conversion result is stored into the self-diagnosis data register (ADRD). ADRD can then be read out by software to determine whether the conversion result falls within the normal range (normal) or not (abnormal). Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. When the double-trigger mode is set (ADCSR.DBLE = 1), self-diagnosis should be disabled (DIAGM = 0). When self-diagnosis is enabled in group-scan mode, self-diagnosis is separately executed in group A and B. The DIAGM bit should be set while the ADCSR.ADST bit is 0.

ADRFMT bit (A/D Data Register Format Select)

The ADRFMT bit specifies flush-right or flush-left for the data to be stored in ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR, ADCMPDR0/1, ADWINLLB, ADWINULB, or ADRD. The ADRFMT bit should be set while the ADCSR.ADST bit is 0.

39.2.12 A/D Conversion Start Trigger Select Register (ADSTRGR)

Address(es): ADC140.ADSTRGR 4005 C010h



Bit	Symbol	Bit name	Description	R/W
b5 to b0	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
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	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
b15, b14	—	Reserved	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 are used only in group-scan mode; they 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. Therefore, the TRSB[5:0] bits should be set to a value other than 000000b and the ADCSR.TRGE bit should be set to 1 in group-scan mode.

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 3Fh. Note that the issuance period of the trigger for A/D conversion 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 the trigger may have no effect.

When the GPT module is selected as an A/D conversion start trigger, a delay for synchronization processing occurs. See section 39.3.6, Analog Input Sampling and Scan Conversion Time for details. Table 39.6 lists the A/D conversion startup sources selected by the TRSB[5:0] bits.

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. In group-scan mode, the trigger to start scanning of the analog input selected in group A is selected. When scanning is executed in group-scan mode or double-trigger mode, do not use a software trigger or an asynchronous trigger.

- When using the A/D conversion startup source of a synchronous trigger (ELC), set the TRGE bit in ADCSR to 1 and set the EXTRG bit in ADCSR to 0.
- 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. Note that the issuance period of trigger for A/D conversion 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 may have no effect. See section 39.3.6, Analog Input Sampling and Scan Conversion Time for details.

Table 39.7 lists the A/D conversion start sources selected by the TRSA[5:0] bits.

Table 39.6 Selection of A/D activation sources by TRSB[5:0] bits

Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
Trigger source de-selection 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

Table 39.7 Selection of A/D activation sources by TRSA[5:0] bits

Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
Trigger source de-selection 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

39.2.13 A/D Conversion Extended Input Control Register (ADEXICR)

Address(es): ADC140.ADEXICR 4005 C012h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	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	Bit name	Description	R/W
b0	TSSAD	Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select	0: Temperature sensor output A/D-converted value addition/average mode not selected 1: Temperature sensor output A/D-converted value addition/average mode selected.	R/W
b1	OCSAD	Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select	0: Internal reference voltage A/D-converted value addition/average mode not selected 1: Internal reference voltage A/D-converted value addition/average mode selected.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TSSA	Temperature Sensor Output A/D Conversion Select	0: A/D conversion of temperature sensor output disabled 1: A/D conversion of temperature sensor output enabled.	R/W
b9	OCSA	Internal Reference Voltage A/D Conversion Select	0: A/D conversion of internal reference voltage disabled 1: A/D conversion of internal reference voltage enabled.	R/W
b15 to b10	—	Reserved	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 by the ADC[2:0] bits in ADADC. The maximum addition count depends on the conversion accuracy (see section 39.2.1). 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.

The TSSAD bit should be set 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 by the ADC[2:0] bits in ADADC. The maximum addition count depends on the conversion accuracy (see section 39.2.1). 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.

The OCSAD bit should be set 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. Set 0 in all bits of the ADANSA0/1, ADANSB0/1, ADCSR.DBLE and OCSA, and use single-scan mode when executing the A/D conversion of the temperature sensor output. The TSSA bit should be set while the ADCSR.ADST bit is 0.

When executing the A/D conversion of the temperature sensor output, the ADDISCR register is set to 0Fh and the A/D converter executes discharge (15 ADCLK). After executing discharge, the A/D converter executes sampling. The minimum sampling time is 5 µs. The A/D converter executes discharge each time it executes A/D conversion of 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. Set 0 in all bits of the ADANSA0/1, ADANSB0/1, ADCSR.DBLE and TSSA, and use single-scan mode when executing the A/D conversion of the internal reference voltage. The OCSA bit should be set while the ADCSR.ADST bit is 0.

When executing the A/D conversion of the internal reference voltage, the ADDISCR register is set to 0Fh and the A/D converter executes discharge (15 ADCLK). After executing discharge, the A/D converter executes sampling.The minimum sampling time is 5 µs. The A/D converter executes discharge each time it executes A/D conversion of the internal reference voltage.

39.2.14 A/D Sampling State Register n (ADSSTRn) (n = 00 to 15, L, T, O)

Address(es): ADC140.ADSSTR00 4005 C0E0h to ADC140.ADSSTR15 4005 C0EFh, ADC140.ADSSTRL 4005 C0DDh, ADC140.ADSSTRT 4005 C0DEh, ADC140.ADSSTRO 4005 C0DFh

	b7	b6	b5	b4	b3	b2	b1	b0
	SST[7:0]							
Value after reset:	0	0	0	0	1	1	0	1

Bit	Symbol	Bit name	Description	R/W
b7 to b0	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.

If one state is one ADCLK (A/D conversion clock) cycle and the ADCLK clock is 64 MHz, one state is 15.625 ns. The initial value is 13 states. 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 sampling time can be adjusted. The SST[7:0] bits should be set while the ADCSR.ADST bit is 0. The lower limit of the sampling time setting depends on the frequency ratio:

- Frequency ratio of PCLKB to PCLKC (ADCLK) = 1:1, 2:1, 4:1, or 8:1, the sampling time must be set to a value more than 5 states.
- Frequency ratio of PCLKB to PCLKC (ADCLK) = 1:2 or 1:4, the sampling time must be set to a value more than 6 states.

Table 39.8 shows the association between the A/D sampling state register and the relevant channels. For details, see section 39.3.6, Analog Input Sampling and Scan Conversion Time.

Table 39.8 Association between A/D sampling state register and relevant channels

Bit name	Associated channels
ADSSTR00.SST[7:0] bits*1	AN000
ADSSTR01.SST[7:0] bits	AN001
ADSSTR02.SST[7:0] bits	AN002
ADSSTR03.SST[7:0] bits	AN003
ADSSTR04.SST[7:0] bits	AN004
ADSSTR05.SST[7:0] bits	AN005
ADSSTR06.SST[7:0] bits	AN006
ADSSTR07.SST[7:0] bits	AN007
ADSSTR08.SST[7:0] bits	AN008
ADSSTR09.SST[7:0] bits	AN009
ADSSTR10.SST[7:0] bits	AN010
ADSSTR11.SST[7:0] bits	AN011
ADSSTR12.SST[7:0] bits	AN012
ADSSTR13.SST[7:0] bits	AN013
ADSSTR14.SST[7:0] bits	AN014
ADSSTR15.SST[7:0] bits	AN015
ADSSTRL.SST[7:0] bits	AN016-AN027
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 ADSSTR00.SST[7:0] is applied to it.

Note 2. When the temperature sensor output or the internal reference voltage is converted, set the sampling time more than 5µs. Because this register can be set only as high as 255, the ADCLK frequency must be such that the resulting sampling time is at least 5 µs when the temperature sensor output or the internal reference voltage is converted. For example, when ADCLK is 64 MHz, the sampling time does not reach 5 µs even if this register is set to 255.)

39.2.15 A/D Disconnection Detection Control Register (ADDISCR)

Address(es): ADC140.ADDISCR 4005 C07Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PCHG	ADNDIS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	ADNDIS[3:0]	Precharge/Discharge Period	b3 to b0 0000: The disconnection detection assist function is disabled. 0001: Setting prohibited. Others: The number of states for the period of discharge or precharge.	R/W
b4	PCHG	Precharge/Discharge Select	0: Discharge 1: Precharge.	R/W
b7 to b5	—	Reserved	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 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. The operation is achieved by automatically setting the ADDISCR register to 0Fh (15 ADCLK) when ADEXICR.OCSA or TSSA is set to 1. After executing discharge, the A/D converter executes sampling. The required sampling time is 5 µs or more. If any of the following functions are used, the disconnection detection assist function should be disabled.

- The temperature sensor
- The internal reference voltage
- A/D self-diagnosis.

ADNDIS[3:0] bits (Precharge/Discharge Period)

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 assistance function is enabled.

PCHG bits (Precharge/Discharge Select)

Setting the PCHG bit to 1 selects Precharge and setting the PCHG bit to 0 selects Discharge.

39.2.16 A/D Group Scan Priority Control Register (ADGSPCR)

Address(es): ADC140.ADGSPCR 4005 C080h

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

Bit	Symbol	Bit name	Description	R/W
b0	PGS	Group A Priority Control Setting*1	0: Operation without group A priority control 1: Operation with group A priority control.	R/W
b1	GBRSCN	Group B Restart Setting	Enabled only when PGS = 1. Reserved when PGS = 0. 0: Scanning for group B is not restarted after having been discontinued due to group A priority control 1: Scanning for group B is restarted after having been discontinued due to group A priority control.	R/W
b14 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	GBRP	Group B Single Scan Continuous Start*2	Enabled only when PGS = 1. Reserved when PGS = 0. 0: Single scan for group B is not continuously activated 1: Single scan for group B is continuously activated.	R/W

Note 1. When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group-scan mode). If the bits are set to any other values, proper operation is not guaranteed.

Note 2. When the GBRP bit is set to 1, single-scan is performed continuously for group B regardless of the setting of the GBRSCN bit.

PGS bit (Group A Priority Control Setting)

Set this bit to 1 when giving priority to operation on group A.

When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group-scan mode). If the bits are set to any other values, proper operation is not guaranteed.

When the PGS bit is set to 0, a clear operation must be performed by software according to section 39.8.2, Notes on Stopping A/D Conversion. When the PGS bit is set to 1, use settings according to section 39.3.4.3, Operation under Group A Priority Control.

GBRSCN bit (Group B Restart Setting)

This bit controls the restarting of scan operation on group B when operation on group A is given priority.

If a scan operation on group B is stopped by a group A trigger input with the GBRSCN bit set to 1, the scan operation is restarted on completion of the A/D conversion on group A. Also, if a group B trigger is input during A/D conversion on group A, the scan operation on group B is restarted on completion of the A/D conversion on group A.

If the GBRSCN bit is set to 0, triggers that are input during A/D conversion are ignored. Also, the ADCSR.ADST bit must be 0 when the GBRSCN bit is to be set.

The setting of the GBRSCN bit is valid when the PGS bit is 1.

GBRP bit (Group B Single Scan Continuous Start)

This bit is set when a single-scan operation is to be performed continuously on group B.

Setting the GBRP bit to 1 starts a single scan on group B. On completion of the scan, another single scan on group B is automatically started. If an A/D conversion on group B is stopped by an operation on group A, the group A operation takes priority, and single scan on group B is automatically restarted on completion of the A/D conversion on group A.

Disable group B trigger input before setting the GBRP bit to 1. Setting the GBRP bit to 1 invalidates the setting of the GBRSCN bit. The ADCSR.ADST bit must be 0 when the GBRP bit is to be set. The setting of the GBRP bit is valid when the PGS bit is 1.

39.2.17 A/D Compare Function Control Register (ADCMPCR)

Address(es): ADC140.ADCMPCR 4005 C090h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit name	Description	R/W
b1, b0	CMPAB[1:0]	Window A/B Composite Conditions Setting	b1 b0 0 0: ADC140_WCMPM is output when window A comparison conditions are met OR window B comparison conditions are met. ADC140_WCMPUM is output in other cases. 0 1: ADC140_WCMPM is output when window A comparison conditions are met EXOR window B comparison conditions are met. ADC140_WCMPUM is output in other cases. 1 0: ADC140_WCMPM is output when window A comparison conditions are met AND window B comparison conditions are met. ADC140_WCMPUM is output in other cases. 1 1: Setting prohibited. These bits are valid when both window A and window B are enabled (CMPAE = 1 and CMPBE = 1).	R/W
b8 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	CMPBE	Compare Window B Operation Enable	0: Compare window B operation is disabled. ADC140_WCMPM and ADC140_WCMPUM outputs are disabled. 1: Compare window B operation is enabled.	R/W
b10	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b11	CMPAE	Compare Window A Operation Enable	0: Compare window A operation is disabled. ADC140_WCMPM and ADC140_WCMPUM outputs are disabled. 1: Compare window A operation is enabled.	R/W
b12	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13	CMPBIE	Compare B Interrupt Enable	0: ADC140_CMPBI interrupt is disabled when comparison conditions (window B) are met. 1: ADC140_CMPBI interrupt is enabled when comparison conditions (window B) are met.	R/W
b14	WCMPE	Window Function Setting	0: Window function is disabled. 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: Window function is enabled. 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
b15	CMPAIE	Compare A Interrupt Enable	0: ADC140_CMPAI interrupt disabled when comparison conditions (window A) are met 1: ADC140_CMPAI interrupt enabled when comparison conditions (window A) are met.	R/W

CMPAB[1:0] bits (Window A/B Composite Conditions Setting)

These bits are valid when both window A and window B are enabled (CMPAE = 1 and CMPBE = 1) in single-scan mode. These bits are used to select compare function match/mismatch event output conditions and monitoring conditions of ADWINMON.MONCONB. Set the CMPAB[1:0] bits while the ADCSR.ADST bit is 0.

CMPBE bit (Compare Window B Operation Enable)

This 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 in the A/D conversion extended input control register (ADEXICR.{OCSA, TSSA})
- CMPCHB[5:0] in the window B channel select register (ADCMPBNSR.CMPCHB[5:0]).

CMPAE bit (Compare Window A Operation Enable)

This 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)
- OCSA or TSSA in the A/D conversion extended input control register (ADEXICR.{OCSA, TSSA})
- Window A channel select registers 0/1 (ADCMPANSR0, ADCMPANSR1)
- Window A extended input select register (ADCMPANSER).

CMPBIE bit (Compare B Interrupt Enable)

This bit enables or disables the interrupt output ADC140_CMPBI when the comparison conditions (window B) are met.

WCMPE bit (Window Function Setting)

This bit enables or disables the window function. Set the WCMPE bit while the ADCSR.ADST bit is 0.

CMPAIE bit (Compare A Interrupt Enable)

This bit enables or disables the interrupt output ADC140_CMPAI when the comparison conditions (window A) are met.

39.2.18 A/D Compare Function Window A Channel Select Register 0 (ADCMPANSR0)

Address(es): ADC140.ADCMPANSR0 4005 C094h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPCHA15	CMPCHA14	CMPCHA13	CMPCHA12	CMPCHA11	CMPCHA10	CMPCHA09	CMPCHA08	CMPCHA07	CMPCHA06	CMPCHA05	CMPCHA04	CMPCHA03	CMPCHA02	CMPCHA01	CMPCHA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	CMPCHA15 to CMPCHA00	Compare Window A Channel Select	0: Compare function disabled for associated input channel. 1: Compare function enabled for associated input channel. Bit 15 (CMPCHA15) corresponds to AN015 and bit 0 (CMPCHA00) corresponds to AN000.	R/W

CMPCHAn bits (n = 00 to 15) (Compare Window A Channel Select)

The compare function is enabled by writing 1 to the CMPCHAn bit of the same number as the A/D conversion channel selected by the ADANSA0.ANSAn bits (n = 00 to 15) and the ADANSB0.ANSBn bits (n = 00 to 15).

Set the CMPCHAn bits while the ADCSR.ADST bit is 0.

39.2.19 A/D Compare Function Window A Channel Select Register 1 (ADCMPANSR1)

Address(es): ADC140.ADCMPANSR1 4005 C096h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPCHA27	CMPCHA26	CMPCHA25	CMPCHA24	CMPCHA23	CMPCHA22	CMPCHA21	CMPCHA20	CMPCHA19	CMPCHA18	CMPCHA17	CMPCHA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b11 to b0	CMPCHA27 to CMPCHA16	Compare Window A Channel Select	0: Compare function disabled for associated input channel 1: Compare function enabled for associated input channel. Bit 11 (CMPCHA27) corresponds to AN027 and bit 0 (CMPCHA16) corresponds to AN016.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPCHAn bits (n = 16 to 27) (Compare Window A Channel Select)

The compare function is enabled by writing 1 to the CMPCHAn bit of the same number as the A/D conversion channel selected by the ADANSA1.ANSAn bits (n = 16 to 27) and the ADANSB1.ANSBn bits (n = 16 to 27).

Set the CMPCHAn bits while the ADCSR.ADST bit is 0.

39.2.20 A/D Compare Function Window A Extended Input Select Register (ADCMPANSER)

Address(es): ADC140.ADCMPANSER 4005 C092h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPOCA	CMPTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CMPTSA	Temperature Sensor Output Compare Select	0: Excludes the temperature sensor output from the compare window A target range 1: Includes the temperature sensor output in the compare window A target range.	R/W
b1	CMPOCA	Internal Reference Voltage Compare Select	0: Excludes the internal reference voltage from the compare window A target range 1: Includes the internal reference voltage in the compare window A target range.	R/W
b7 to b2	—	Reserved	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 while the ADEXICR.TSSA bit is 1. Set the CMPTSA bit while 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 while the ADEXICR.OCSA bit is 1. Set the CMPOCA bit while the ADCSR.ADST bit is 0.

39.2.21 A/D Compare Function Window A Comparison Condition Setting Register 0 (ADCMPLR0)

Address(es): ADC140.ADCMPLR0 4005 C098h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPLCHA15	CMPLCHA14	CMPLCHA13	CMPLCHA12	CMPLCHA11	CMPLCHA10	CMPLCHA09	CMPLCHA08	CMPLCHA07	CMPLCHA06	CMPLCHA05	CMPLCHA04	CMPLCHA03	CMPLCHA02	CMPLCHA01	CMPLCHA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	CMPLCHA15 to CMPLCHA00	Compare Window A Comparison Condition Select	These bits set comparison conditions of channels AN000 to AN015 to which window A comparison conditions are applied. Comparison conditions are shown in Figure 39.3. When the window function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 value > A/D converted value 1: ADCMPDR0 value < A/D converted value. When the window function is enabled (ADCMPCR.WCMPE bit = 1): 0: (A/D converted value < ADCMPDR0 value) or (ADCMPDR1 value < A/D converted value) 1: ADCMPDR0 value < A/D converted value < ADCMPDR1 value.	R/W

CMPLCHAn bits (n = 00 to 15) (Compare Window A Comparison Condition Select)

These bits are used to set comparison conditions of channels AN000 to AN015 to which window A comparison conditions are applied. These bits can be set for each analog input to be compared. CMPLCHA00, CMPLCHA07, and CMPLCHA15 correspond to AN000, AN007, and AN015, respectively. When the comparison result of each analog input meets the set condition, the ADCMPSR0.CMPSTCHAn bit is set to 1 and a compare interrupt (ADC140_CMPAI) is generated.

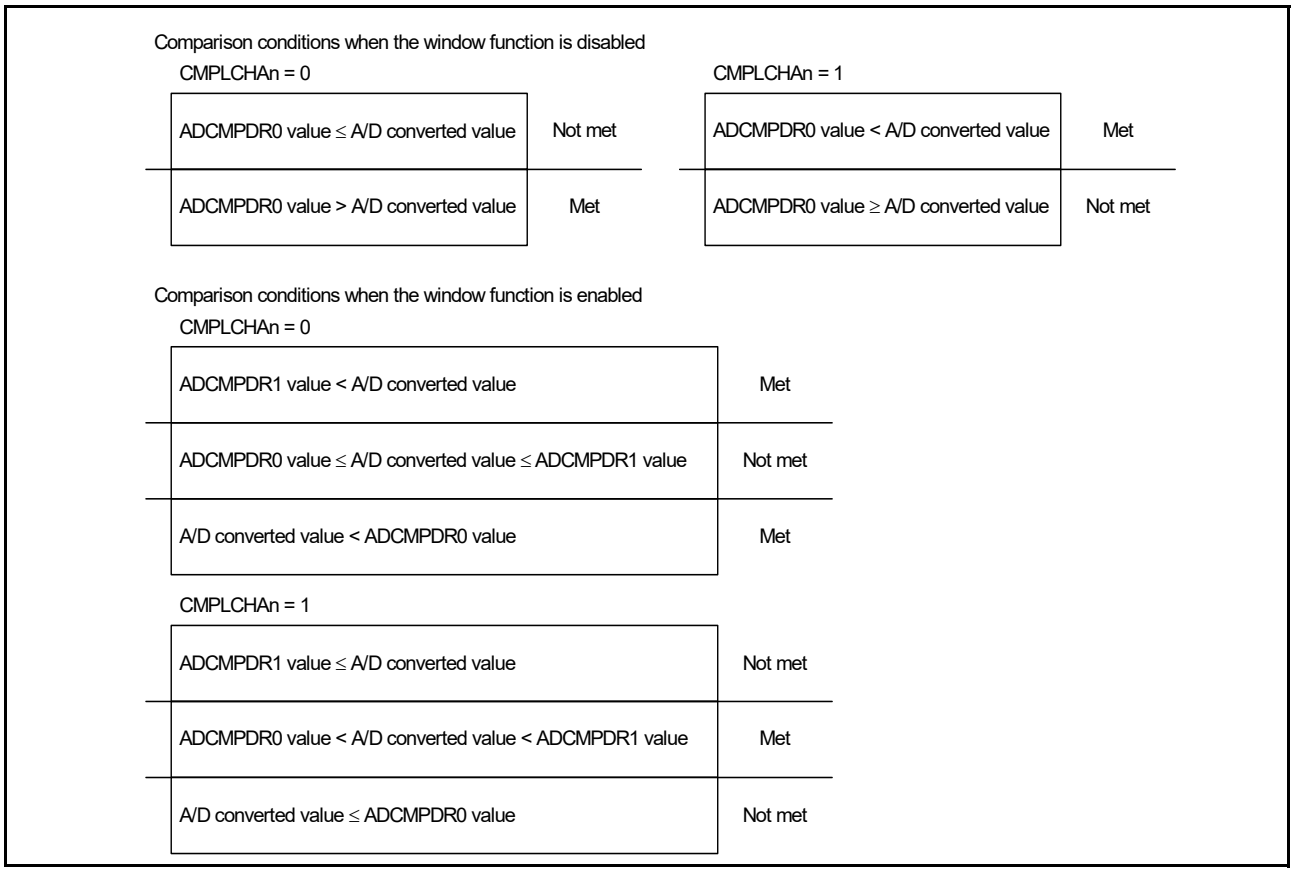


Figure 39.3 Explanation of compare function window A comparison conditions

39.2.22 A/D Compare Function Window A Comparison Condition Setting Register 1 (ADCMPLR1)

Address(es): ADC140.ADCMPLR1 4005 C09Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPLCHA27	CMPLCHA26	CMPLCHA25	CMPLCHA24	CMPLCHA23	CMPLCHA22	CMPLCHA21	CMPLCHA20	CMPLCHA19	CMPLCHA18	CMPLCHA17	CMPLCHA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b11 to b0	CMPLCHA27 to CMPLCHA16	Compare Window A Comparison Condition Select	These bits set comparison conditions of channels AN016 to AN027 to which window A comparison conditions are applied. Comparison conditions are shown in Figure 39.3. When the window function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 value > A/D converted value 1: ADCMPDR0 value < A/D converted value. When the window function is enabled (ADCMPCR.WCMPE bit = 1): 0: A/D converted value < ADCMPDR0 value or ADCMPDR1 value < A/D converted value 1: ADCMPDR0 value < A/D converted value < ADCMPDR1 value.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R

CMPLCHAn bits (n = 16 to 27) (Compare Window A Comparison Condition Select)

These bits are used to set comparison conditions of channels AN016 to AN027 to which window A comparison conditions are applied. These bits can be set for each analog input to be compared. CMPLCHA16, CMPLCHA23, and CMPLCHA27 correspond to AN016, AN023, and AN027, respectively. When the comparison result of each analog input meets the set condition, the ADCMPSR1.CMPSTCHAn bit is set to 1 and a compare interrupt (ADC140_CMPAI) is generated.

39.2.23 A/D Compare Function Window A Extended Input Comparison Condition Setting Register (ADCMPLER)

Address(es): ADC140.ADCMPLER 4005 C093h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPLOCA	CMPLTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CMPLTSA	Compare Window A Temperature Sensor Output Comparison Condition Select	Comparison conditions are shown in Figure 39.3. When the window A function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 value > A/D converted value 1: ADCMPDR0 value < A/D converted value. When the window A function is enabled (ADCMPCR.WCMPE bit = 1): 0: (A/D converted value < ADCMPDR0 value) or (A/D converted value > ADCMPDR1 value) 1: ADCMPDR0 value < A/D converted value < ADCMPDR1 value.	R/W
b1	CMPLOCA	Compare Window A Internal Reference Voltage Comparison Condition Select	Comparison conditions are shown in Figure 39.3. When window A function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADCMPDR0 register value > A/D-converted value 1: ADCMPDR0 register value < A/D-converted value. When window A function is enabled (ADCMPCR.WCMPE bit = 1): 0: A/D-converted value < ADCMPDR0 register value or A/D-converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value.	R/W
b7 to b2	—	Reserved	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)

This bit is used to set comparison conditions when the temperature sensor output is the target of the window A comparison condition.

When the temperature sensor output comparison result meets the set condition, the ADCMPSER.CMPSTTSA bit is set to 1 and a compare interrupt (ADC140_CMPAI) is generated.

CMPLOCA bit (Compare Window A Internal Reference Voltage Comparison Condition Select)

This bit is used to set comparison conditions when the internal reference voltage is the target of the window A comparison condition.

When the internal reference voltage comparison result meets the set condition, the ADCMPSER.CMPSTOCA bit is set to 1 and a compare interrupt (ADC140_CMPAI) is generated.

39.2.24 A/D Compare Function Window A Lower-Side Level Setting Register (ADCMPDR0), A/D Compare Function Window A Upper-Side Level Setting Register (ADCMPDR1), A/D Compare Function Window B Lower-Side Level Setting Register (ADWINLLB), A/D Compare Function Window B Upper-Side Level Setting Register (ADWINULB)

Address(es): ADC140.ADCMPDR0 4005 C09Ch, ADC140.ADCMPDR1 4005 C09Eh, ADC140.ADWINLLB 4005 C0A8h, ADC140.ADWINULB 4005 C0AAh

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

Bit	Symbol	Bit name	Description	R/W
b15 to b0	—	—	Reference value	R/W

The ADCMPDRy (y = 0, 1) register sets the reference data when the compare window A function is used. ADCMPDR0 sets the lower-side reference of window A, and ADCMPDR1 sets the upper-side reference of window A.

ADWINULB and ADWINLLB set the reference data when the compare window B function is used. ADWINLLB sets the lower-side reference of window B, and ADWINULB sets the upper-side reference of window B. The ADCMPDRy, ADWINULB, and ADWINLLB are readable and writable.

ADCMPDRy, ADWINULB, and ADWINLLB are writable even during A/D conversion. The reference data can be dynamically modified by rewriting register values during A/D conversion.*1

Set these registers so that the upper-side reference is not less than the lower-side reference (ADCMPDR1 ≥ ADCMPDR0, ADWINULB ≥ ADWINLLB). ADCMPDR1 and ADWINULB are not used when the window function is disabled.

Note 1. The lower-side and the upper-side references are changed when each register is written. For example, when the upper-side reference value has been changed and the lower-side reference value is being changed, the MCU compares the upper-side reference (after rewrite), and the lower-side reference (before rewrite) with the A/D conversion result. See Figure 39.4. 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 (ADCMPCR.CMPAE or ADCMPCR.CMPBE) is 0.

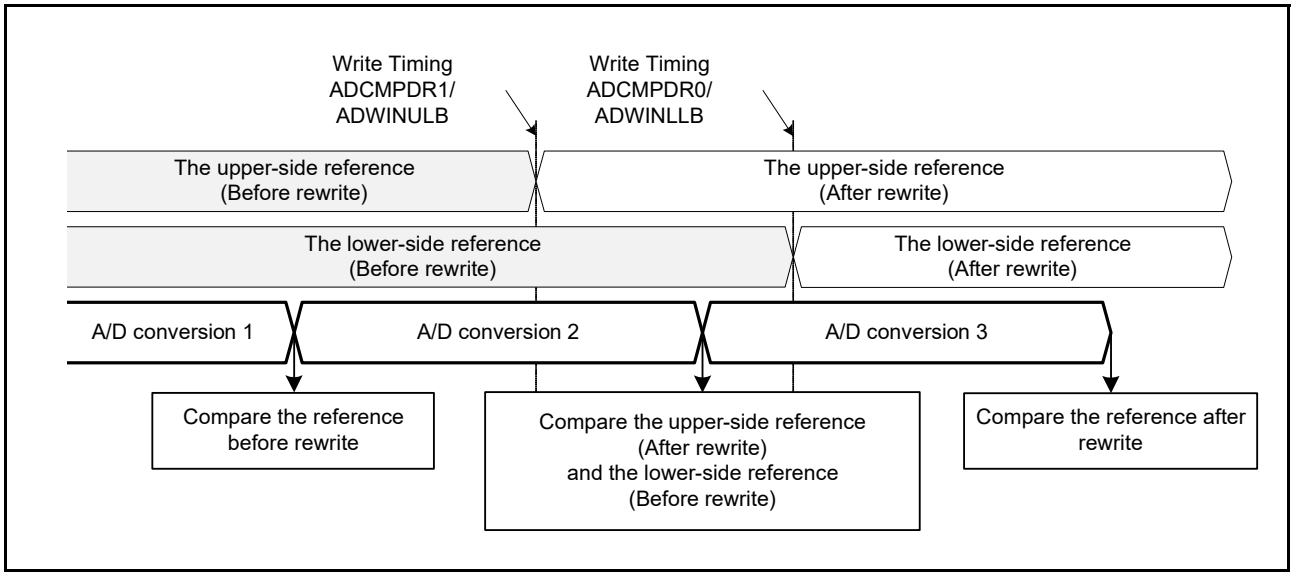


Figure 39.4 Comparison between upper-side reference and lower-side reference before and after a rewrite

The ADCMPDRy, ADWINLLB, and ADWINULB registers use different formats depending on the following conditions:

- The value of the A/D data register format select bit (flush-right or flush-left)
- The value of the A/D-conversion accuracy specification bit (14-bit or 12-bit)
- The value of the A/D-converted value addition/average channel select register (A/D-converted value addition mode selected or not selected).

The data formats for each given condition are as follows:

(1) When A/D-Converted Value Addition Mode is Not Selected

- Flush-right data with 14-bit accuracy: Lower 14 bits (b13 to b0) are valid
- Flush-right data with 12-bit accuracy: Lower 12 bits (b11 to b0) are valid
- Flush-left data with 14-bit accuracy: Upper 14 bits (b15 to b2) are valid
- Flush-left data with 12-bit accuracy: Upper 12 bit (b15 to b4) are valid.

(2) When A/D-Converted Value Addition Mode is Selected

- Flush-right data with 14-bit accuracy: All bits (b15 to b0) are valid
- Flush-right data with 12-bit accuracy: Lower 14 bits (b13 to b0) are valid
- Flush-left data with 14-bit accuracy: All bits (b15 to b0) are valid
- Flush-left data with 12-bit accuracy: Upper 14 bits (b15 to b2) are valid.

39.2.25 A/D Compare Function Window A Channel Status Register 0 (ADCMPSR0)

Address(es): ADC140.ADCMPSR0 4005 C0A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPSTCHA15	CMPSTCHA14	CMPSTCHA13	CMPSTCHA12	CMPSTCHA11	CMPSTCHA10	CMPSTCHA09	CMPSTCHA08	CMPSTCHA07	CMPSTCHA06	CMPSTCHA05	CMPSTCHA04	CMPSTCHA03	CMPSTCHA02	CMPSTCHA01	CMPSTCHA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b15 to b0	CMPSTCHA15 to CMPSTCHA00	Compare Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1b), these bits indicate the comparison result of channels AN000 to AN015 to which window A comparison conditions are applied: 0: Comparison conditions are not met 1: Comparison conditions are met.	R/W

CMPSTCHAn bits (n = 00 to 15) (Compare Window A Flag)

These bits are comparison result status flags of channels AN000 to AN015 to which window A comparison conditions are applied. When the comparison condition set by ADCMPLR0.CMPLCHAn is met at the end of A/D conversion, the corresponding bit is set to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt (ADC140_CMPAI) request is generated when this flag is set to 1. CMPSTCHA00, CMPSTCHA07, and CMPSTCHA15 correspond to AN000, AN007, and AN015, respectively.

Writing 1 to the CMPSTCHAn bits is disabled.

[Setting condition]

- The condition set in ADCMPLR0.CMPLCHAn is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

39.2.26 A/D Compare Function Window A Channel Status Register1 (ADCMPSR1)

Address(es): ADC140.ADCMPSR1 4005 C0A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPSTCHA27	CMPSTCHA26	CMPSTCHA25	CMPSTCHA24	CMPSTCHA23	CMPSTCHA22	CMPSTCHA21	CMPSTCHA20	CMPSTCHA19	CMPSTCHA18	CMPSTCHA17	CMPSTCHA16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b11 to b0	CMPSTCHA27 to CMPSTCHA16	Compare Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1), these bits indicate the comparison result of channels AN016 to AN027 to which window A comparison conditions are applied: 0: Comparison conditions are not met 1: Comparison conditions are met.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPSTCHAn bits (n = 16 to 27) (Compare Window A Flag)

These bits are comparison result status flags of channels AN016 to AN027 to which window A comparison conditions are applied. When the comparison condition set by ADCMPLR1.CMPLCHAn is met at the end of A/D conversion, the corresponding is set to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt (ADC140_CMPAI) request is generated when this flag is set to 1. CMPSTCHA16, CMPSTCHA20, CMPSTCHA27 correspond to AN016, AN020, and AN027, respectively. Writing 1 to the CMPSTCHAn bits is disabled.

[Setting condition]

- The condition set by ADCMPLR1.CMPLCHAn is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after the 1 state is read.

39.2.27 A/D Compare Function Window A Extended Input Channel Status Register (ADCMPSER)

Address(es): ADC140.ADCMPSER 4005 C0A4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMPSTOCA	CMPSTTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	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
b1	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
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPSTTSA bit (Compare Window A Temperature Sensor Output Compare Flag)

This bit is a status flag that indicates the temperature sensor output comparison result. When the comparison condition set by ADCMPLER.CMPLTSA is met at the end of A/D conversion, this bit is set to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt (ADC140_CMPAI) request is generated when this flag is set to 1.

Writing 1 to the CMPSTTSA bit is disabled.

[Setting condition]

- The condition set by ADCMPLER.CMPLTSA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after the 1 state is read.

CMPSTOCA bit (Compare Window A Internal Reference Voltage Compare Flag)

This bit is a status flag that indicates the internal reference voltage comparison result. When the comparison condition set by ADCMPLER.CMPLOCA is met at the end of A/D conversion, this bit is set to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt (ADC140_CMPAI) request is generated when this flag is set to 1.

Writing 1 to the CMPSTOCA bit is disabled.

[Setting condition]

- The condition set by ADCMPLER.CMPLOCA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after the 1 state is read.

39.2.28 A/D Compare Function Window B Channel Select Register (ADCMPBNSR)

Address(es): ADC140.ADCMPBNSR 4005 C0A6h

	b7	b6	b5	b4	b3	b2	b1	b0
	CMPLB	—	CMPCHB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	CMPCHB[5:0]	Compare Window B Channel Select	These bits select channels to be compared with the compare window B conditions: b5 b0 0 0 0 0 0 0: AN000 0 0 0 0 0 1: AN001 0 0 0 0 1 0: AN002 : : 0 1 1 0 0 1: AN025 0 1 1 0 1 0: AN026 0 1 1 0 1 1: AN027 1 0 0 0 0 0: Temperature sensor 1 0 0 0 0 1: Internal reference voltage 1 1 1 1 1 1: Not select. Other settings are prohibited.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	CMPLB	Compare Window B Comparison Condition Setting	This bit sets comparison conditions of channels for window B. The comparison conditions are shown in Figure 39.5. When the window function is disabled (ADCMPCR.WCMPE bit = 0): 0: ADWINLLB value > A/D converted value 1: ADWINLLB value < A/D converted value. When the window function is enabled (ADCMPCR.WCMPE bit = 1): 0: (A/D converted value < ADWINLLB value) or (ADWINULB value < A/D converted value) 1: ADWINLLB value < A/D converted value < ADWINULB value.	R/W

CMPCHB[5:0] bits (Compare Window B Channel Select)

These bits are used to select channels to be compared with the compare window B conditions. Selections available are AN000 to AN027, the temperature sensor, and the internal reference voltage. The compare window B function is enabled by specifying the hexadecimal number of the A/D conversion channel selected by ADANSA0.ANSAn bits (n = 0 to 15), ADANSA1.ANSAn bits (n = 16 to 27), ADANSB0.ANSBn bits (n = 0 to 15), and ADANSB1.ANSBn bits (n = 16 to 27).

Set CMPCHB[5:0] bits while the ADCSR.ADST bit is 0.

CMPLB bit (Compare Window B Comparison Condition Setting)

This bit is used to set comparison conditions of channels for window B. When the comparison result of each analog input meets the set condition, the ADCMPBSR.CMPSTB bit is set to 1 and a compare interrupt (ADC140_CMPBI) request is generated.

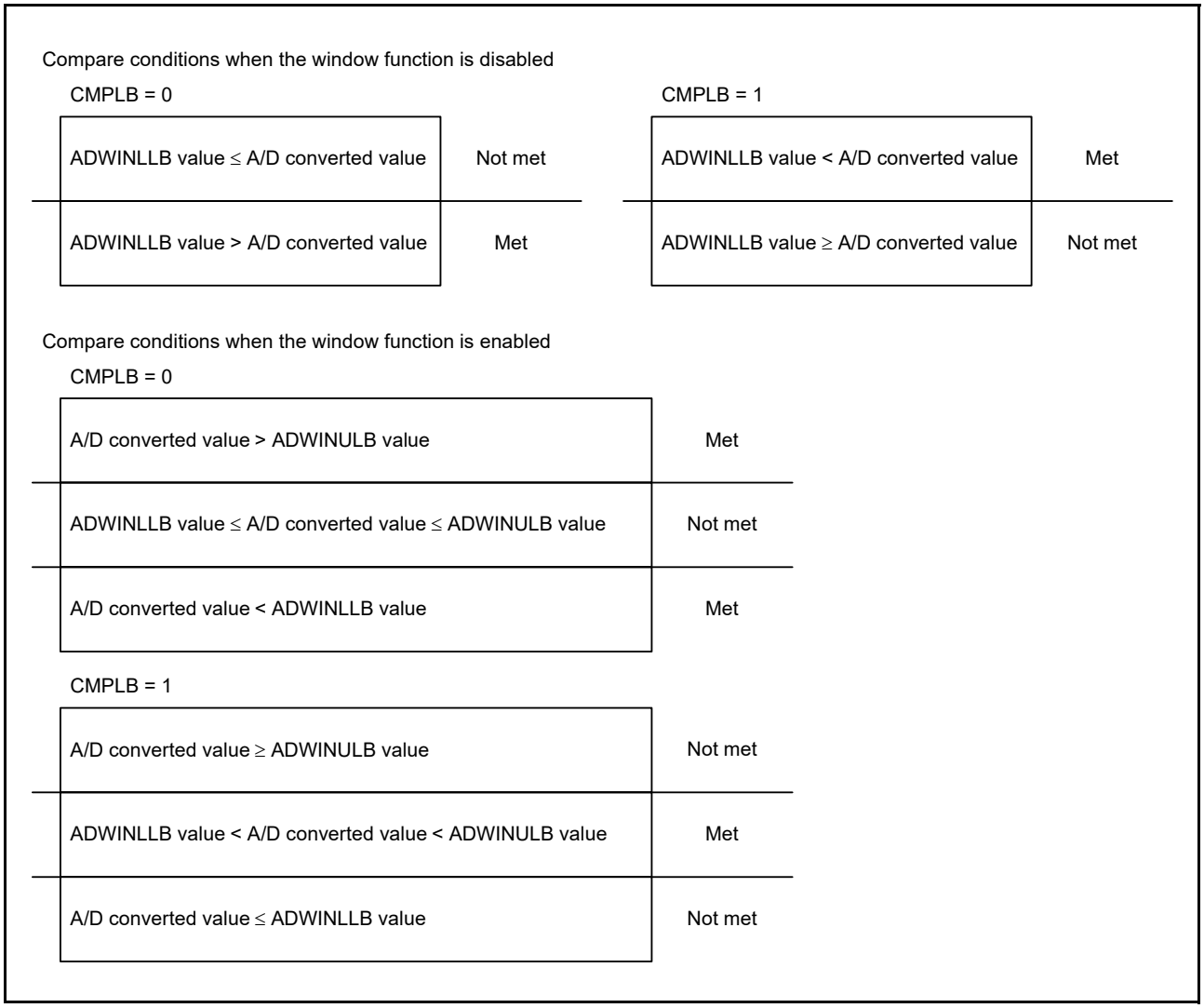


Figure 39.5 Explanation of compare function window B compare conditions

39.2.29 A/D Compare Function Window B Status Register (ADCMPBSR)

Address(es): ADC140.ADCMPBSR 4005 C0ACh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPSTB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CMPSTB	Compare Window B Flag	When window B operation is enabled (ADCMPCR.CMPBE = 1), this bit indicates the comparison result of channels AN000 to AN027, temperature sensor output, and internal reference voltage to which window B comparison conditions are applied: 0: Comparison conditions are not met 1: Comparison conditions are met.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CMPSTB bit (Compare Window B Flag)

This bit is a status flag that indicates the comparison result of channels (AN000 to AN027, the temperature sensor, and the internal reference voltage) to which window B comparison conditions are applied. When the comparison condition set by ADCMPBNSR.CMPLB is met at the end of A/D conversion, this bit is set to 1. When the ADCMPCR.CMPBIE bit is 1, a compare interrupt (ADC140_CMPBI) request is generated when this flag is set to 1.

Writing 1 to the CMPSTB bit is disabled.

[Setting condition]

- The condition set by ADCMPBNSR.CMPLB is met when ADCMPCR.CMPBE = 1.

[Clearing condition]

- Writing 0 after the 1 state is read.

39.2.30 A/D Compare Function Window A/B Status Monitor Register (ADWINMON)

Address(es): ADC140.ADWINMON 4005 C08Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MONCMPB	MONCMPA	—	—	—	MONCOMB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	MONCOMB	Combination Result Monitor	This bit indicates the combination result and is valid when both window A operation and window B operation are enabled. 0: Window A/window B composite conditions are not met 1: Window A/window B composite conditions are met.	R
b3 to b1	—	Reserved	These bits are read as 0.	R
b4	MONCMPA	Comparison Result Monitor A	0: Window A comparison conditions are not met 1: Window A comparison conditions are met.	R
b5	MONCMPB	Comparison Result Monitor B	0: Window B comparison conditions are not met 1: Window B comparison conditions are met.	R
b7, b6	—	Reserved	These bits are read as 0.	R

MONCOMB bit (Combination Result Monitor)

This read-only bit indicates the result in combination of comparison condition result A and comparison result condition B with the combination condition set by the ADCMPCR.CMPAB[1:0] bits.

[Setting condition]

- The combined result meets the combination condition set by 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 by the ADCMPCR.CMPAB[1:0] bits.
- ADCMPCR.CMPAE = 0 or ADCMPCR.CMPBE = 0.

MONCMPA bit (Comparison Result Monitor A)

This read-only bit is read as 1 when the A/D converted value of the window A target channel meets the condition set by ADCMPLR0/ADCMPLR1 and ADCMPLER, and is read as 0 in other cases.

[Setting condition]

- The A/D converted value meets the condition set by ADCMPLR0.CMPLCHAn when ADCMPCR.CMPAE = 1.

[Clearing conditions]

- The A/D converted value does not meet the condition set by ADCMPLR0.CMPLCHAn 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)

This read-only bit is read as 1 when the A/D converted value of the window B target channel meets the condition set by the ADCMPBNSR.CMPLB bit, and is read as 0 in other cases.

[Setting condition]

- The A/D converted value meets the condition set by ADCMPBNSR.CMPLB when ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The A/D converted value does not meet the condition set by ADCMPBNSR.CMPLB when ADCMPCR.CMPBE = 1
- ADCMPCR.CMPBE = 0 (Automatically cleared when the ADCMPCR.CMPBE value changes from 1 to 0.)

39.2.31 A/D High-Potential/Low-Potential Reference Voltage Control Register (ADHVREFCNT)

Address(es): ADC140.ADHVREFCNT 4005 C08Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	ADSLP	—	—	LVSEL	—	—	HVSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	HVSEL[1:0]	High-Potential Reference Voltage Select	b1 b0 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: Internal node discharge. No reference voltage pin is selected.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	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
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ADSLP	Sleep	0: Normal operation 1: Standby state.	R/W

HVSEL[1:0] bits (High-Potential Reference Voltage Select)

These bits are used to set the high-potential reference voltage. AVCC0, VREFH0, or the internal reference voltage (1.45 V) is selectable as the high-potential reference voltage.

Before selecting the internal reference voltage by setting these bits to 10b, set HVSEL[1:0] = 11b to discharge the path of the high-potential reference voltage. After the discharge is completed, 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 channels AN000 to AN027, but A/D conversion of the internal reference voltage or the temperature sensor output is prohibited.

LVSEL bit (Low-Potential Reference Voltage Select)

This bit is used to set the low-potential reference voltage. AVSS0 or VREFL0 is selectable as the low-potential reference voltage.

ADSLP bit (Sleep)

This bit is used to transition 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 clearing this bit to 0. In addition, 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 39.8.8, ADHSC Bit Rewriting Procedure.

39.3 Operation

39.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels. A scan conversion is performed in three operating modes and two conversion modes:

The three operating modes are:

- Single-scan mode
- Continuous-scan mode
- Group-scan mode.

The two conversion modes are:

- High-speed A/D conversion mode
- Low-power A/D conversion mode.

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 the ADST bit in ADCSR is set to 0 from 1 by software. In group-scan mode, the selected channels of group A and the selected channels of group B are scanned once after scan is started according to the respective synchronous trigger (ELC).

In single-scan mode and continuous-scan mode, A/D conversion is performed for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n. In group-scan mode, A/D conversion is performed for ANn channels of group A selected by the ADANSA0 and ADANSA1 registers first, and then performed for ANn channels of group B selected by the ADANSB0 and ADANSB1 registers, respectively, starting from the channel with the smallest number n.

When self-diagnosis is selected, self-diagnosis is executed once at the beginning of each scan and one of the three voltages internally generated in the 14-bit ADC 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, single-scan mode should be used.

Double-trigger mode is to be used with single-scan mode or group-scan mode. With double-trigger mode enabled (ADCSR.DBLE is 1), A/D conversion data of a channel selected by the DBLANS[4:0] bits in ADCSR is duplicated only if the conversion is started by the synchronous trigger (ELC) selected by the TRSA[5:0] bits in ADSTRGR. Only group A can use the double-trigger mode in group-scan mode.

The extended operation of double-trigger mode means the A/D conversion operation is generated from the synchronous trigger combination. This trigger combination is selected by ADSTRGR.TRSA[5:0] in double-trigger mode.

In extended operation of double-trigger mode, in addition to normal double-trigger mode operation, A/D conversion data with odd number trigger (ELC_AD00) is stored into A/D data duplexing register A (ADDBLDRA), and A/D conversion data with even number triggers (ELC_AD01) is stored into A/D data duplexing register B (ADDBLDRB). In extended operation of double-trigger mode, when one of the trigger combination occurs at the same time, data duplexing register selection by the specified triggers does not work and A/D conversion data is stored into A/D data duplexing register B (ADDBLDRB). When one synchronous trigger is input during the A/D conversion started by another synchronous trigger, the trigger that is input during another A/D conversion is canceled.

39.3.2 Single-Scan Mode

39.3.2.1 Basic Operation

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 ADST bit in ADCSR is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (ELC), or an asynchronous trigger input, A/D conversion is performed for ANn channels selected by 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 into the associated A/D data register (ADDRy).
3. When A/D conversion of all the selected channels is completed, an ADC140_ADI interrupt request is generated (without register setting).
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. Then, the 14-bit ADC enters a wait state.

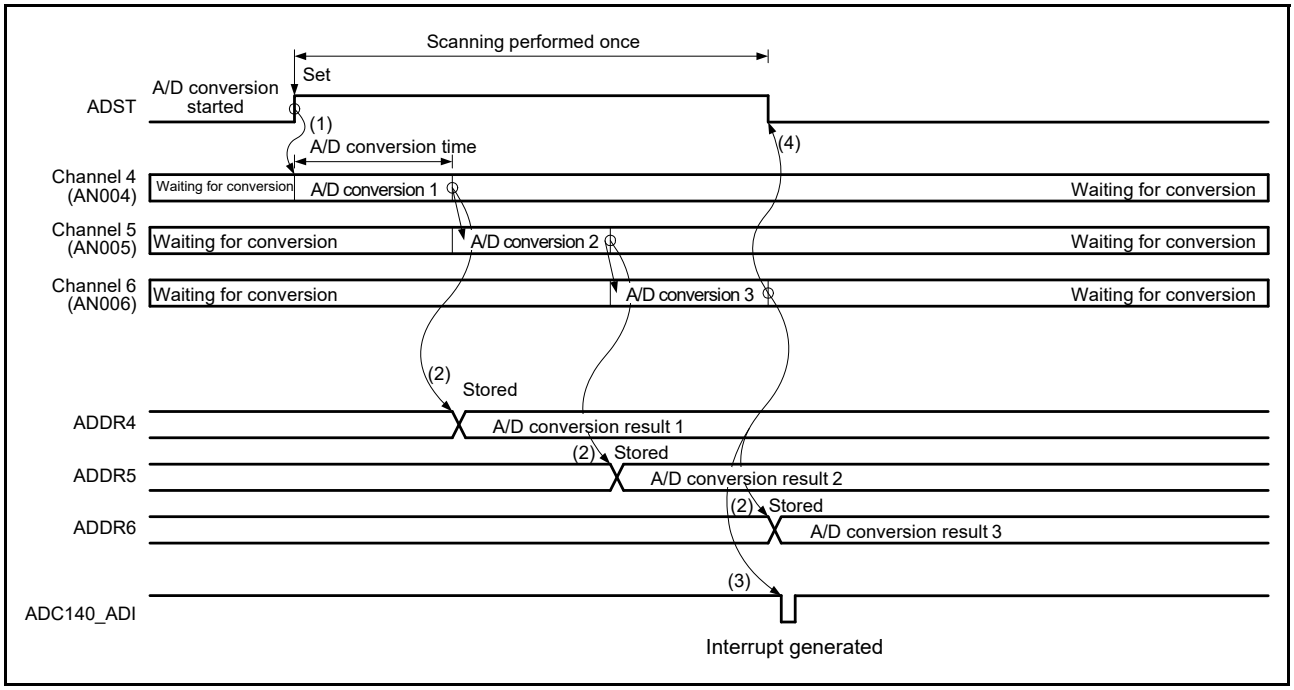


Figure 39.6 Example of operation in single-scan mode (basic operation: AN004 to AN006 selected)

39.3.2.2 Channel Selection and Self-Diagnosis

When channels and self-diagnosis are selected, A/D conversion is first performed for the reference voltage VREFH0 (x0, x1/2, or x1) supplied to the A/D converter, 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 ADST bit in ADCSR is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (ELC), or an asynchronous trigger input.
2. When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored into the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by 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 into the associated A/D data register (ADDRy).
4. When A/D conversion of all the selected channels is completed, an ADC140_ADI interrupt request is generated (without register setting).
5. 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. Then, the 14-bit ADC enters a wait state.

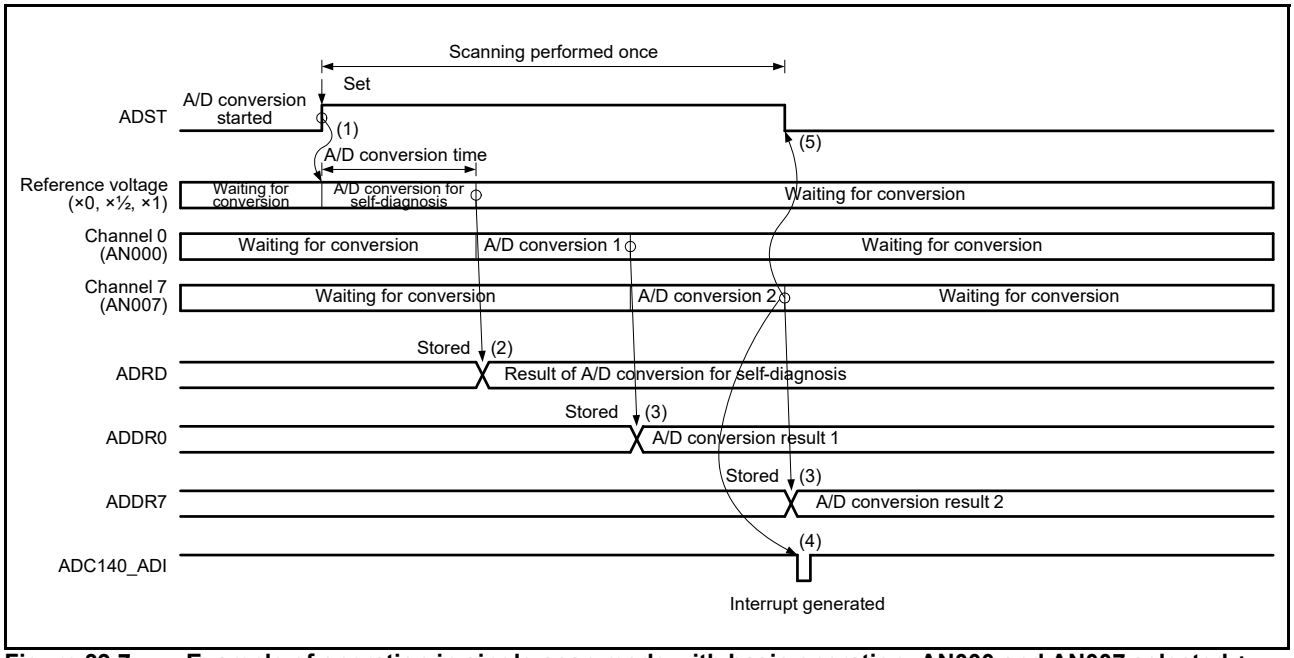


Figure 39.7 Example of operation in single-scan mode with basic operation: AN000 and AN007 selected + self-diagnosis

39.3.2.3 A/D Conversion of Temperature Sensor Output/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.

All channels should be deselected (by setting the ADANSA0 and ADANSA01 registers to all 0 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 (not selected). When selecting A/D conversion of internal reference voltage, set the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) to 0 (not selected).

1. Set the sampling time to 5 µs or longer. Take note of the settings of the sampling state registers (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 corresponding temperature sensor data register (ADTSDR) or A/D internal reference voltage data register (ADOCDR), and an ADC140_ADI interrupt request is generated (without register setting).
4. The ADST bit remains 1 during A/D conversion and is automatically set to 0 on completion of the A/D conversion. The 14-bit ADC enters a wait state.

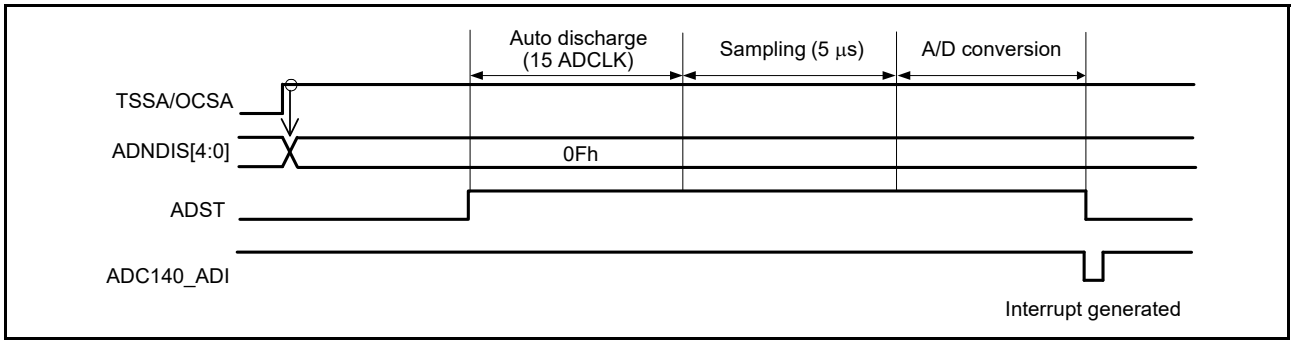


Figure 39.8 Example of operation in single-scan mode with basic operation: AN000 and temperature sensor output or internal reference voltage selected

39.3.2.4 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) are performed as a sequence as shown in this section.

Self-diagnosis should be deselected, and the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) must both be set to 0.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the DBLANS[4:0] bits in ADCSR and setting the DBLE bit in ADCSR to 1. When the DBLE bit in ADCSR is set to 1, channel selection using the ADANSA0 and ADANSA1 registers is invalid. In double-trigger mode, a synchronous trigger (ELC) should be selected using the TRSA[5:0] bits in ADSTRGR; the EXTRG bit and TRGE bit in ADCSR must be set to 0 and 1, respectively. Software trigger should not be used.

1. When the ADST bit in ADCSR is set to 1 (A/D conversion start) by a synchronous trigger input (ELC), A/D conversion is started on the single channel selected by the DBLANS[4:0] bits in ADCSR.
2. Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the associated A/D data register (ADDRy).
3. The ADST bit is automatically set to 0 and the 14-bit ADC enters a wait state. Here, an ADC140_ADI interrupt request is not generated.
4. When the ADST bit in ADCSR is set to 1 (A/D conversion start) by the second trigger input, A/D conversion is started on the single channel selected by the DBLANS[4:0] bits in ADCSR.
5. When A/D conversion is completed, the A/D conversion result is stored into the A/D data duplexing register (ADDBLDR), which is exclusively used in double-trigger mode.
6. An ADC140_ADI interrupt request is generated (without register setting).
7. The ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when A/D conversion is completed. The 14-bit ADC enters a wait state.

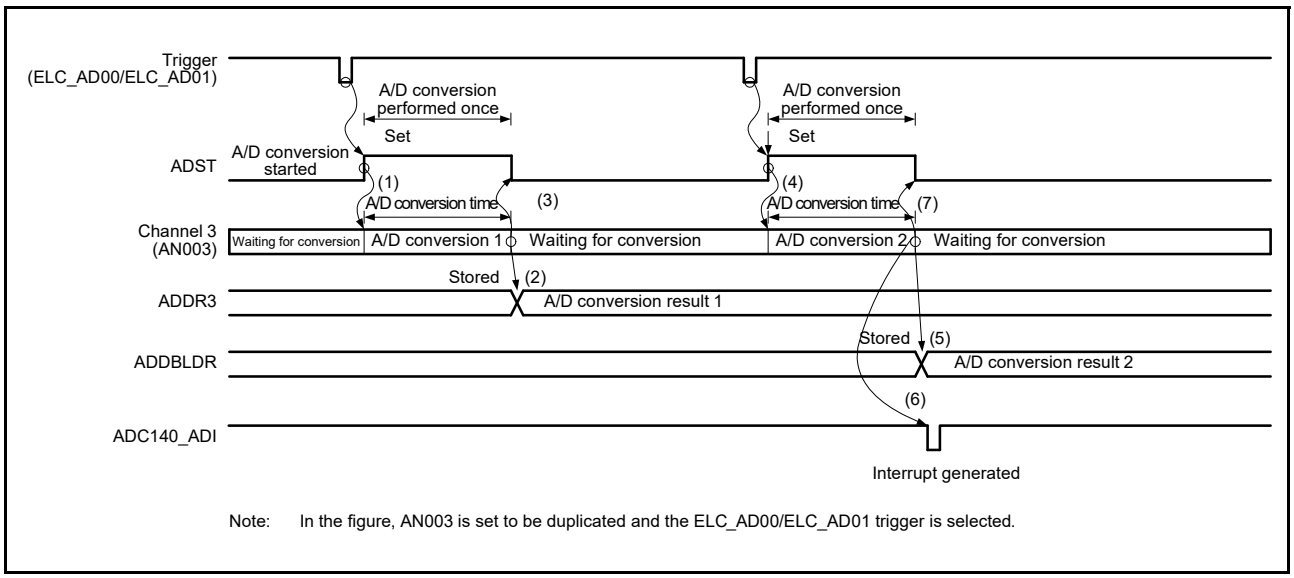


Figure 39.9 Example of operation in single-scan mode with double-trigger mode selected - an003 duplicated

39.3.2.5 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 as shown in this section.

Self-diagnosis should be deselected, and the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) must both be set to 0.

Duplication of A/D conversion data is enabled by setting the channel numbers 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, a synchronous trigger ELC_AD00/ELC_AD01 should be selected using the ADSTRGR.TRSA[5:0] bits (setting 0Bh to ADSTRGR.TRSA[5:0]), the ADCSR.EXTRG bit and ADCSR.TRGE bit must be set to 0 and 1, respectively. Software trigger must not be used.

1. A/D conversion for the single channel selected by the ADCSR.DBLANS[4:0] bits starts when the ELC_AD00/ ELC_AD01 input sets the ADCSR.ADST bit to 1 (starting A/D conversion).
2. When A/D conversion of a single channel completes, the conversion result is stored in the corresponding A/D data register (ADDRy) and in A/D Data Duplexing Register A (ADDBLDRA) or A/D Data Duplexing Register B (ADDBLDRB) when the trigger of ELC_AD00 or ELC_AD01 is input, respectively.
3. The ADCSR.ADST bit is automatically cleared and the 14-bit ADC enters a wait state. An ADC140_ADI interrupt is not generated.
4. When the ELC_AD00/ELC_AD01 input sets the ADCSR.ADST bit to 1 (starting A/D conversion), conversion for the single channel selected by the ADCSR.DBLANS[4:0] bits starts.
5. When A/D conversion completes, the 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 trigger of ELC_AD00 or ELC_AD01 is input, respectively.
6. An ADC140_ADI interrupt request is generated (without register setting).
7. The ADCSR.ADST bit retains the value 1 (starting A/D conversion) during A/D conversion and is automatically cleared on completion of conversion, after which the A/D converter enters a wait state.

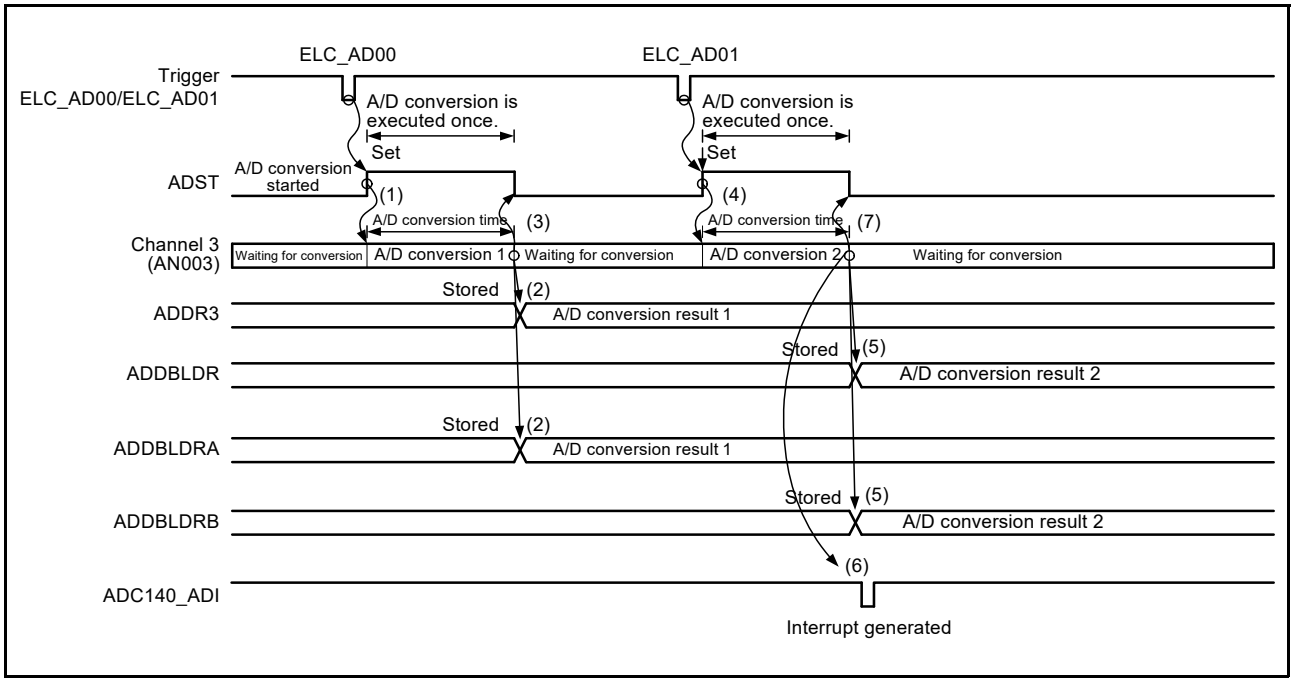


Figure 39.10 Example of extended operation in double-trigger mode (1) with duplication selected for AN003, ELC_AD00/ELC_AD01 selected

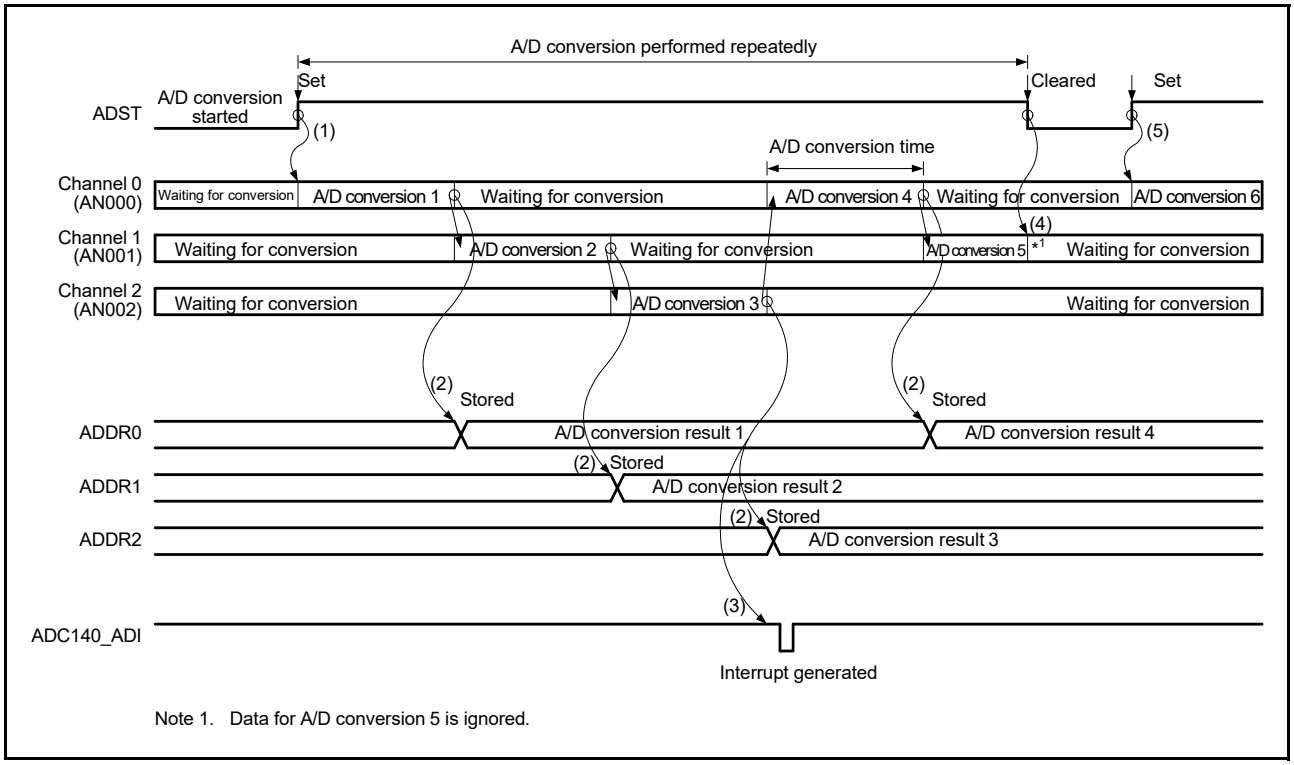
39.3.3 Continuous-Scan Mode

39.3.3.1 Basic Operation

In continuous-scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels as shown in this section.

In continuous-scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) must both be set to 0 (deselected).

1. When the ADST bit in ADCSR is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (ELC), or an asynchronous trigger input, A/D conversion is performed for ANn channels selected by 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 into the associated A/D data register (ADDRy).
3. When A/D conversion of all the selected channels is completed, an ADC140_ADI interrupt request is generated (without register setting). The 14-bit ADC sequentially starts A/D conversion for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
4. The ADST bit in ADCSR is not automatically cleared and steps 2 and 3 are repeated as long as the bit 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 14-bit ADC enters a wait state.
5. When the ADST bit is later set to 1 (A/D conversion start), A/D conversion is started again for ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.



Note 1. Data for A/D conversion 5 is ignored.

Figure 39.11 Example of operation in continuous-scan mode with basic operation: AN000 to AN002 selected

39.3.3.2 Channel Selection and Self-Diagnosis

When channels and self-diagnosis are selected at the same time, A/D conversion is first performed for the reference voltage VREFH0 (×0, ×1/2, or ×1) supplied to the 14-bit ADC, then A/D conversion is performed on the analog input of the selected channels. This sequence is repeated as shown in this section.

In continuous-scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) must both be set to 0 (deselected).

1. A/D conversion for self-diagnosis is first started when the ADST bit in ADCSR is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (ELC), or an asynchronous trigger input.
2. When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored into the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by 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 into the corresponding A/D data register (ADDRy).
4. When A/D conversion of all the selected channels is completed, an ADC140_ADI interrupt request is generated (without register setting). At the same time, the 14-bit ADC starts A/D conversion for self-diagnosis and then starts A/D conversion on ANn channels selected by the ADANSA0 and ADANSA1 registers, starting from the channel with the smallest number n.
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 14-bit A/D converter 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.

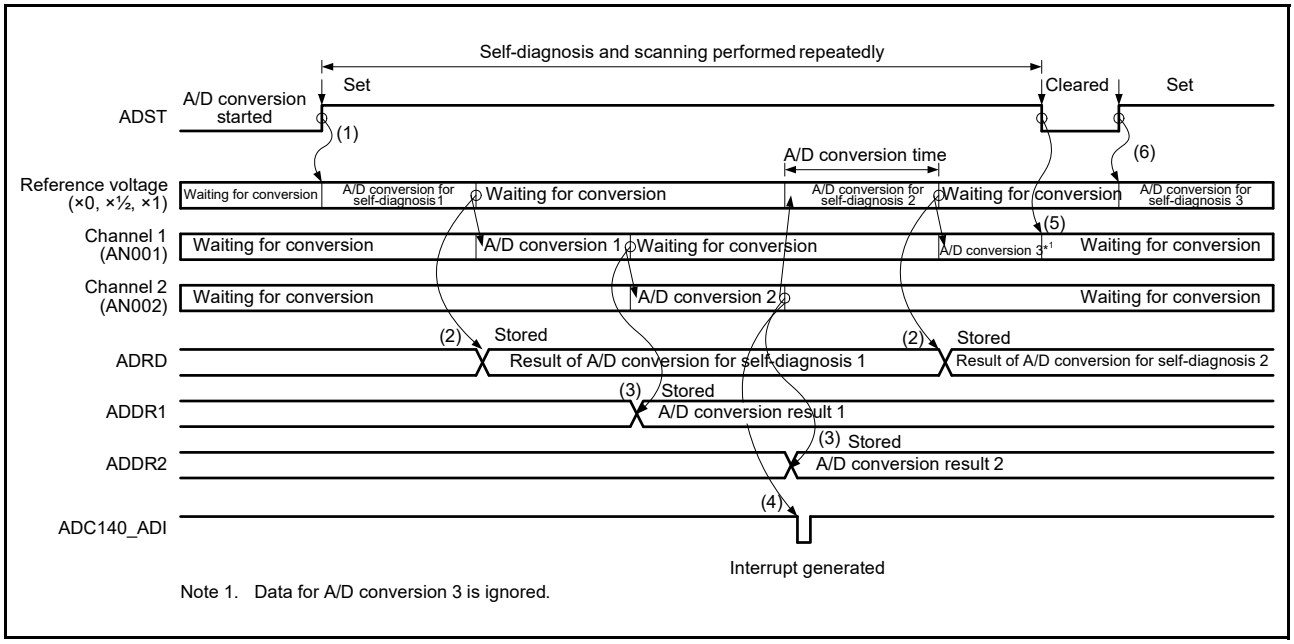


Figure 39.12 Example of operation in continuous-scan mode with basic operation AN001 and AN002 selected and with self-diagnosis

39.3.4 Group-Scan Mode

39.3.4.1 Basic Operation

In group-scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in group A and group B after scanning is started by a synchronous trigger (ELC) as below. The scan operation of each group is similar to the scan operation in single-scan mode.

The synchronous triggers of group A and B can be selected using the ADSTRGR.TRSA[5:0] bits and ADSTRGR.TRSB[5:0] bits, respectively. The different triggers should be used for group A and group B to prevent simultaneous A/D conversion of group A and group B. Software trigger should not be used.

The group A channels to be A/D-converted are selected using the ADANSA0 and ADANSA1 registers, while the group B channels to be A/D-converted are selected using the ADANSB0 and ADANSB1 registers. Group A and group B cannot use the same channels.

In group-scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) must both be set to 0 (deselected). When self-diagnosis is selected in group-scan mode, self-diagnosis is separately executed for group A and group B.

The following describes operation in group-scan mode using a synchronous trigger from the ELC. Specifically, the ELC_AD00 and ELC_AD01 triggers from ELC are assumed to be used to start conversion of group A and group B, respectively. Also, the ELC_AD00 and ELC_AD01 are selected for the GPT event by the associated ELC.ELSRn registers.

1. Scanning of group A is started by ELC_AD00.
2. When group A scanning is completed, an ADC140_ADI interrupt is generated (without register setting).
3. Scanning of group B is started by ELC_AD01.
4. When group B scanning is completed, an ADC140_GBADI interrupt is generated if the ADCSR.GBADIE bit is 1 (ADC140_GBADI interrupt upon scanning completion is enabled).

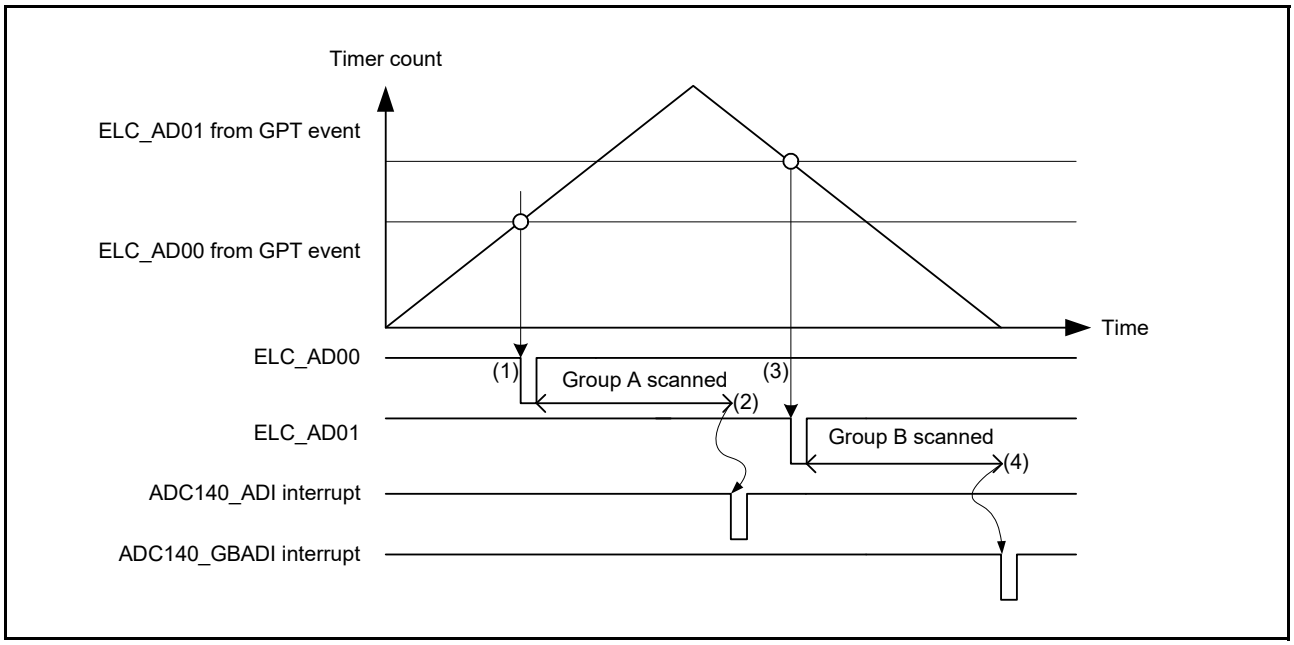


Figure 39.13 Example of operation in group-scan mode with basic operation synchronous triggers from ELC

39.3.4.2 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) are performed as a sequence for group A. For group B, single-scan operation started by a synchronous trigger (ELC) is performed once.

In group-scan mode, the synchronous triggers of group A and B can be selected using the TRSA[5:0] and TRSB[5:0] bits in ADSTRGR, respectively. The different triggers should be used for group A and group B to prevent simultaneous A/D conversion of group A and group B. Software trigger or asynchronous trigger (ADTRG0) should not be used.

When an ELC_AD00/ELC_AD01 is selected as the group A synchronous triggers by ADSTRGR.TRSA[5:0] bits (setting 0Bh to ADSTRGR.TRSA[5:0]), operation proceeds in extended double-trigger mode.

The group A and group B channels to be A/D-converted are selected using the DBLANS[4:0] bits in the ADCSR register and the ADANSB0 and ADANSB1 registers, respectively. The same channels cannot be selected for both groups.

In group-scan mode, the temperature sensor output A/D conversion select bit (ADEXICR.TSSA) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA) should both be set to 0 (deselected).

When double-trigger mode is selected in group scan mode, self-diagnosis cannot be selected.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the DBLANS[4:0] bits in ADCSR and setting the DBLE bit in ADCSR to 1.

The following steps describe operation in group-scan mode with double-trigger mode using a synchronous trigger from the ELC. Specifically, the ELC_AD00 and ELC_AD01 triggers from ELC are assumed to be used to start conversion of group A and group B, respectively. Also, the ELC_AD00 and ELC_AD01 are selected for the GPT event by the associated ELC.ELSRn registers.

1. Scanning of group B is started by the ELC_AD00 trigger from the ELC.
2. When group B scanning is completed, an ADC140_GBADI interrupt is output if the GBADIE bit in ADCSR is 1 (ADC140_GBADI interrupt upon scanning completion is enabled).
3. The first scanning of group A is started by the first ELC_AD01 trigger.
4. When the first scanning of group A is completed, the conversion result is stored into the corresponding A/D data register (ADDRy); an ADC140_ADI interrupt request is not generated.
5. The second scanning of group A is started by the second ELC_AD01 trigger.
6. When the second scanning of group A is completed, the conversion result is stored into ADDBLDR. An ADC140_ADI interrupt is generated (without register setting).

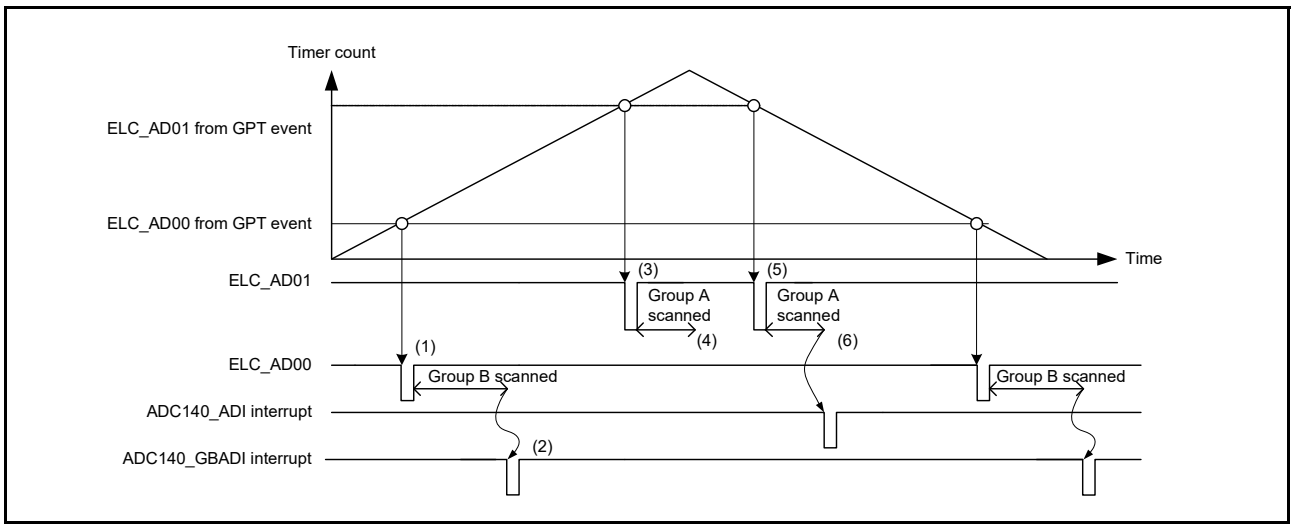


Figure 39.14 Example of operation in group-scan mode with double-trigger mode and basic operation synchronous triggers from ELC

39.3.4.3 Operation under Group A Priority Control

Setting the ADGSPCR.PGS bit to 1 in group-scan mode makes operation proceed under group A priority control. When setting the PGS bit in the ADGSPCR register to 1, follow the procedure described in Figure 39.15. If the procedure is not followed, A/D conversion operation and stored data are not guaranteed.

In basic group-scan mode, input of while A/D conversion is underway for one group in group A or group B, input of the trigger for A/D conversion for the other group is ignored. Under group A priority control, if a group A trigger is input during A/D conversion for group B, A/D conversion for group B is discontinued and A/D conversion for group A proceeds. If the setting of the ADGSPCR.GBRSCN bit is 0, the converter enters wait state on completion of the A/D conversion for group A. If the setting of the ADGSPCR.GBRSCN bit is 1, the converter automatically restarts scanning for group B from the head of the group after completion of the A/D conversion for group A. Table 39.9 summarizes operations in response to the input of a trigger during A/D conversion with the settings of the ADGSPCR.GBRSCN bit.

Scan operations in group A or group B are the same in single-scan mode. Additionally, single scanning continues to proceed if the ADGSPCR.GBRP bit is set to 1 during scanning operations for group B.

For the trigger settings in group-scan mode, select a synchronous trigger for group A using the ADSTRGR.TRSA[5:0] bits and select a synchronous trigger for group B different from that of group A using the ADSTRGR.TRSB[5:0] bits. Set the ADSTRGR.TRSB[5:0] bits to 3Fh when setting the ADGSPCR.GBRP bit to 1.

Additionally, as targets for A/D conversion, select channels for group A using the ADANSA0 and ADANSA1 registers, and for group B, select channels different from those for group A using the ADANSB0 and ADANSB1 registers.

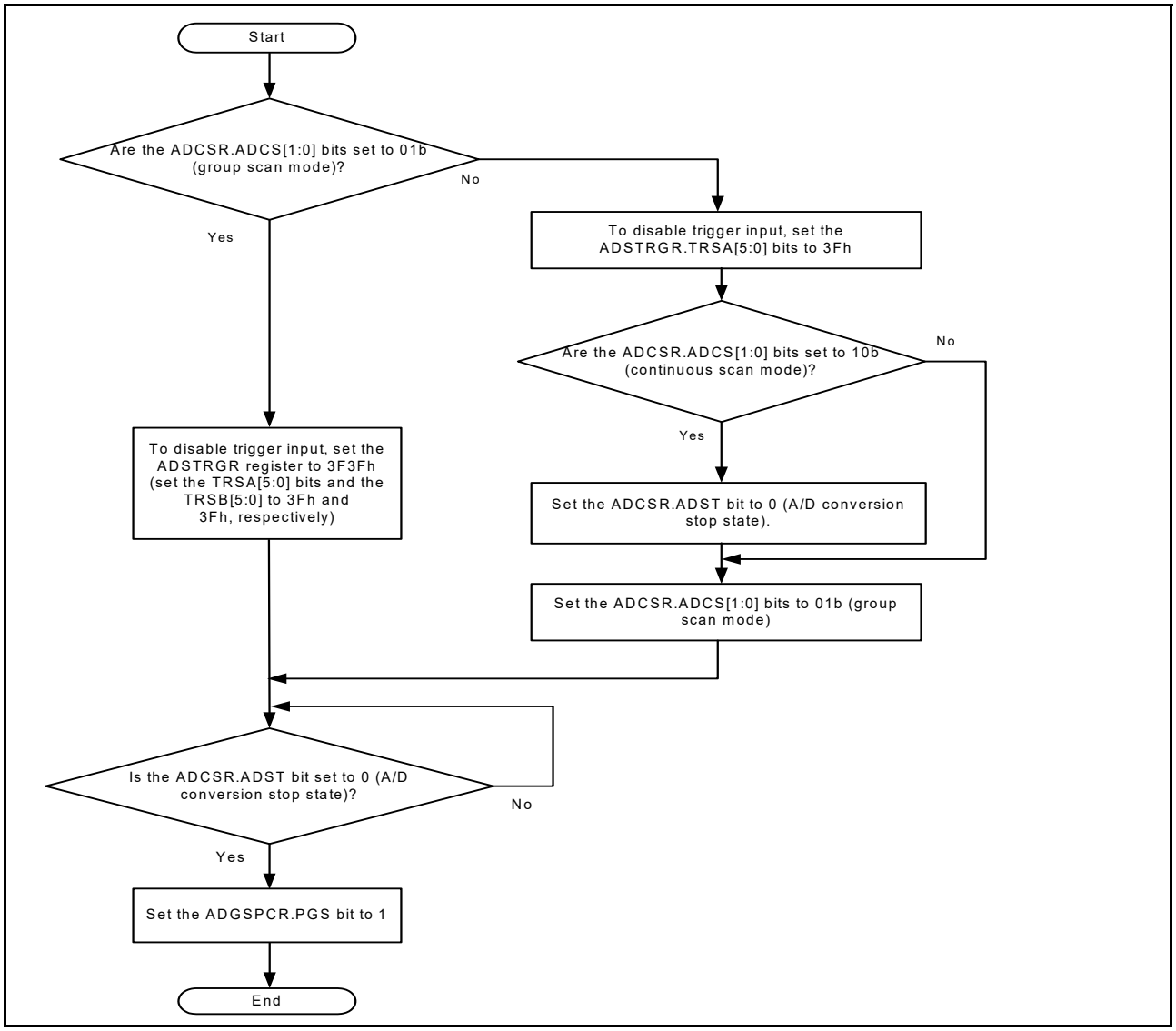


Figure 39.15 Flow of ADGSPCR.PGS bit setting

Table 39.9 Control of A/D conversion operations according to ADGSPCR.GBRSCN bit setting

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 is performed on group B after A/D conversion on group A is completed
When A/D conversion for group B is in progress	Input of trigger for group A	Conversion for group B that is in progress is discontinued and conversion for group A starts	• Conversion in progress for group B is discontinued and conversion for group A starts • Conversion for group B starts after conversion for group A is completed.
	Input of trigger for group B	Trigger input is ineffective	Trigger input is ineffective

The following describes operations in group-scan mode under group A priority control (for example, ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

1. When input of a trigger for group B sets the ADCSR.ADST bit to 1 (starting A/D conversion), conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the smallest number n.
2. On completion of A/D conversion, the result is stored in the corresponding A/D data register (ADDRy).
3. The ADCSR.ADST bit is cleared on the input of a trigger for group A while operation for A/D conversion in group B is in progress, and the latter is discontinued. After that, the ADCSR.ADST bit is set to 1 (starting A/D conversion), and conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the smallest number n.
4. On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
5. An ADC140_ADI interrupt request is generated without register setting.
6. After the ADST bit is automatically cleared, again, the bit is automatically set to 1 (starting A/D conversion) and conversion for the ANn channels of group B selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the smallest number n.
7. On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
8. An ADC140_GBADI interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (ADC140_GBADI interrupt upon group B scanning completion enabled).
9. The ADST bit retains the value 1 (starting A/D conversion) during A/D conversion and is automatically cleared on completion of conversion, after which the A/D converter enters wait state.

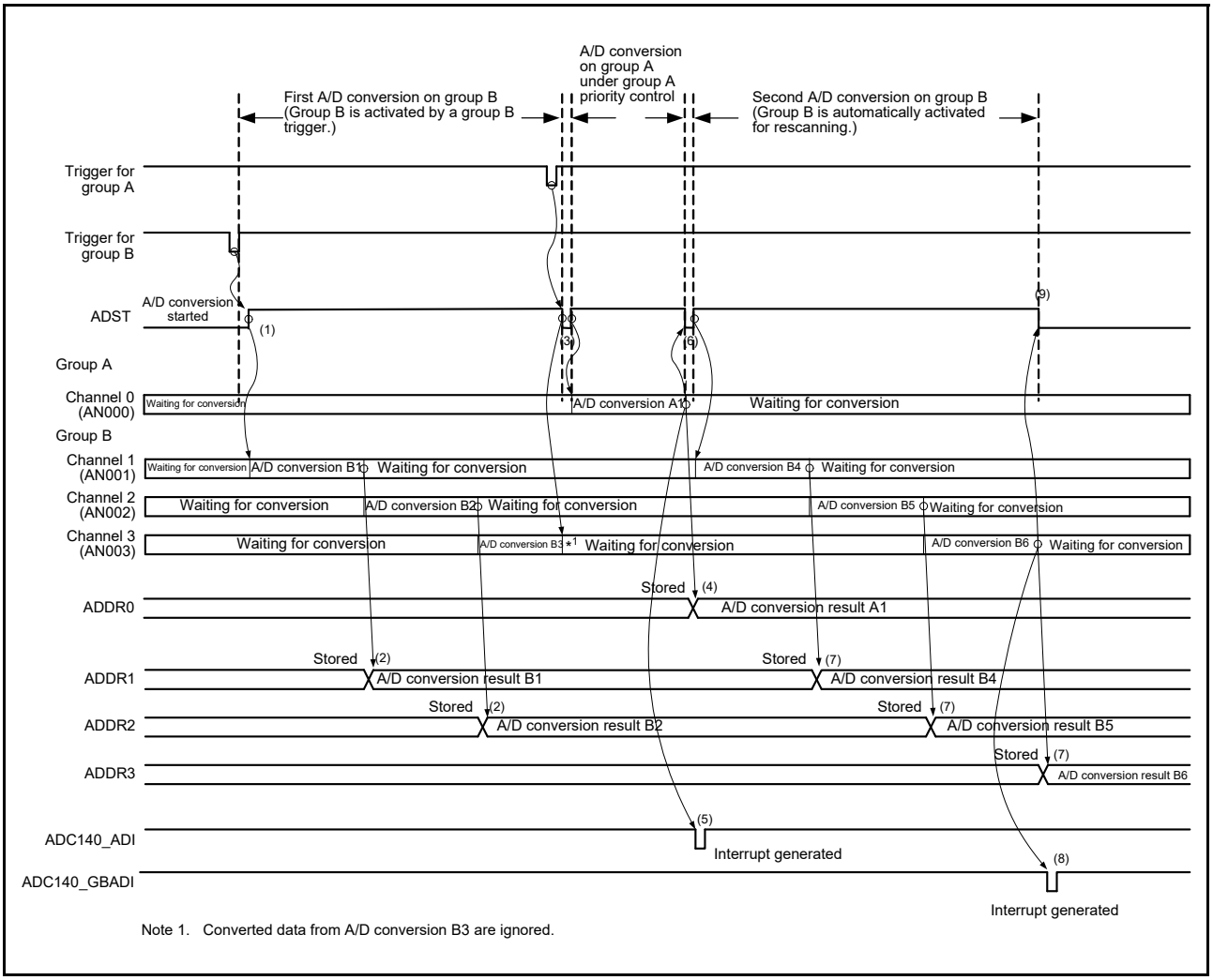
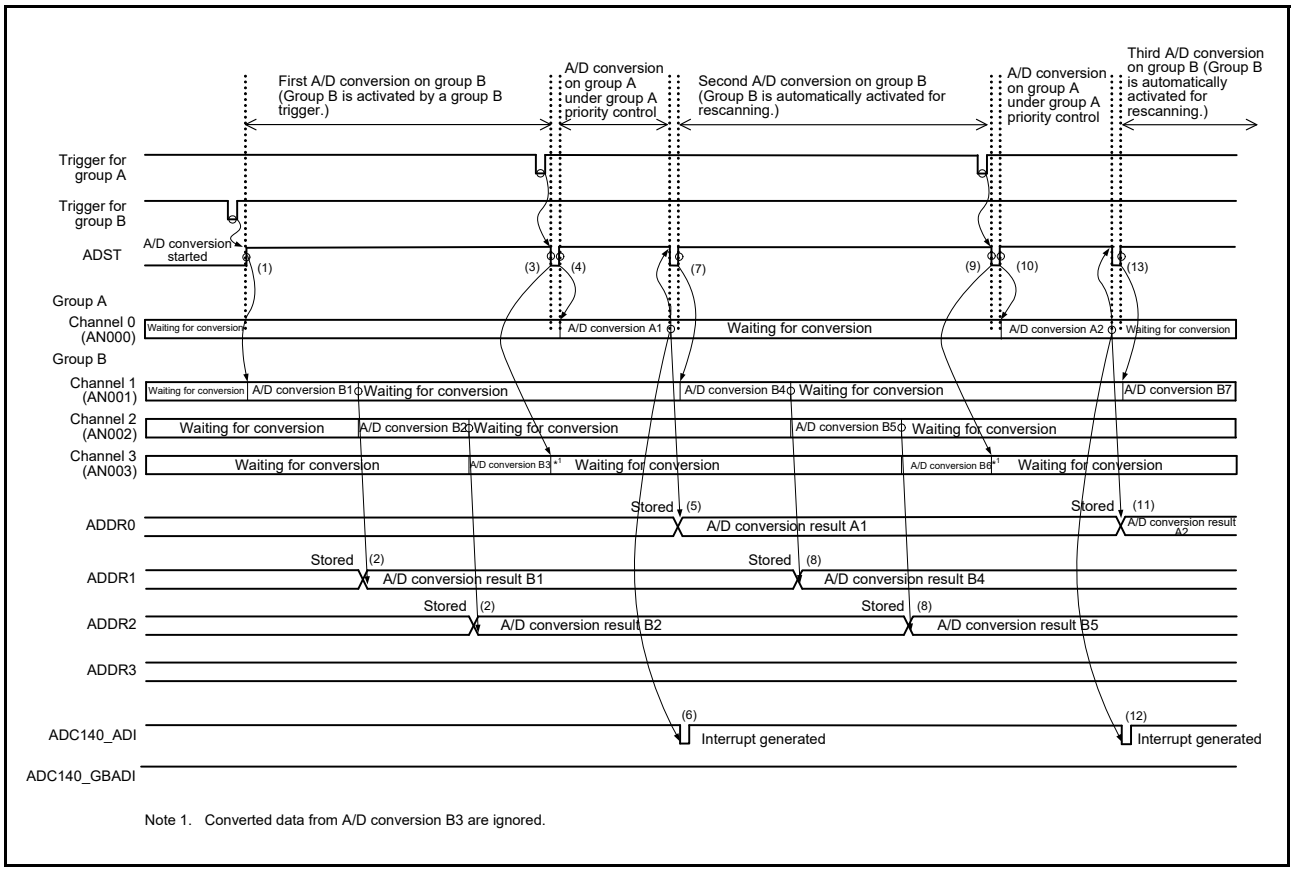


Figure 39.16 Example of operations under group A priority control (1) when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0

The following is an example when a group A trigger is input again during rescanning operation on group B. In this example, channel 0 is selected for group A and channels 1 to 3 are selected for group B when operation on group A is given priority (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0).

1. When a group B trigger input sets the ADCSR.ADST bit to 1 (starting A/D conversion), conversion for the ANn channels of group B selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the smallest number n.
2. On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
3. The ADCSR.ADST bit is set to 0 (stopping A/D conversion) on the input of a trigger for group A while operation for A/D conversion in group B is in progress, and the latter is discontinued.
4. After that, the ADCSR.ADST bit is set to 1 automatically and A/D conversion for the ANn group A channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the smallest number n.
5. On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
6. An ADC140_ADI interrupt request is generated (without register setting).
7. On completion of A/D conversion on the group A, rescanning operation on group B sets the ADCSR.ADST bit to 1 automatically if the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning operation). After that, A/D conversion for the ANn group B channels selected in the ADANSB0 and ADANSB1 registers starts again in order from the channel with the smallest number n.

8. On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
9. If a group A trigger is input during A/D conversion on group B for rescanning, the ADCSR.ADST bit is set to 0 (stopping A/D conversion) and the ongoing A/D conversion on group B is stopped.
10. After that, the ADCSR.ADST bit is set to 1 automatically and A/D conversion for the ANn group A channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the smallest number n.
11. On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
12. An ADC140_ADI interrupt request is generated (without register setting).
13. On completion of A/D conversion on group A, rescanning operation on group B sets the ADCSR.ADST bit to 1 automatically if the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning operation). After that, A/D conversion for the ANn group B channels selected in the ADANSB0 and ADANSB1 registers starts again in order from the channel with the smallest number n.
14. If a group A trigger is input during A/D conversion on group B for rescanning, steps 9 to 13 are repeated. If a group A trigger is not input, the ADCSR.ADST bit is cleared automatically on completion of A/D conversion on group B and the 14-bit ADC enters a wait state.



Note 1. Converted data from A/D conversion B3 are ignored.

Figure 39.17 Example of operations under group A priority control (2) when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0

The following is an example of a rescanning operation in which a group B trigger is input during A/D conversion on group A. In this example, channels 1 to 3 are selected for group A and channel 0 is selected for group B when operation on group A is given priority (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0).

1. When input of a trigger for group A sets the ADCSR.ADST bit to 1 (starting A/D conversion), conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the smallest number n.
2. On completion of A/D conversion on a single channel, the result is stored in the associated A/D data register (ADDRy).
3. If a group B trigger is input during A/D conversion on group A, A/D conversion on group B can be performed after the A/D conversion on group A is completed. However, if group A triggers are input continuously, the scan operation on group B is canceled by group A and is not performed.
4. On completion of the A/D conversion on the group A, an ADC140_ADI interrupt request is generated (without register setting).
5. On completion of the A/D conversion on the group A, activation of group B for rescanning sets the ADCSR.ADST bit to 1 automatically. Following that, conversion for the ANn channels of group B selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the smallest number n.
6. On completion of A/D conversion on a single channel, the result is stored in the associated A/D data register (ADDRy).
7. On completion of the rescanning operation on the group B, an ADC140_GBADI interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (ADC140_GBADI interrupt when scanning completion is enabled).
8. The ADST bit retains the value 1 (starting A/D conversion) during A/D conversion and is automatically cleared on completion of conversion, after which the A/D converter enters a wait state.

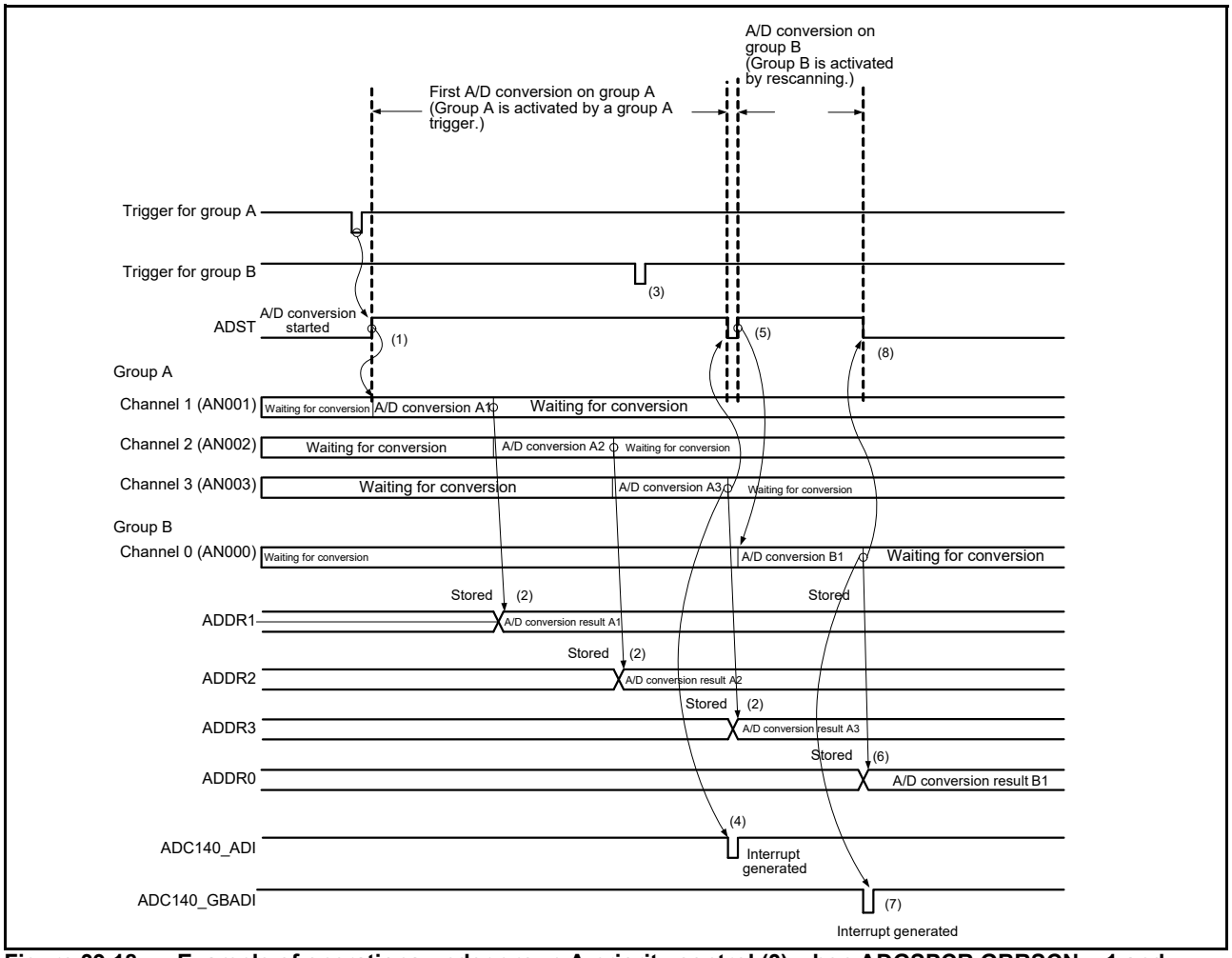


Figure 39.18 Example of operations under group A priority control (3) when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0

The following is an example of operation under group A priority control in which channel 0 is selected for group A and channels 1 to 3 are selected for group B (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0).

1. When input of a trigger for group B sets the ADCSR.ADST bit to 1 (starting A/D conversion), conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the smallest number n.
2. On completion of A/D conversion on a single channel, the result is stored in the associated A/D data register (ADDRy).
3. If a group A trigger is input during A/D conversion on group B, the ADCSR.ADST bit is set to 0 and the ongoing A/D conversion on group B is stopped. After that, the ADCSR.ADST bit is set to 1 (starting A/D conversion) and conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the smallest number n.
4. On completion of A/D conversion on a single channel, the result is stored in the associated A/D data register (ADDRy).
5. An ADC140_ADI interrupt request is generated without register setting.
6. The ADCSR.ADST bit retains the value 1 (starting A/D conversion) during A/D conversion and is cleared on completion of conversion, after which the A/D converter enters wait state.

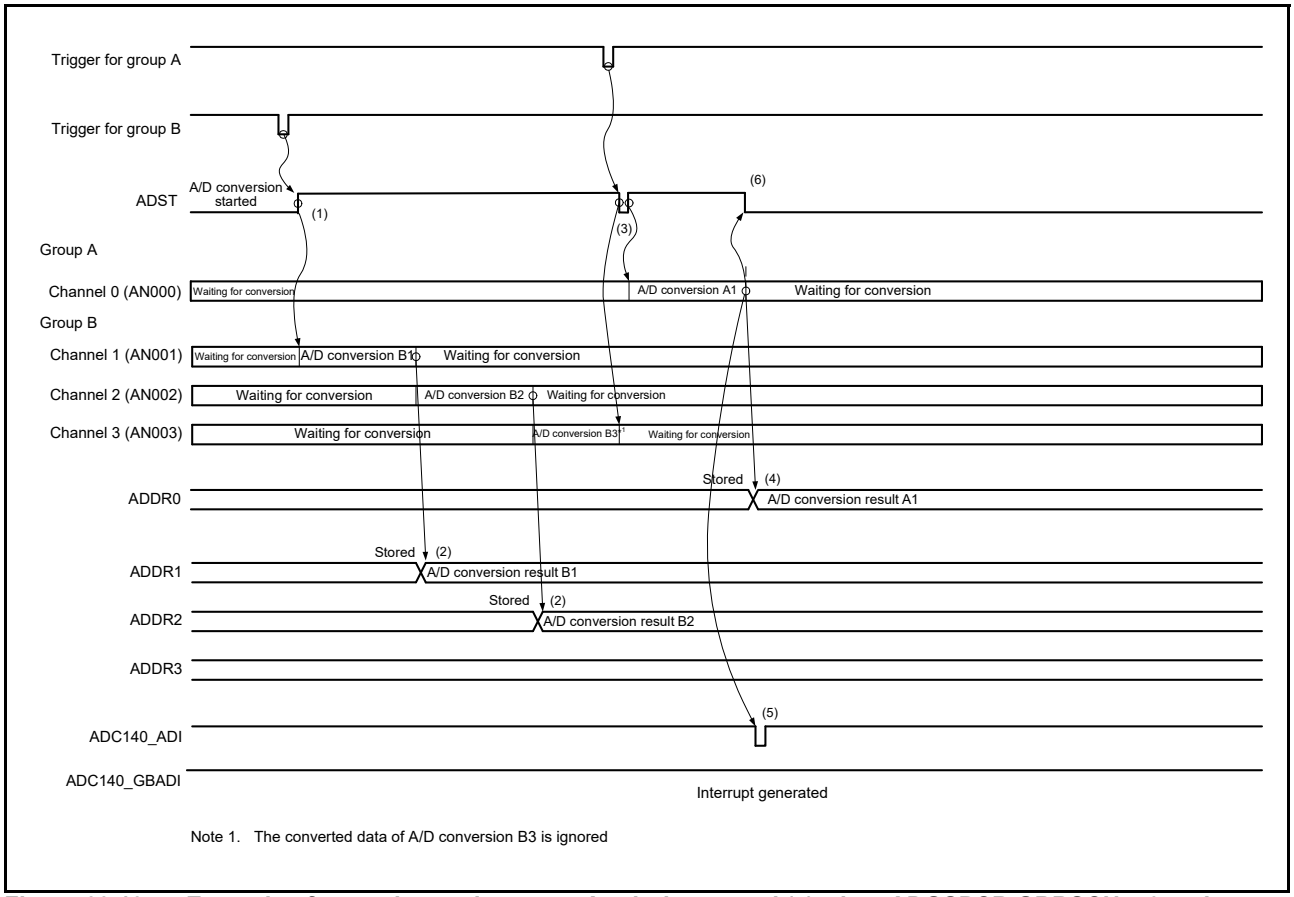
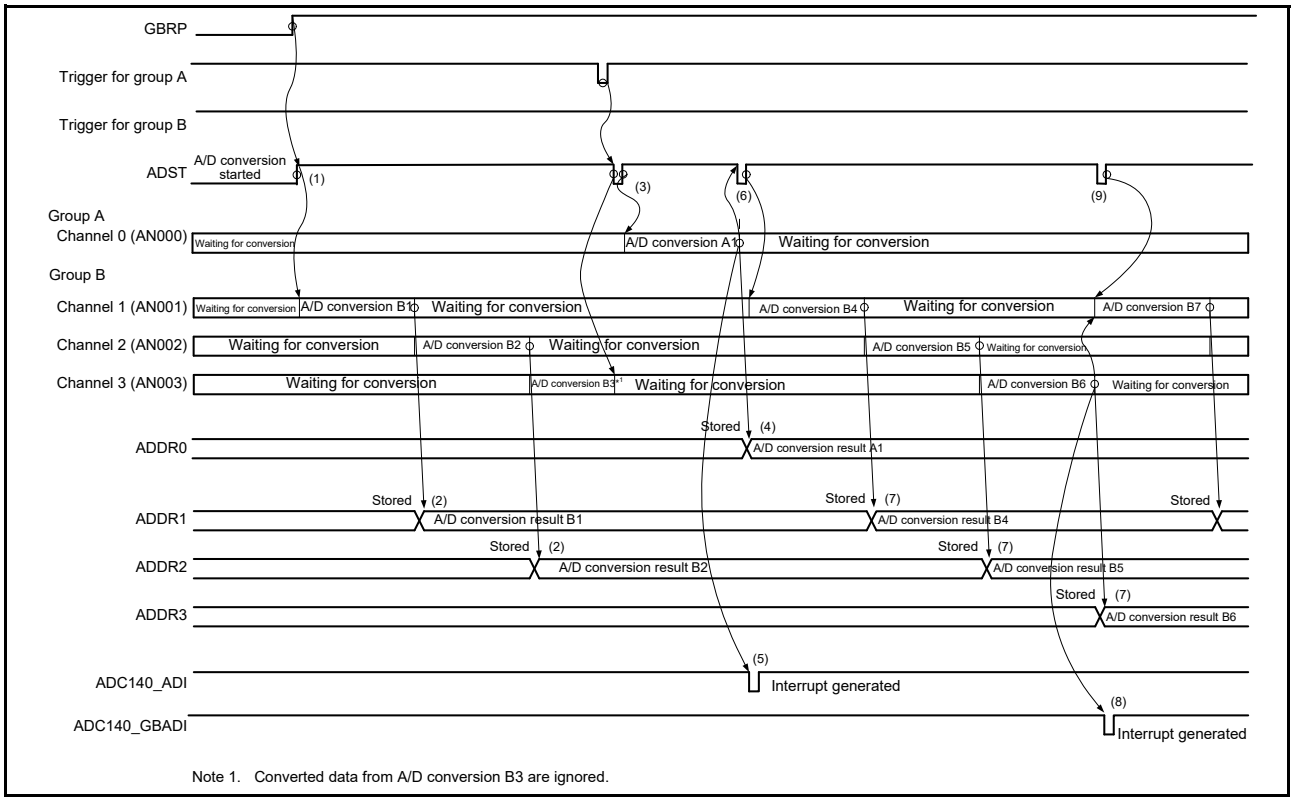


Figure 39.19 Example of operation under group A priority control (4) when ADGSPCR.GBRSCN = 0 and ADGSPCR.GBRP = 0

The following is an example of operation under group A priority control in which channel 0 is selected for group A and channels 1 to 3 are selected for group B (ADGSPCR.GBRP = 1).

1. The ADCSR.ADST bit is set to 1 (starting A/D conversion) when ADGSPCR.GBRP is set to 1, and conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the smallest number n.
2. On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
3. If a group A trigger is input during A/D conversion on group B, the ADCSR.ADST bit is set to 0 and the ongoing A/D conversion on group B is stopped. After that, the ADCSR.ADST bit is set to 1 (starting A/D conversion) and conversion for the ANn channels selected in the ADANSA0 and ADANSA1 registers starts in order from the channel with the smallest number n.
4. On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
5. An ADC140_ADI interrupt request is generated without register setting.
6. After the ADST bit is automatically cleared, again, the ADCSR.ADST bit is automatically set to 1 (starting A/D conversion) and conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the smallest number n.
7. On completion of A/D conversion on a single channel, the result is stored in the associated A/D data register (ADDRy).
8. An ADC140_GBADI interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1.
9. After the ADST bit is automatically cleared, the bit is automatically set to 1 (starting A/D conversion) and conversion for the ANn channels selected in the ADANSB0 and ADANSB1 registers starts in order from the channel with the smallest number n. Steps 6 to 9 are repeated as long as the ADGSPCR.GBRP bit remains 1. Clearing of the ADCSR.ADST bit to 0 is prohibited while the ADGSPCR.GBRP bit is set to 1. Follow the procedure for clearing the ADCSR.ADST bit operation by software, shown in Figure 39.31, to forcibly stop A/D conversion while ADGSPCR.GBRP = 1.



Note 1. Converted data from A/D conversion B3 are ignored.

Figure 39.20 Example of operation under group A priority control (5) when ADGSPCR.GBRP = 1

39.3.5 Compare Function (Window A, Window B)

39.3.5.1 Compare Function Window A/B

The compare function compares a reference value with the A/D conversion result. The reference value can be independently set for window A and window B. When the compare function is in use, the self-diagnosis function and double-trigger mode cannot be used. The significant differences between window A and window B include their different interrupt output signals and the limitation that window B can select only one channel.

The following sequence describes operations that combine of continuous-scan mode and the compare function:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the software, a synchronous trigger (ELC), or an asynchronous trigger, A/D conversion starts in the order 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 (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 set ADWINULB/ADWINLLB register value.
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) is set to 1. If the ADCMPCR.CMPAIE bit is 1, an ADC140_CMPAI interrupt request (level) is generated. In the same way, when window B meets the condition set in ADCMPBNSR.CMPLB, the Compare Window B Flag (ADCMPBSR.CMPSTB) is set to 1. If the ADCMPCR.CMPBIE bit is 1, an ADC140_CMPBI interrupt request is generated.
4. Upon completion of all selected A/D conversions and comparisons, scan restarts.
5. After the ADC140_CMPAI and ADC140_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, an ADC140_CMPAI interrupt request is canceled. In the same way, when all compare flags of window B are cleared, an ADC140_CMPBI interrupt request is reset. To perform comparison again, restart the A/D conversion.

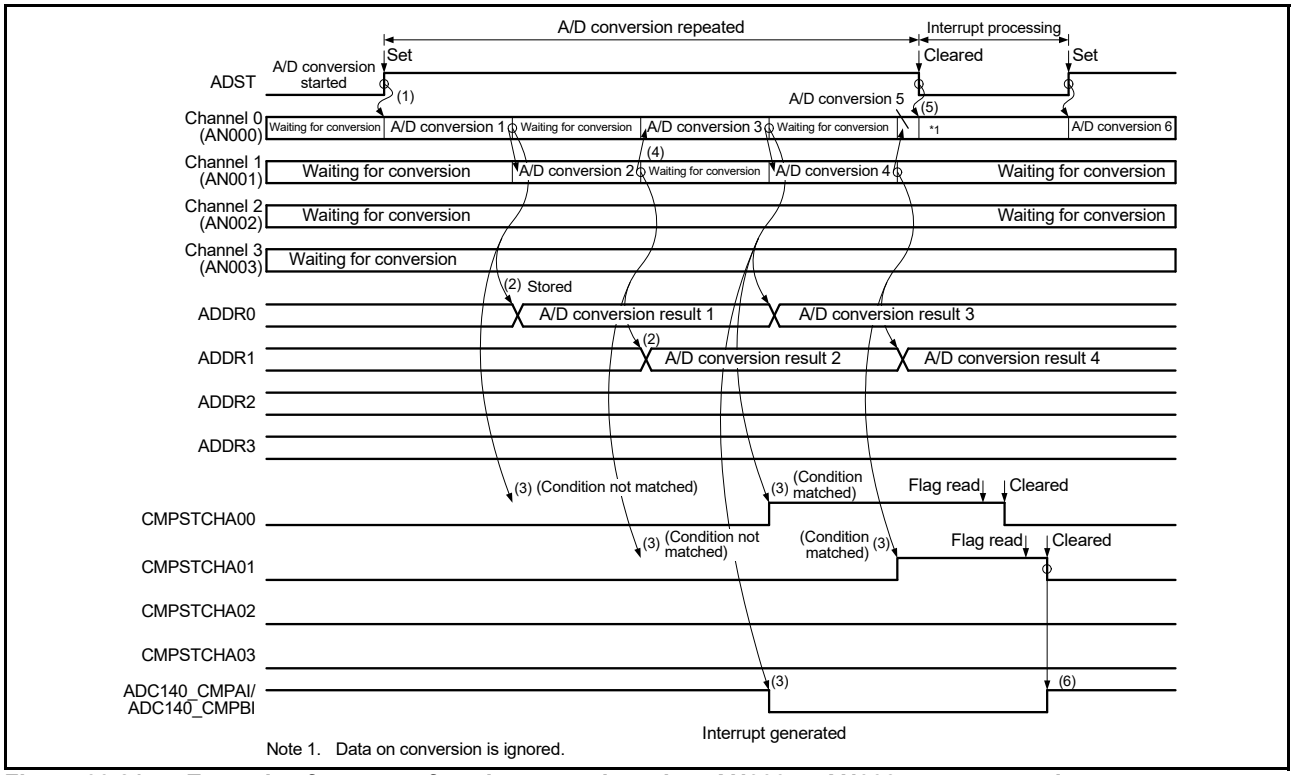


Figure 39.21 Example of compare function operation when AN000 to AN003 are compared

39.3.5.2 Event Output of Compare Function

The event output of compare function specifies the upper-side reference voltage value and the lower-side reference voltage value for window A and window B respectively, compares the A/D converted value of the selected channel with the upper/lower side reference voltage value, then outputs events (ADC140_WCMPM/ADC140_WCMPUM) according to 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 one channel in window A meets comparison condition, comparison result of window A becomes met. When using this function, perform A/D conversion in single-scan mode. Channels AN000 to AN027, internal reference voltage, and temperature sensor output are selectable for window A. However, when selecting the internal reference voltage or the temperature sensor output, it cannot be selected together with any other channel. Furthermore, 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 AN000 to AN027, internal reference voltage, and temperature sensor output is selectable for window B. However, when selecting the internal reference voltage or the temperature sensor output, it cannot be selected together with any other channel. Additionally, if the internal reference voltage is selected as the upper-potential reference voltage, the internal reference voltage or the temperature sensor output cannot be A/D converted.

The following sequence describes the setting procedure and example when using event output of compare function:

1. Confirm that the value of the ADCSR.ADCS[1:0] bits is 00b (single-scan mode).
2. Select using channel for window A by ADCMPANSR0/1 and ADCMPANSER. Set window comparison conditions in registers ADCMPLR0/1, ADCMPLER. Set the upper-side and lower-side reference values in registers ADCMPDR0/1.
3. Select using channel and comparison conditions for window B by ADCMPBNSR register, and set the upper/lower-side reference values in ADWINULB/ADWINLLB registers.
4. Set composite conditions for window A/B, window A/B operation enable, and interrupt output enable in ADCMPCR.

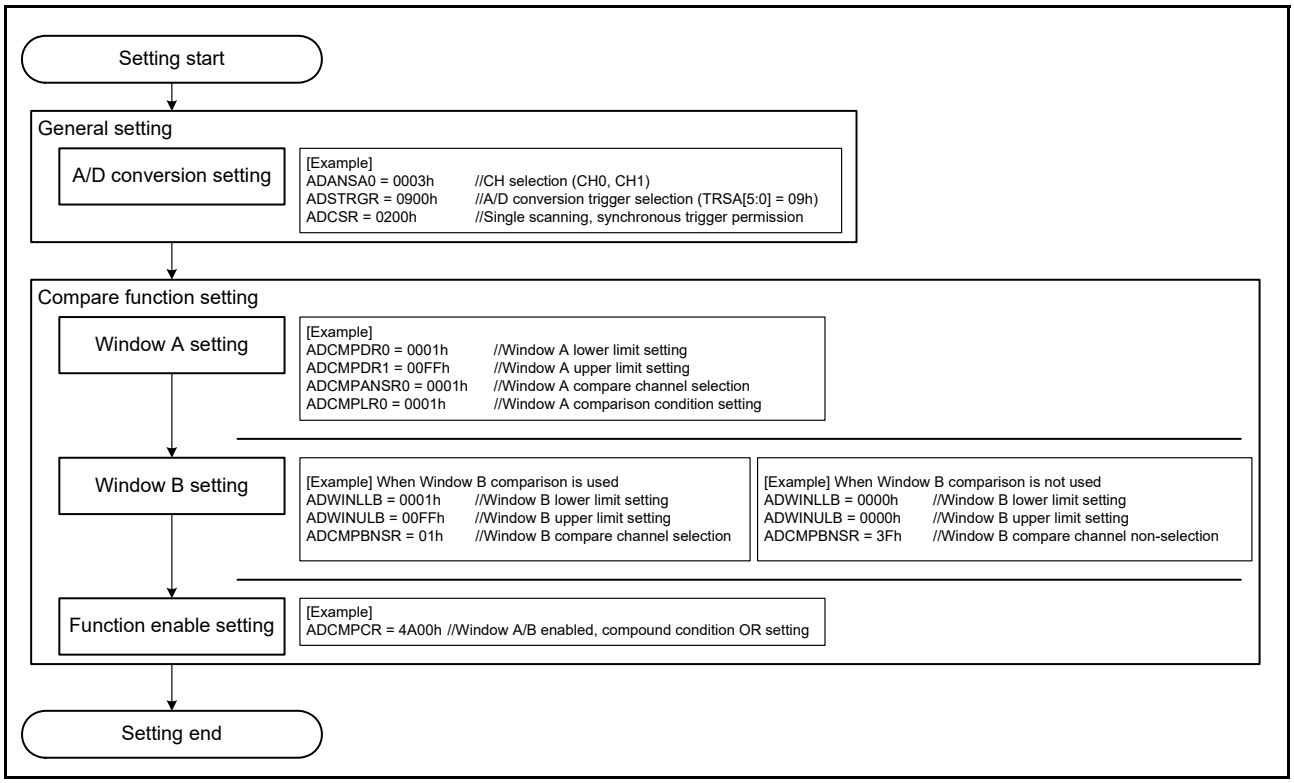


Figure 39.22 Setting example when using event output of the compare function

Notes on the event output usage when using only the window A for compare function:

- Set both window A and window B enable. (ADCMPCR.CMPAE = 1, ADCMPCR.CMPBE = 1).
- Set the compound condition of window A and B to "OR condition". (ADCMPCR.CMPAB[1:0] = 00b).
- Set the compared channel of window B to "Not select". (ADCMPBNSR.CMPCHB[5:0] = 111111b).
- Set the compare condition of window B to "0 < results < 0" means always mismatch. (ADCMPCR.WCMPE = 1, ADWINLLB = ADWINULB = 0000h, ADCMPBNSR.CMPLB = 1).

Figure 39.23 shows the event output operation example of compare function.

A scan end event (ADC140_ADI) is output at the same time as one-time single-scan completion. A match or mismatch event (ADC140_WCMPM/ADC140_WCMPUM) is output with 1 PCLKB clock delay depending on ADCMPCR.CMPAB[1:0] settings.

Note: The match and mismatch events are exclusive therefore both events are never output simultaneously.

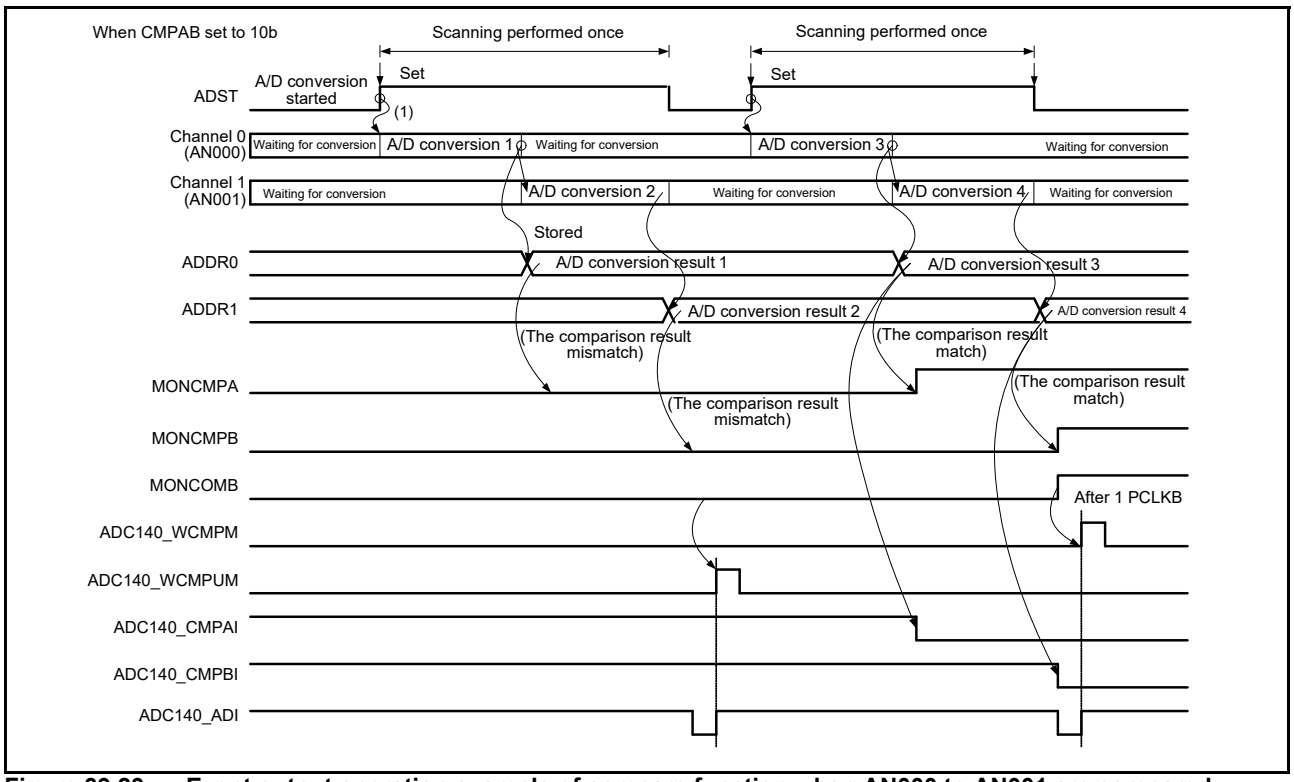


Figure 39.23 Event output operation example of compare function when AN000 to AN001 are compared

Note: Event output of compare function outputs match/mismatch from the comparison results of window A and window B, according to the ADCMPCR.CMPAB[1:0] settings.

Note: The comparison result of window A is the logical addition of the comparison results of comparison target channels of window A. The comparison results of window A and B are updated by each A/D conversion, and are kept even when single scan ends. To clear the comparison results to 0, set ADCMPCR.CMPAE and ADCMPCR.CMPBE to 0.

39.3.5.3 Restrictions for Compare Function

The following restrictions apply to the compare function:

1. The compare function must not be used with the self-diagnosis function or double-trigger mode. (The compare function is not available for ADRD, ADDBLDR, ADDBLDRA, and ADDBLDRB).
2. Specify single-scan mode when using match/mismatch event outputs.
3. When temperature sensor or internal reference voltage is selected for window A, window B operations are disabled.
4. When temperature sensor or internal reference voltage is selected for window B, window A operations are disabled.
5. It is prohibited to set the same channel for window A and window B.
6. Set the reference voltage values so that the high-potential reference voltage value is equal or larger than the low-potential reference voltage value.

39.3.6 Analog Input Sampling and Scan Conversion Time

Scan conversion can be activated either by a software trigger, a synchronous trigger (ELC), or an asynchronous trigger (ADTRG0). After the start-of-scanning-delay time (t_D) elapses, processing for disconnection detection assistance, and processing of conversion for self-diagnosis all proceed, followed by processing for A/D conversion.

Figure 39.24 shows the scan conversion timing, in which scan conversion is activated by a software trigger or a synchronous trigger (ELC). Figure 39.25 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), disconnection detection assistance processing time (t_{DIS})*1, self-diagnosis A/D conversion processing time (t_{DIAG} and t_{DSD})*2, A/D conversion processing time (t_{CONV}), 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 ADSSTR register.

The time for conversion by successive approximation (t_{SAM}) is 37.5 ADCLK states with 14-bit accuracy and high-speed mode selected, 46.5 ADCLK states with 14-bit accuracy and low-current mode selected, 31.5 ADCLK states with 12-bit accuracy and high-speed mode selected, and 40.5 ADCLK states with 12-bit accuracy and low-current mode selected. Table 39.10 shows the scan conversion time.

The scan conversion time (t_{SCAN}) in single-scan mode for which the number of selected channels is n can be determined as follows:

$t_{SCAN} = t_D + (t_{DIS} \times n) + t_{DIAG} + t_{DSD} + (t_{CONV}{}^{*3} \times n) + t_{ED}$

The scan conversion time for the 1st cycle in continuous-scan mode is t_{SCAN} for single scan minus t_{ED}. The scan conversion time for the second and subsequent cycles in continuous-scan mode is fixed at $(t_{DIS} \times n) + t_{DIAG} + t_{DSD} + (t_{CONV}{}^{*3} \times n)$.

Note 1. 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 2. When the self-diagnosis function is not used, $t_{DIAG} = 0$, $t_{DSD} = 0$.

Note 3. When input sampling time (tSPL) of all selected channels are the same, this element equals tCONV × n. If each channel has a different sampling time, this element equals sum of tSPL and tSAM which are set to each selected channel respectively.

Table 39.10 Times for conversion during scanning (in numbers of cycles of ADCLK and PCLKB) (1 of 2)

Parameter			Symbol	Type/Conditions			Unit
				Synchronous trigger*5	Asynchronous trigger	Software trigger	
Scan start processing 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 due to an A/D conversion source of group A.)	t_D	3 PCLKB + 6 ADCLK, 5 PCLKB + 3 ADCLK*6	—	—	Cycle
		Group B is not to be stopped. (Activation by an A/D conversion source of 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 + 6 ADCLK	4 PCLKB + 6 ADCLK	6 ADCLK	
	Other than above			2 PCLKB + 4 ADCLK	2 PCLKB + 4 ADCLK	4 ADCLK	
Disconnection detection assistance processing time			t_{DIS}	The setting of ADNDIS[3:0] (initial value = 00h) × ADCLK*3			

Table 39.10 Times for conversion during scanning (in numbers of cycles of ADCLK and PCLKB) (2 of 2)

Parameter			Symbol		Type/Conditions: Synchronous trigger*5	Type/Conditions: Asynchronous trigger	Type/Conditions: Software trigger	Unit
Self-diagnosis conversion processing time*1	Sampling time		t_{DIAG}	t_{SPL}	The setting of ADSSTR00 (initial value = 0Dh) × ADCLK*4 + 0.5 ADCLK*4			
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	31.5 ADCLK at High-speed mode 40.5 ADCLK at Low-current mode			
		14-bit conversion accuracy			37.5 ADCLK at High-speed mode 46.5 ADCLK at Low-current mode			
	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		t_{CONV}	t_{SPL}	The setting of ADSSTRn (n = 0 to 15, L, T, O) (initial value = 0Dh) × ADCLK + 0.5 ADCLK			
	Time for conversion by successive approximation	12-bit conversion accuracy		t_{SAM}	31.5 ADCLK at High-speed mode 40.5 ADCLK at Low-current mode			
		14-bit conversion			37.5 ADCLK at High-speed mode 46.5 ADCLK at Low-current mode			
Scan end processing time*1			t_{ED}		1 PCLKB + 3 ADCLK, 2 PCLKB + 3 ADCLK*6			

Note 1. See Figure 39.24 and Figure 39.25 for illustration of times t_D, 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 0Fh (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.
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).

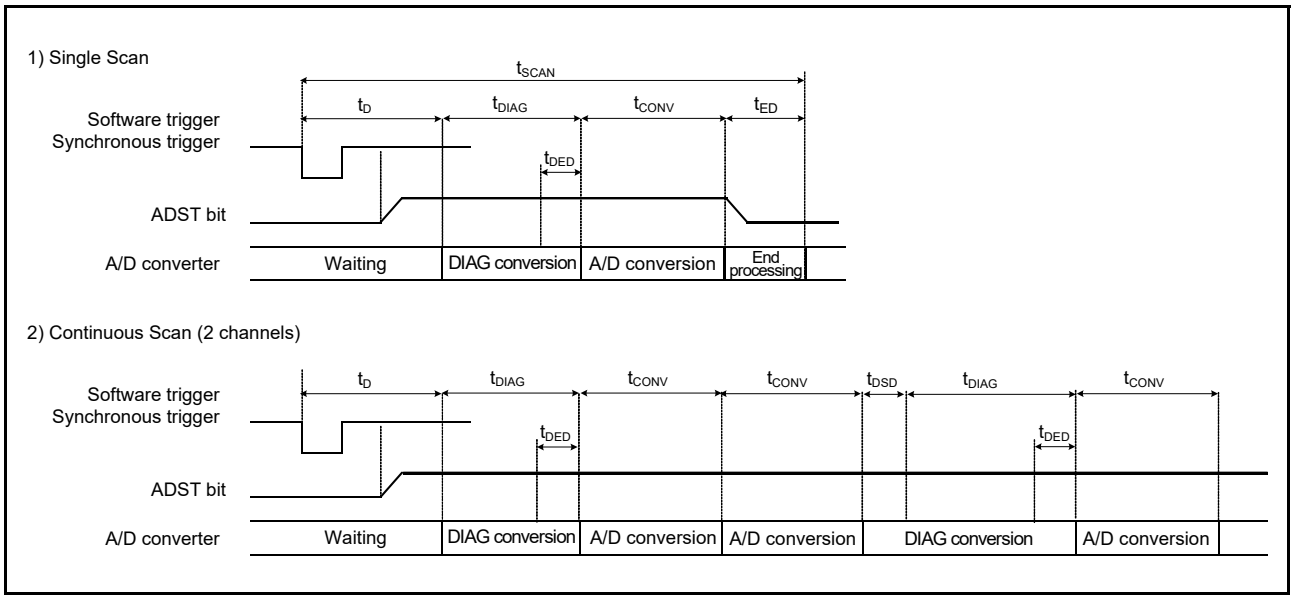


Figure 39.24 Scan conversion timing activated by software or synchronous trigger ELC input

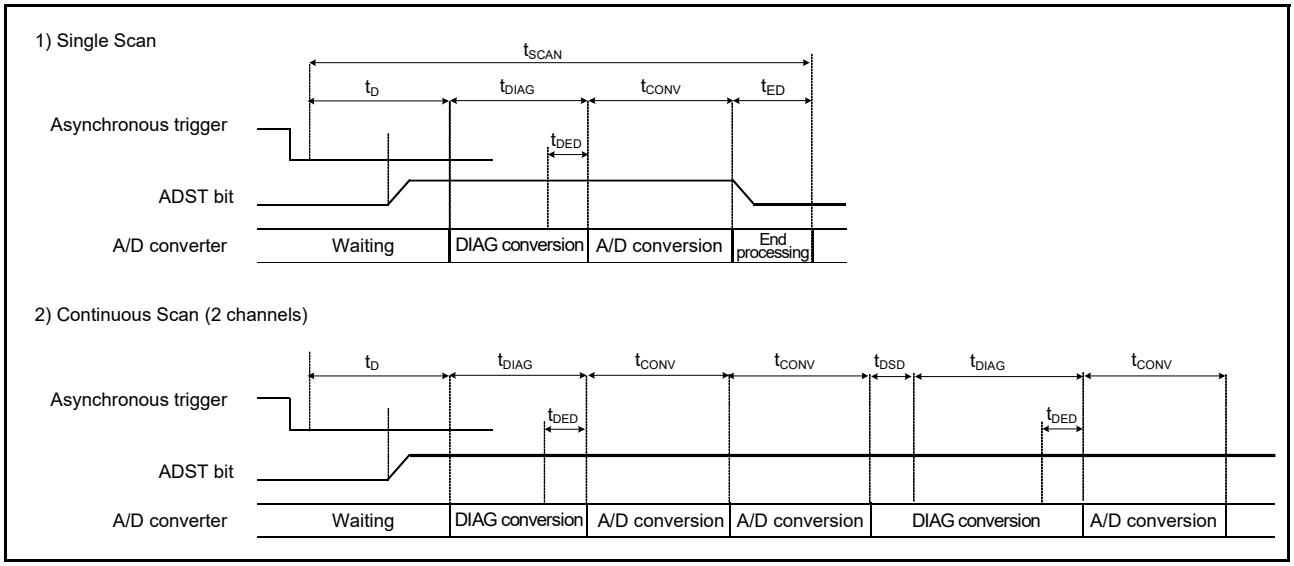


Figure 39.25 Scan conversion timing activated by asynchronous trigger input (ADTRG0)

39.3.7 Usage Example of A/D Data Register Automatic Clearing Function

A/D-converted value addition/average mode can be used when A/D conversion of the analog input of the selected channels, A/D conversion of the temperature sensor output, or A/D conversion of the internal reference voltage is selected.

Setting the ACE bit in ADCER to 1 automatically clears the A/D data registers (ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR) to 0000h when the A/D data registers are read by the CPU, DTC, or DMAC.

This function enables detection of update failures of the A/D data registers (ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR). The following describes examples in which the function to automatically clear the ADDRy register is enabled and disabled.

In a case where the ACE bit in ADCER is 0 (automatic clearing is disabled), if the A/D conversion result (0222h) is not written to the ADDRy register for some reason, the ADDRy value will remain the old data (0111h). In addition, if this ADDRy value is read into a general-purpose register using an A/D scan end interrupt, the old data (0111h) 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 a general-purpose register.

In a case where the ACE bit in ADCER is 1 (automatic clearing is enabled), when ADDRy = 0111h is read by the CPU, DTC, or DMAC, ADDRy is automatically set to 0000h. After that, if the A/D conversion result 0222h cannot be transferred to ADDRy for some reason, the cleared data (0000h) remains as the ADDRy value. If this ADDRy value is read into a general-purpose register using an A/D scan end interrupt at this point, 0000h is saved in the general-purpose register. Occurrence of an ADDRy update failure can be determined by simply checking that the read data value is 0000h.

39.3.8 A/D-Converted Value Addition/Average Mode

In A/D-converted value addition mode, the same channel is A/D-converted 1, 2, 3, 4, or 16*1 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 mode can be specified when A/D conversion of the channel select analog input, temperature sensor output, or internal reference voltage is selected.

Note 1. The addition count can be set to 16 only when 12-bit accuracy is selected.

39.3.9 Disconnection Detection Assist Function

This converter incorporates the disconnection detection assist 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 39.26 shows the A/D conversion operation when the disconnection detection assist function is used. Figure 39.27 shows an example of disconnection detection when precharge is selected. Figure 39.28 shows an example of disconnection detection when discharge is selected.

If any of the following functions are used, the disconnection detection assist function should be disabled:

- The temperature sensor
- The internal reference voltage
- A/D self-diagnosis.

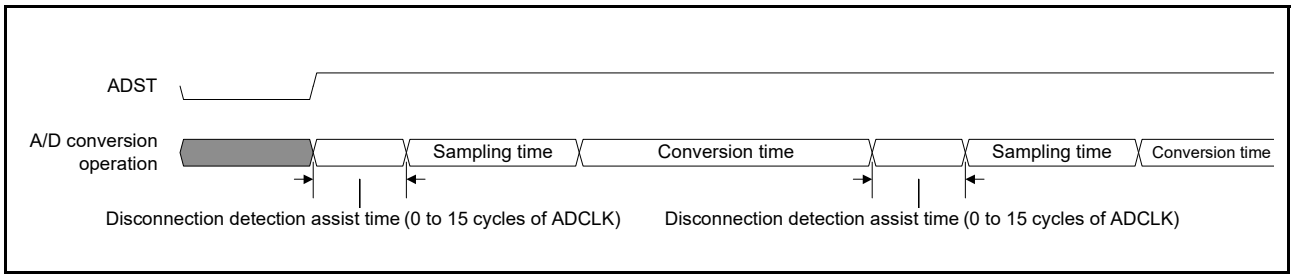
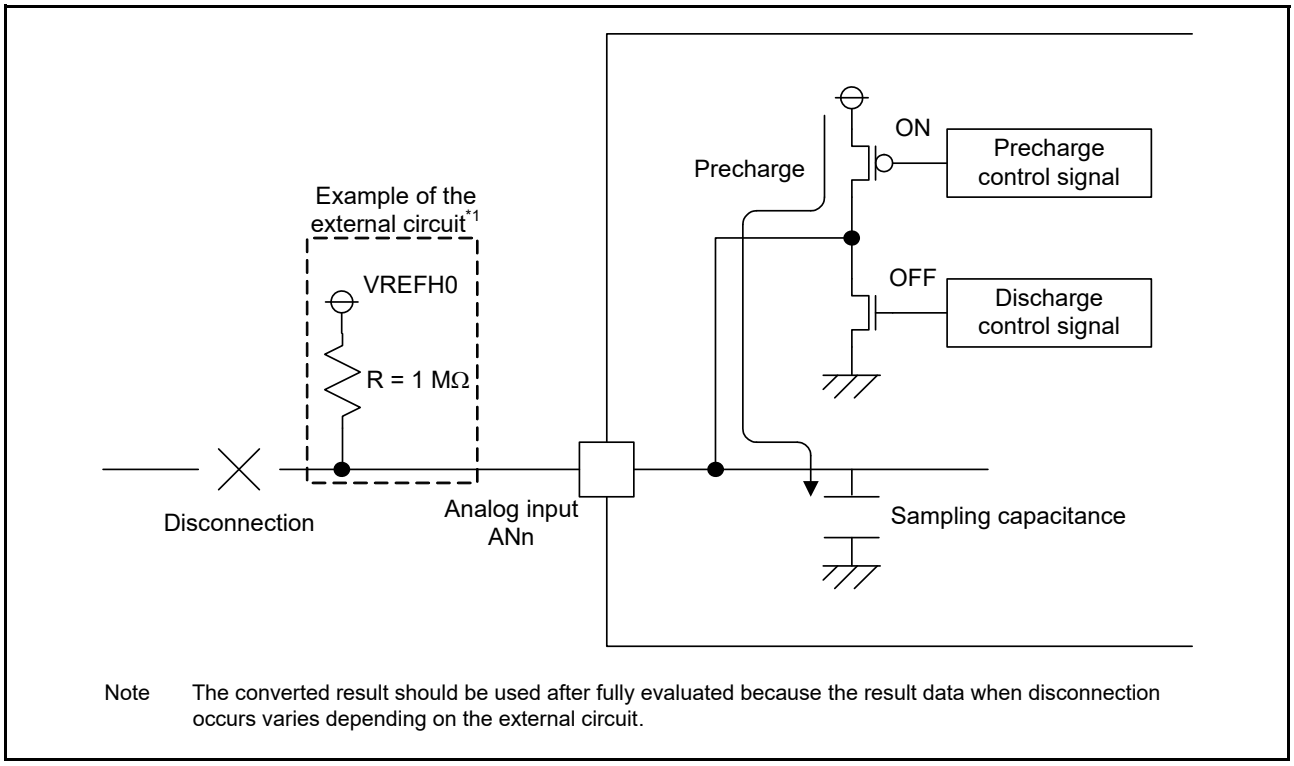
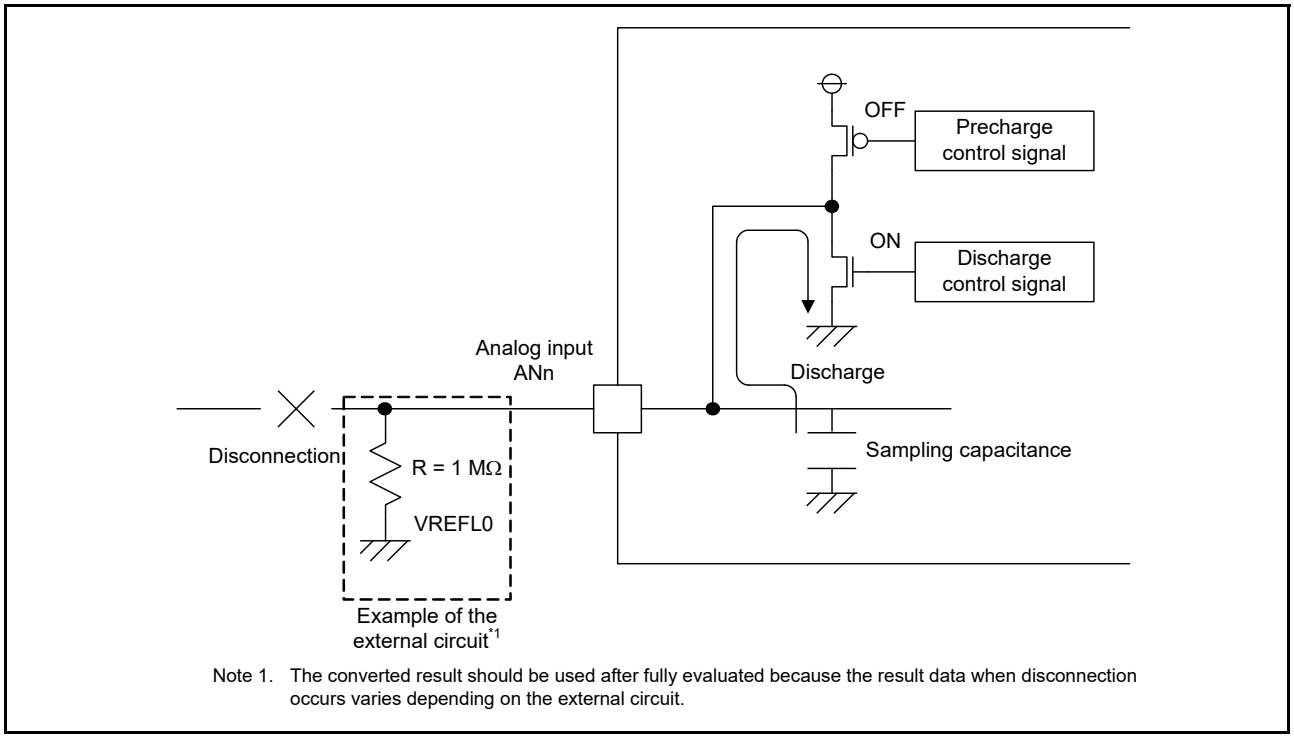


Figure 39.26 Operation of A/D conversion when disconnection detection assist function is used



Note The converted result should be used after fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 39.27 Example of disconnection detection when precharge is selected



Note 1. The converted result should be used after fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 39.28 Example of disconnection detection when discharge is selected

39.3.10 Starting A/D Conversion with Asynchronous Trigger

The A/D conversion can be started by the input of an asynchronous trigger. To start the A/D converter by an asynchronous trigger, first the pin function must be set by the PmnPFS register, the A/D conversion start trigger select bits (ADSTRGR.TRSA[5:0]) must be set to 000000b, then a high-level signal must be input to the asynchronous trigger (ADTRG0 pin). Both the ADCSR.TRGE and ADCSR.EXTRG bits must then be set to 1. Figure 39.29 shows timing of the asynchronous trigger input.

An asynchronous trigger cannot be selected by the A/D conversion start trigger select bits (ADSTRGR.TRSB[5:0]) for group B used in group scan mode. For details on setting the pin function, see section 20, I/O Ports.

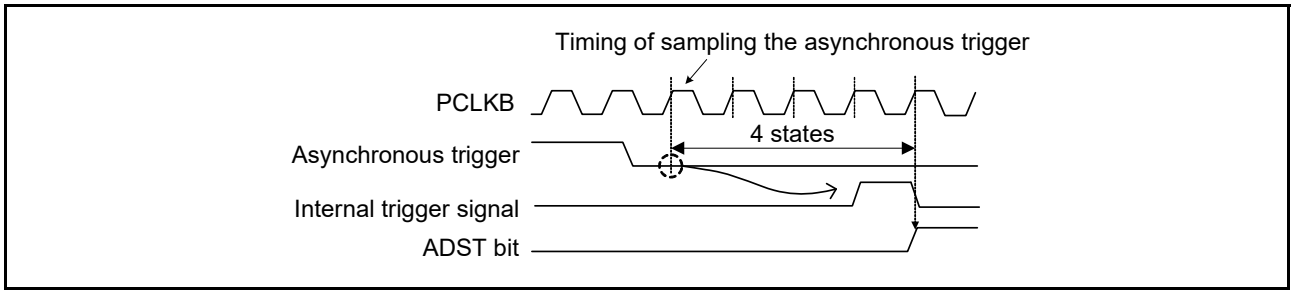


Figure 39.29 Asynchronous trigger input timing

39.3.11 Starting A/D Conversion with Synchronous Trigger from Peripheral Module

The A/D conversion can be started by a synchronous trigger (ELC). To start the A/D conversion by a synchronous trigger:

1. Set the ADCSR.TRGE bit to 1.
2. Clear the ADCSR.EXTRG bit to 0.
3. Select the relevant sources with the ADSTRGR.TRSA[5:0] and ADSTRGR.TRSB[5:0] bits.

39.4 Interrupt Sources and DTC/DMAC Transfer Requests

39.4.1 Interrupt Requests

The 14-bit ADC can send scan end interrupt requests ADC140_ADI and ADC140_GBADI to the CPU. The module also generates the ADC140_CMPAI/ADC140_CMPBI interrupt for the CPU in response to matches with a condition for comparison.

An ADC140_ADI interrupt is always generated. An ADC140_GBADI interrupt can be generated by setting the ADCSR.GBADIE bit to 1, similarly an ADC140_CMPAI and ADC140_CMPBI interrupts can be generated by setting the ADCMPCR.CMPAIE and ADCMPCR.CMPBIE bit to 1.

In addition, the DTC or DMAC can be started when an ADC140_ADI or an ADC140_GBADI interrupt is generated. Using an ADC140_ADI or an ADC140_GBADI interrupt to allow the DTC or DMAC to read the converted data enables continuous conversion without a burden on software.

Table 39.11 ADC14 interrupt source and ELC event

Operation			Interrupt request or ELC event	Interrupt request	DTC/DMAC activation	ELC event request	Function
Scan mode	Double-trigger mode	Compare function window A/B					
Single-scan mode	deselect	deselect	ADC140_ADI	✓	✓	✓	ADC140_ADI is generated at the end of single scan
		select	ADC140_ADI	✓	✓	✓	ADC140_ADI is generated at the end of single scan
			ADC140_CMPAI	✓	×	×	ADC140_CMPAI is generated in the match comparison condition of window A
			ADC140_CMPBI	✓	×	×	ADC140_CMPBI is generated in the match comparison condition of window B
			ADC140_WCMPM	×	✓	✓	ADC140_WCMPM is generated in the match conditions of the window A/B compare function
			ADC140_WCMPUM	×	✓	✓	ADC140_WCMPUM is generated in the mismatch conditions of the window A/B compare function
	select	deselect	ADC140_ADI	✓	✓	✓	ADC140_ADI is generated at the end of scans in even-numbered multiples
Continuous-scan mode	deselect	deselect	ADC140_ADI	✓	✓	✓	ADC140_ADI is generated at the end of all the selected channels scan
		select	ADC140_CMPAI	✓	×	×	ADC140_CMPAI is generated in the match comparison condition of window A
			ADC140_CMPBI	✓	×	×	ADC140_CMPBI is generated in the match comparison condition of window B
Group-scan mode	deselect	deselect	ADC140_ADI	✓	✓	✓	ADC140_ADI is generated at the end of group A scan
			ADC140_GBADI	✓	✓	×	ADC140_GBADI dedicated to group B is generated at the end of group B scan
		select	ADC140_ADI	✓	✓	✓	ADC140_ADI is generated at the end of group A scan
			ADC140_GBADI	✓	✓	×	ADC140_GBADI dedicated to group B is generated at the end of group B scan
			ADC140_CMPAI	✓	×	×	ADC140_CMPAI is generated in the match comparison condition of window A
			ADC140_CMPBI	✓	×	×	ADC140_CMPBI is generated in the match comparison condition of window B
	select	deselect	ADC140_ADI	✓	✓	✓	ADC140_ADI is generated at the end of Group A scans in even-numbered multiples
			ADC140_GBADI	✓	✓	×	ADC140_GBADI dedicated to group B is generated at the end of Group B scan

For details on DTC settings, see section 18, Data Transfer Controller (DTC), and for details on DMAC settings, see section 17, DMA Controller (DMAC).

39.5 Event Link Function

39.5.1 Event output to the ELC

The ELC uses the ADC140_ADI interrupt request signal as an event signal, enabling link operation for the preset module. The ADC140_GBADI interrupt and ADC140_CMPAI/ADC140_CMPBI interrupts cannot be used as an event signal. For details, see Table 39.11 Interrupt and ELC event function.

39.5.2 14-bit ADC operation by an event from the ELC

The 14-bit ADC can start A/D conversion by the preset event signal (ELC_AD00 and ELC_AD01) depending on the ELSRn setting of the ELC.

ELC_AD00 is the signal selected by ELC.ELSR8 register.

ELC_AD01 is the signal selected by ELC.ELSR9 register.

If an event ELC_AD00/ELC_AD01 occurs during A/D conversion, the event is disabled.

39.6 Selecting Reference Voltage

The A/D converter can select VREFH0 or AVCC0 as the high-potential reference voltage, and can select VREFL0 or AVSS0 as the internal reference voltage and the low-potential reference voltage. Set these before starting A/D conversion. For details of this setting, see the section on the ADHVREFCNT register.

39.7 A/D Conversion Procedure when Selecting Internal Reference Voltage as High-Potential Reference Voltage

This section 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 AN000 to AN027, 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 ADC.
2. Wait in the software for a 1 µs discharge period.
3. Set ADHVREFCNT.HVSEL[1:0] to 10b to select internal reference voltage as the high-potential reference voltage.

Note: The A/D converter 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 set forcibly. Select the internal reference voltage again 1 µs later.

4. Wait in the software until the internal reference voltage is stabilized (for 5 µs), and then perform A/D conversion.

Figure 39.30 shows a waveform chart for the procedure to select internal reference voltage as the high-potential reference voltage.

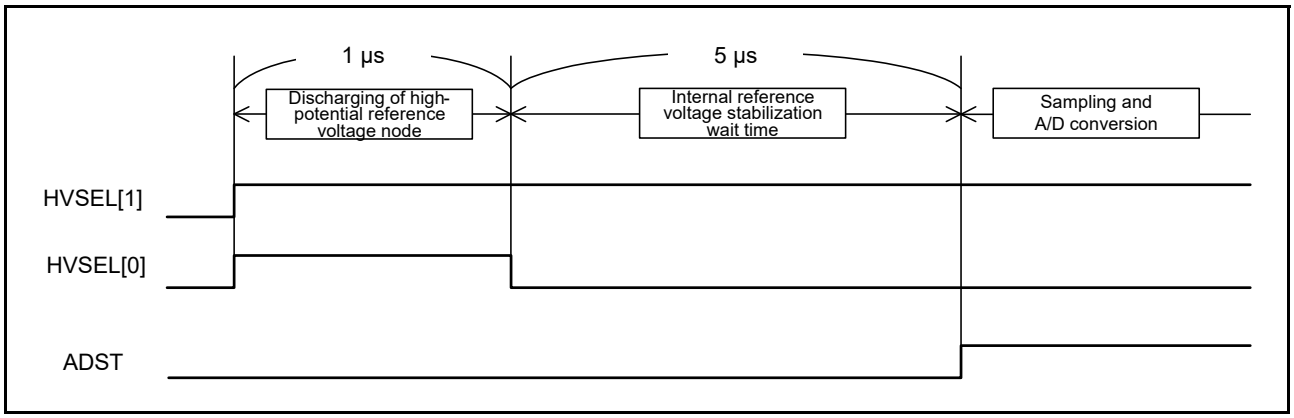


Figure 39.30 Procedure to select internal reference voltage as high-potential reference voltage

39.8 Usage Notes

39.8.1 Notes on Reading Data Registers

The following registers must be read in halfword units:

- A/D data registers
- 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 separately read, the A/D-converted value read initially might conflict with the A/D-converted value read subsequently. To prevent this, the data registers must not be read in byte units.

39.8.2 Notes on Stopping A/D Conversion

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 in Figure 39.31.

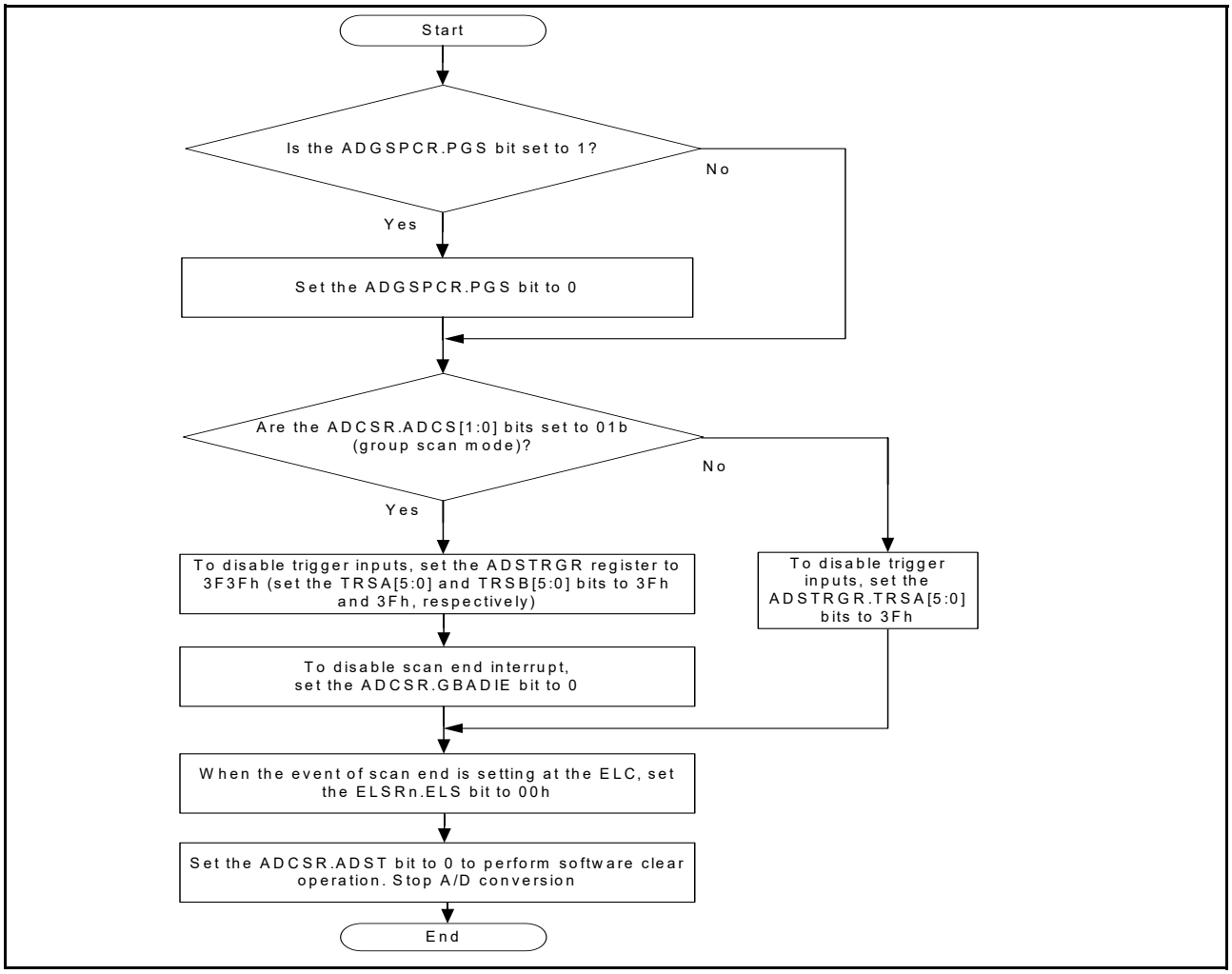


Figure 39.31 Procedures for clearing the ADCSR.ADST bit by software

39.8.3 A/D Conversion Restarting Timing and Termination Timing

It takes a maximum of 6 ADCLK cycles for the 14-bit ADC idle analog unit to be restarted by setting the ADST bit in ADCSR to 1. It takes a maximum of 3 ADCLK cycles for the 14-bit ADC operating analog unit to be terminated by setting the ADST bit in ADCSR to 0.

39.8.4 Notes 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 if the CPU does not complete reading out 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.

39.8.5 Module Stop Function Setting

Operation of the 14-bit ADC can be disabled or enabled using the Module Stop Control Register. The default setting is for operation of the 14-bit ADC to be stopped. Releasing the module-stop state enables access to the registers. After release from the module-stop state, wait for at least 1 µs before starting A/D conversion. For details, see section 11, Low Power Modes.

39.8.6 Notes on Entering Low Power Consumption States

Before entering the module-stop state or Software Standby mode, be sure to stop A/D conversion. Set the ADST bit in ADCSR to 0 and secure a period of time until the analog unit of the 14-bit ADC is stopped.

Follow the procedure for clearing the ADCSR.ADST bit by software, shown in Figure 39.31. Then, wait for 3 clock cycles of ADCLK before entering the module-stop state or Software Standby mode.

39.8.7 Error in Absolute Accuracy when Disconnection Detection Assistance is in Use

Using the disconnection detection assistance function leads to an error in absolute accuracy of the A/D converter. This error arises because an error 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. Only use disconnection detection assistance after thorough evaluation.

Maximum error in absolute accuracy (LSB) = 4095 × Rs/(Rs + Rp)

39.8.8 ADHSC Bit Rewriting Procedure

Before rewriting the A/D conversion select bit (ADCSR.ADHSC) from 0 to 1 or from 1 to 0, the A/D converter must be in the standby state. Carry out steps 1 to 3 in the following section to modify the ADCSR.ADHSC bit. After clearing the sleep bit (ADHVREFCNT.ADSLP) to 0, wait for at least 1 µs and then start A/D conversion.

[ADCSR.ADHSC bit rewriting procedure]

1. Set the sleep bit (ADHVREFCNT.ADSLP) to 1.
2. Wait for at least 0.2 µs, and then modify the A/D conversion select bit (ADCSR.ADHSC).
3. Wait for at least 4.8 µs, and then clear the sleep bit (ADHVREFCNT.ADSLP) to 0.

Note 1. It is prohibited to set the sleep bit (ADHVREFCNT.ADSLP) to 1 except for modifying the A/D conversion select bit (ADCSR.ADHSC)

Note 2. Do not reset the sleep bit while the A/D conversion select bit (ADCSR.ADHSC) is 1.
After the A/D conversion select bit (ADCSR.ADHSC) is set to 0 or the operating mode is transitioned to module stop mode, reset the sleep bit according to the ADCSR.ADHSC bit rewriting procedure.

39.8.9 Notes on Operating Modes and Status Bits

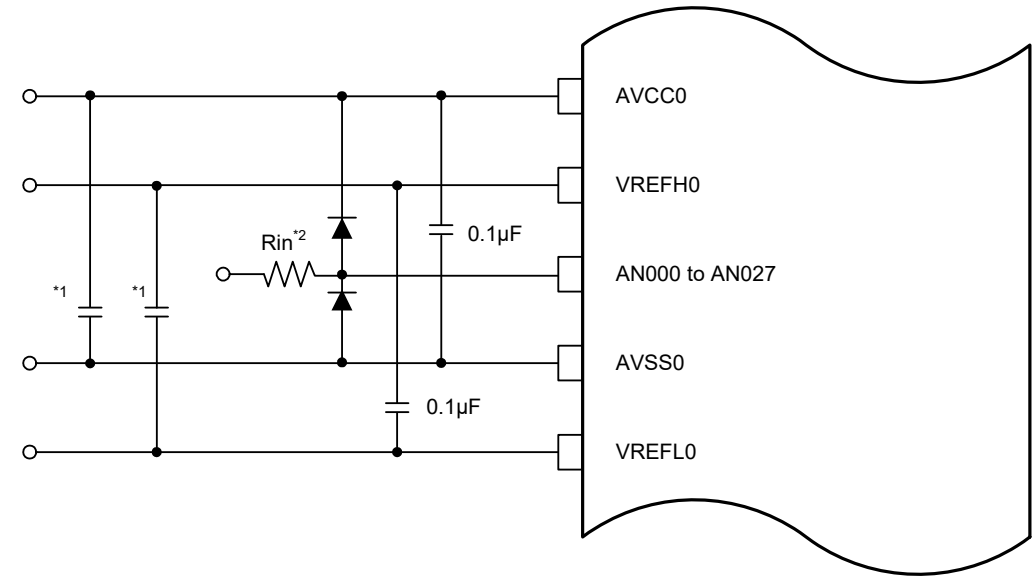
- The voltage values in self-diagnosis is possible to select ADCER.DIAGVAL[1:0] after setting ADCER.DIAGLD to 1.
- The double-trigger mode operates as the first scan after setting ADCSR.DBLE to 1 from 0.
- The status monitor bits (MONCMPA, MONCMPB, and MONCOMB) in the compare function are initialized after setting ADCMPCR.CMPAE and ADCMPCR.CMPBE to 0.

39.8.10 Notes on Board Design

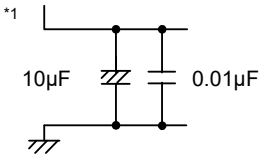
The board should be designed so that the digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or be placed near each other. If these rules are not followed, noise is produced on the analog signals and A/D conversion accuracy is affected. The analog input pins (AN000 to AN027), 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 plate connection).

39.8.11 Notes on Noise Prevention

To prevent the analog input pins (AN000 to AN027) from being destroyed by abnormal voltage such as excessive surge, a capacitor should be inserted between AVCC0 and AVSS0 and between VREFH0 and VREFL0, and a protection circuit should be connected to protect the analog input pins (AN000 to AN027) as shown Figure 39.32.



Note 1. The values shown here are reference values.



Note 2. Rin: Signal source impedance

Figure 39.32 Sample protection circuit for analog inputs

39.8.12 Port Setting when Using the 14-bit A/D Converter Input

When using the high-precision channels, do not use PORT0 as general I/O, IRQ8, IRQ9 inputs, and TS transmission. Renesas recommends that you do not use the digital output that is also used as the A/D analog input, if normal precision channels are used. If the digital output that is also used as the A/D 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.

39.8.13 Relationship between A/D converter, OPAMP, ACMPHS, and ACMPLP

The A/D conversion targets in Table 39.12 should not be selected as an OPAMP, ACMPHS, and ACMPLP input during A/D conversion.

Table 39.12 List of OPAMP, ACMPHS, and ACMPLP pin that should not be selected during A/D conversion

Target of A/D conversion	OPAMP	ACMPHS	ACMPLP
AN000	AMP0+	ACMPHS1.IVREF0, IVCMP0	-
AN001	AMP0-	ACMPHS1.IVREF1, IVCMP1	-
AN002	-	ACMPHS1.IVREF2, IVCMP2	-
AN003	-	ACMPHS1.IVREF3, IVCMP3	-
AN004	-	ACMPHS0.IVCMP0	-
AN005	AMP3+	ACMPHS0.IVREF0	-
AN006	AMP3-	ACMPHS0.IVREF1, ACMPHS1.IVREF4	-
AN007	-	ACMPHS0.IVCMP1, ACMPHS1.IVCMP4	-
AN010	AMP2-	-	-
AN011	AMP2+	-	-
AN012	AMP1-	-	-
AN013	AMP1+	-	-
AN014/DA0	-	ACMPHS0.IVREF2, ACMPHS1.IVREF5	-
AN015/DA1	-	ACMPHS0.IVCMP2, ACMPHS1.IVCMP5	-
AN024	-	-	CMPREF1
AN025	-	-	CMPIN1
AN026	-	-	CMPREF0
AN027	-	-	CMPIN0

39.8.14 Notes on Canceling Software Standby Mode

After the transition from Software Standby mode to normal mode, wait 1 µs before starting A/D conversion.

40. 12-Bit D/A Converter (DAC12)

40.1 Overview

The MCU provides a 12-bit D/A converter.

Table 40.1 lists the 12-bit D/A converter specifications and Figure 40.1 shows the block diagram.

Table 40.1 12-bit D/A converter specifications

Parameter	Specifications
Resolution	12 bits
Output channels	2 channels
Interference reduction between analog modules	Reduces interference between D/A and A/D conversion circuits. D/A converted data update timing is controlled by the synchronous D/A conversion enable input signal from the ADC14, which reduces the effect of DAC12 inrush current on A/D conversion accuracy.
Module-stop function	The module-stop state can be set to reduce power consumption
Event link function (input)	DA0 and DA1 conversion can be started on input of an event signal

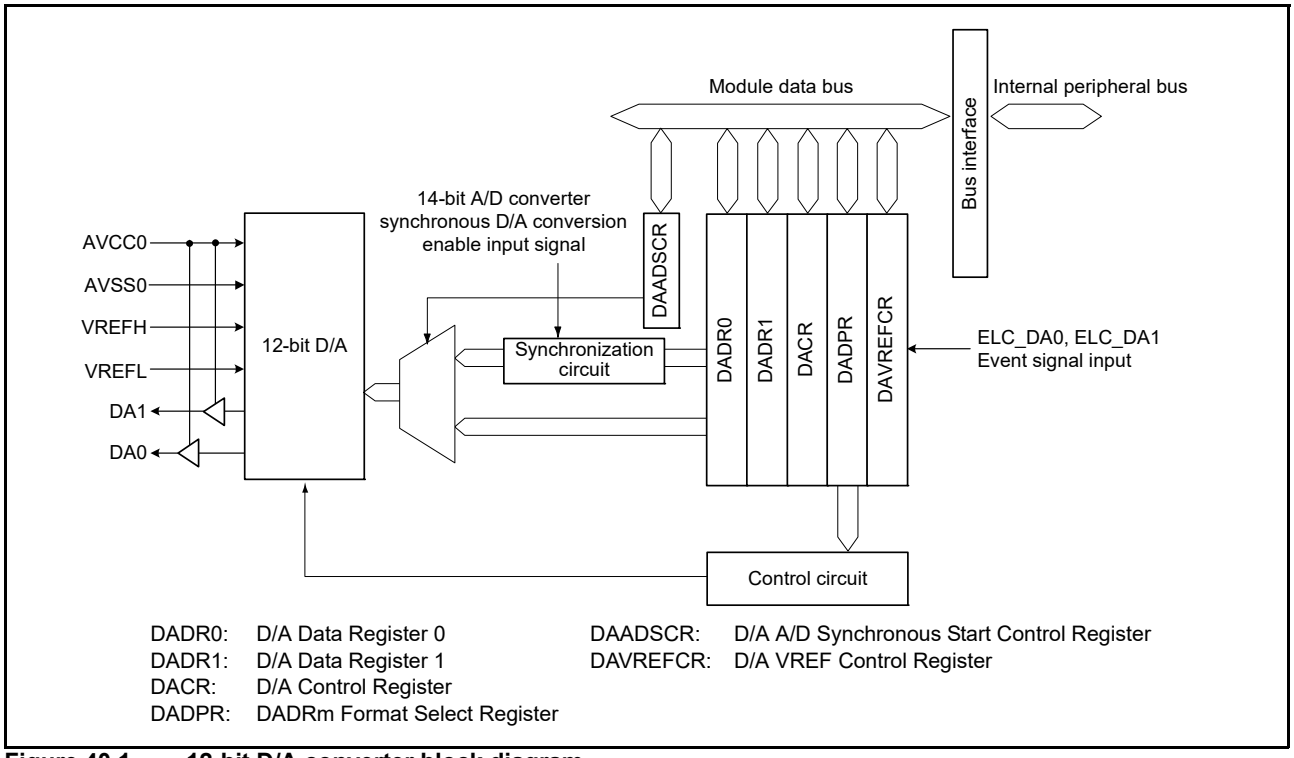


Figure 40.1 12-bit D/A converter block diagram

Table 40.2 lists the pin configuration of the 12-bit D/A converter.

Table 40.2 DAC12 pin configuration

Pin name	I/O	Function
AVCC0	Input	• Analog power supply pin for ADC14, DAC12, Comparator, and OPAMP • Connect to VCC when these modules are not used.
AVSS0	Input	• Analog ground pin for ADC14, DAC12, Comparator, and OPAMP • Connect to VSS when these modules are not used.
VREFH	Input	Analog reference top voltage supply pin for D/A converter
VREFL	Input	Analog reference ground pin for D/A converter
DA0	Output	Channel 0 analog output pin
DA1	Output	Channel 1 analog output pin

40.2 Register Descriptions

40.2.1 D/A Data Register m (DADRm) (m = 0, 1)

Address(es): DAC12.DADR0 4005 E000h, DAC12.DADR1 4005 E002h

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

The DADRm registers are 16-bit read/write registers that store data for D/A conversion. When an analog output is enabled, the values in DADRm are converted and output to the analog output pins. You can set the DADPR.DPSEL bit to format 12-bit data as left-justified or right-justified.

In right-justified format (DADPR.DPSEL = 0), the lower 12 bits (b11 to b0) are valid. In left-justified format (DADPR.DPSEL = 1), the upper 12 bits (b15 to b4) are valid.

40.2.2 D/A Control Register (DACR)

Address(es): DAC12.DACR 4005 E004h

	b7	b6	b5	b4	b3	b2	b1	b0
	DAOE1	DAOE0	—	—	—	—	—	—
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	DAOE0	D/A Output Enable 0	0: Disable analog output of channel 0 (DA0) 1: Enable D/A conversion and analog output of channel 0 (DA0).	R/W
b7	DAOE1	D/A Output Enable 1	0: Disable analog output of channel 1 (DA1) 1: Enable D/A conversion and analog output of channel 1 (DA1).	R/W

When interference reduction between D/A and A/D conversions is enabled (DAADSCR.DAADST = 1), set this register only while the ADC14 is halted (ADCSR.ADST = 0) and software trigger is selected as the ADC14 trigger.

DAOEi bit (D/A Output Enable i)

The DAOEi bit (i = 0, 1) enables D/A conversion and analog output. When interference reduction between D/A and A/D conversions is enabled (DAADSCR.DAADST = 1), only set the DAOEi bits while the ADC14 is halted (ADCSR.ADST = 0) and software trigger is selected as the ADC14.

The event link function can set the DAOEi bit to 1. The DAOE0 bit becomes 1 when the event specified in the ELSR12 register for the ELC_DA0 event occurs, and output of the D/A conversion results starts. The DAOE1 bit becomes 1 when the event specified in the ELSR13 register for the ELC_DA1 event occurs, and output of the D/A conversion results start.

40.2.3 DADR0 Format Select Register (DADPR)

Address(es): DAC12.DADPR 4005 E005h

	b7	b6	b5	b4	b3	b2	b1	b0
	DPSEL	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DPSEL	DADRm Format Select	0: Right-justified format 1: Left-justified format.	R/W

40.2.4 D/A A/D Synchronous Start Control Register (DAADSCR)

Address(es): DAC12.DAADSCR 4005 E006h

	b7	b6	b5	b4	b3	b2	b1	b0
	DAADST	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DAADST	D/A A/D Synchronous Conversion	0: Do not synchronize DAC12 with ADC14 operation (disable interference reduction between D/A and A/D conversion) 1: Synchronize DAC12 with ADC14 operation (enable interference reduction between D/A and A/D conversion).	R/W

To reduce interference between the D/A and A/D conversion, the DAADSCR register switches off or on the synchronization of the D/A conversion start with the synchronous D/A conversion enable input signal from the ADC14 trigger.

Set this register only while the ADC14 is halted (ADCSR.ADST = 0) and software trigger is selected as the ADC14 trigger.

DAADST bit (D/A A/D Synchronous Conversion)

Setting the DAADST bit to 0 allows the DADRm register value to be converted into analog data at any time. Setting the DAADST bit to 1 selects synchronous of D/A conversion with the synchronous D/A conversion enable input signal from the ADC14, which means that even if the DADRm register value is modified, D/A conversion does not start until the ADC14 completes A/D conversion.

Set this bit only while the ADC14 is halted (ADCSR.ADST = 0) and software trigger is selected as the ADC14 trigger. The event link function cannot be used when the DAADST bit is set to 1. Stop the event link function by setting the ELSR12 and ELSR13 registers of the ELC.

The setting of the DAADST bit is common to channels 0 and 1 of the DAC12.

40.2.5 D/A VREF Control Register (DAVREFCR)

Address(es): DAC12.DAVREFCR 4005 E007h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	REF[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	REF[2:0]	D/A Reference Voltage Select	b2 b0 0 0 0: No reference voltage selected 0 0 1: AVCC0/AVSS0 selected 0 1 1: Internal reference voltage/AVSS0 selected 1 1 0: VREFH/VREFL selected. Other settings are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The D/A VREF Control Register (DAVREFCR) selects the reference voltage of the DAC12.

REF[2:0] bits (D/A Reference Voltage Select)

The REF[2:0] bits select the reference voltage of the DAC12 channel 0 or 1. When changing the value of these bits, write 000b to the DAVREFCR.REF[2:0] bits in advance. Read the REF[2:0] bits after changing their value, and confirm that they are changed. When selecting the internal reference voltage, set the DADR0 and DADR1 registers to 0000h and discharge the VREF path before switching the voltage. As the path remains discharged after the reset is released, the internal reference voltage can be selected. For details on discharging, see section 40.3.2, Notes on Using the Internal Reference Voltage as the Reference Voltage. Do not rewrite this register during A/D conversion using the ADC14. If this register is rewritten, the accuracy of A/D conversion is not guaranteed. When the internal reference voltage is selected, the voltage generation circuit operates and current increases. This circuit does not automatically turned off even when the MCU enters Software Standby mode with the internal reference voltage selected.

40.3 Operation

The DAC12 includes D/A conversion circuits for two channels, each of which can operate independently. When the DAOEn bit (n = 0, 1) in the DACR register is set to 1, the DAC12 is enabled and the conversion result is output.

The following example shows D/A conversion on channel 0. Figure 40.2 shows the timing of this operation.

1. Set the data for D/A conversion in the DADR0 register and the data format in the DADPR.DPSEL bit.
2. Set the DACR.DAOE0 bit to 1 to start D/A conversion. The conversion result is output from the analog output pin DA0 after the conversion time tDCONV elapses. The conversion result continues to be output until DADR0 is written to again or the DAOE0 bit is set to 0. The output value is expressed by the following formula:

$$\frac{\text{Setting value of DADRm}}{4096} \times \text{Reference voltage}$$

3. To start another conversion, write another value to DADR0. The conversion result is output after the conversion time tDCONV elapses.
   When the DAADSCR.DAADST bit is 1 (interference reduction between D/A and A/D conversion is enabled), a maximum of one A/D conversion time is required for D/A conversion to start. When ADCLK is faster than the peripheral clock, a longer time might be required.
4. To disable analog output, set the DAOE0 bit to 0.

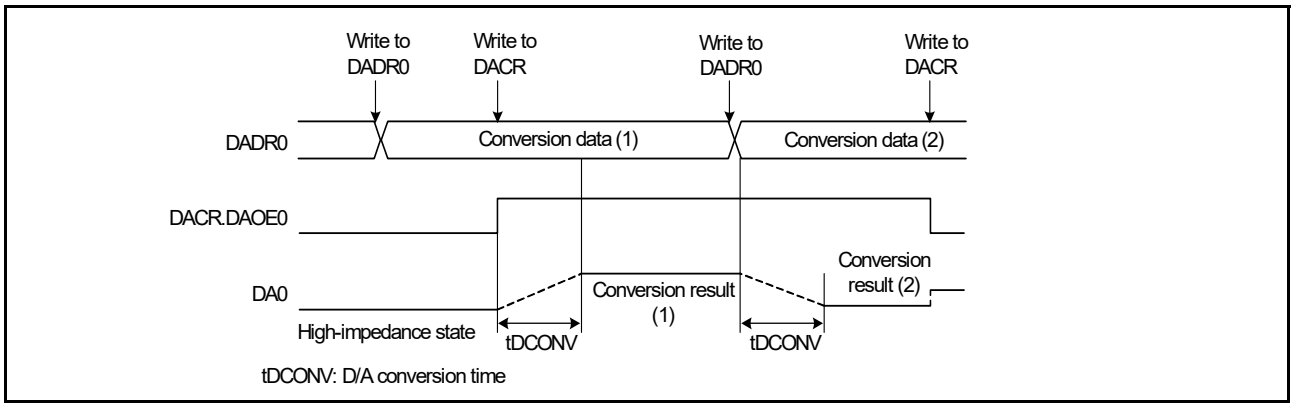


Figure 40.2 Example DAC12 operation

40.3.1 Minimizing Interference between D/A and A/D Conversion

When D/A conversion starts, the DAC12 generates inrush current. Because the DAC12 and ADC14 share the same analog power supply, the generated inrush current can interfere with 14-bit A/D conversion.

While the DAADSCR.DAADST bit is 1, D/A conversion does not start immediately on updating the DADRm register. Instead:

- If the DADRm register data is modified while the ADC14 is halted, D/A conversion starts in one PCLKB cycle
- If the DADRm register is modified while the ADC14 is in progress (ADCSR.ADST bit = 1), D/A conversion starts on A/D conversion completion. Therefore, it takes up to one A/D conversion time for the DADRm register data update to reflect as the D/A conversion circuit output. Until the D/A conversion completes, the DADRm register value does not correspond to the analog output value.

When the DAADSCR.DAADST bit is 1, it is not possible to check through any software means whether the DADRm register value was D/A converted.

The following sequence provides an example of D/A conversion, in which the DAC12 is synchronized with the ADC14.

1. Confirm that the ADC14 is halted and set the DAADSCR.DAADST bit to 1.
2. Confirm that the ADC14 is halted and set the DACR.DAOE0 bit to 1.
3. Set the DADR0 register. If ADCLK is faster than the peripheral clock, D/A conversion might be delayed for longer than one A/D conversion time.

- If the ADC14 is halted (ADCSR.ADST bit = 0) when the DADR0 register is modified, D/A conversion starts in one PCLKB cycle.
- If the ADC14 is in progress (ADCSR.ADST bit = 1) when the DADR0 register is modified, D/A conversion starts on A/D conversion completion. If the DADR0 register is modified twice during A/D conversion, the first update might not be converted.

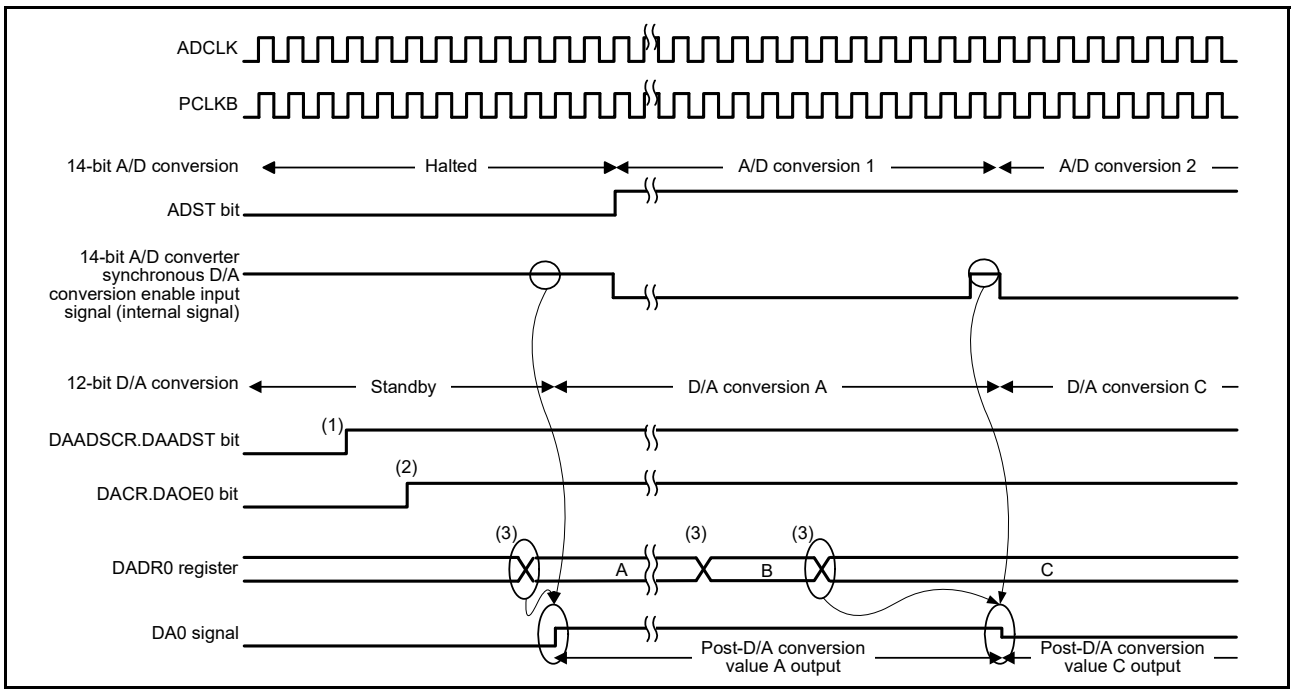


Figure 40.3 Example conversion when DAC12 is synchronized with ADC14

When ADCLK is faster than PCLKB, the DAC12 might not be able to capture the synchronous D/A conversion enable input signal from the ADC14 during the one ADCLK output cycle between A/D conversion 1 and A/D conversion 2, as shown in Figure 40.4. In this case, post-D/A conversion value A is continuously output as the DA0 signal.

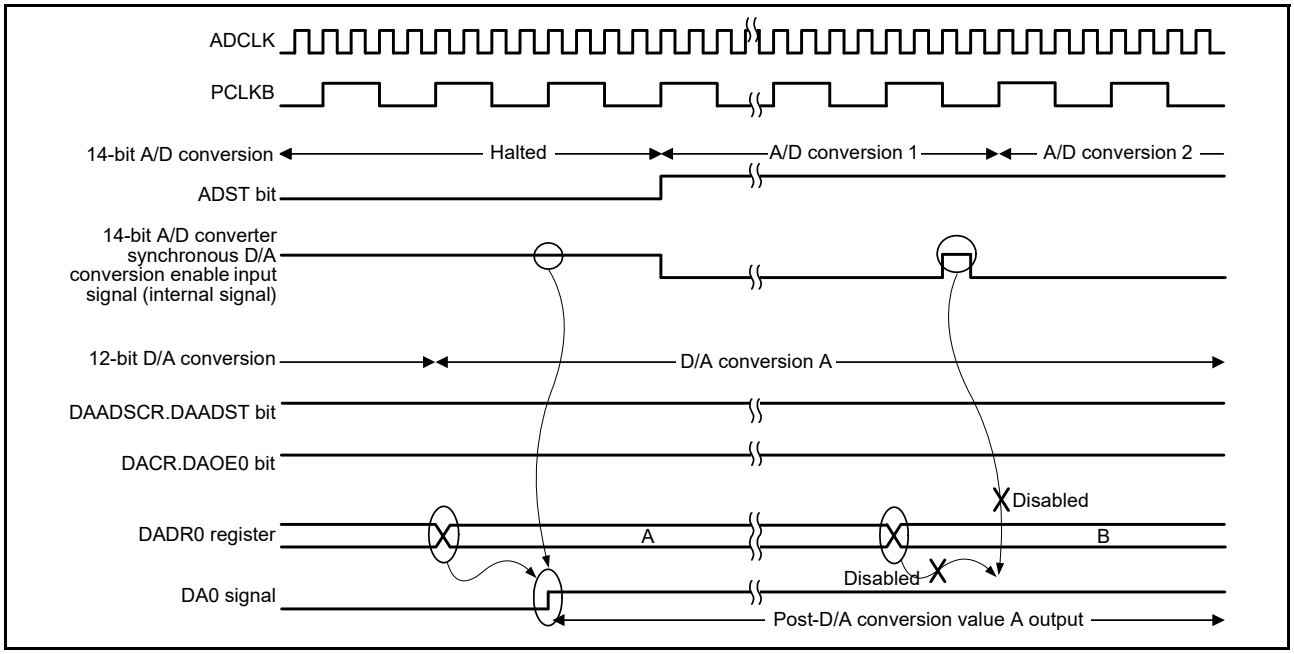


Figure 40.4 Example when the DAC12 cannot capture the synchronous D/A conversion enable input signal from the ADC14

40.3.2 Notes on Using the Internal Reference Voltage as the Reference Voltage

When setting the DAVREFCR.REF[2:0] bits to 011b to use the internal reference voltage/AVSS0 as the reference voltage, the VREF path must be discharged before selecting the voltage. The following sequence shows the discharging procedure:

1. Write 000b to the REF[2:0] bits.
2. Set the DADR0 register to 0000h and the DADR1 register to 0000h.
3. Keep the state of step 2. for 10 µs (discharging).
4. After discharging is complete, write 011b to the DAVREFCR.REF[2:0] bits and select the internal reference voltage/AVSS0.
5. Set the DACR.DAOEn bit to 1 and wait 5 µs stabilization time of the internal reference voltage.
6. Write data to the DADRm register and start D/A conversion.

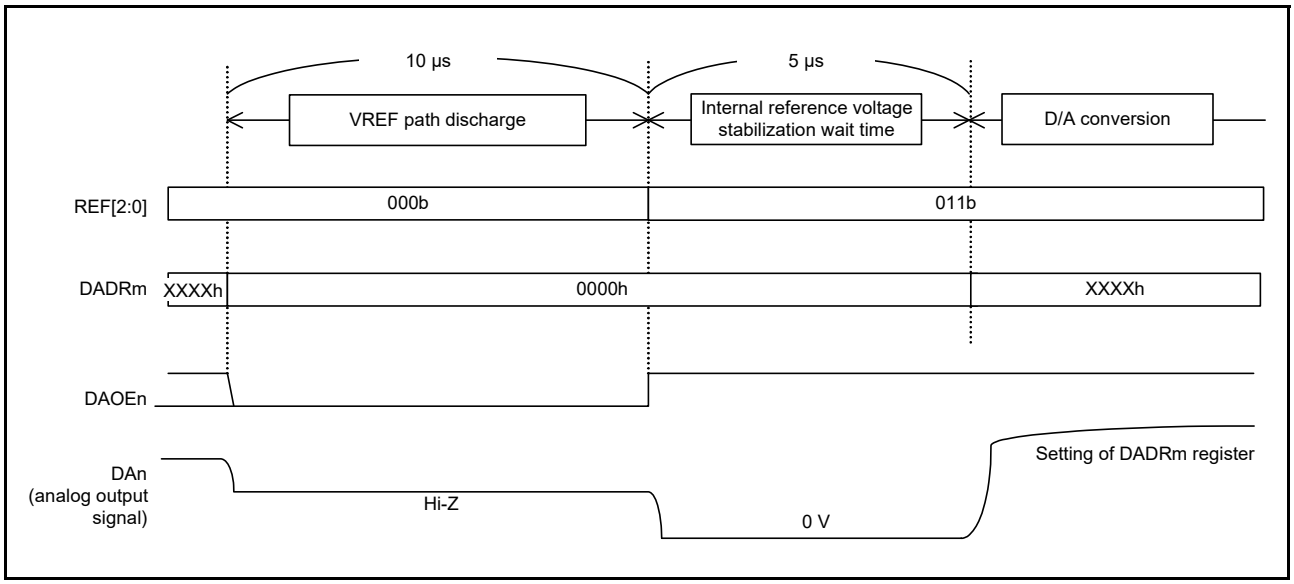


Figure 40.5 Procedure for selecting the internal reference voltage as the reference voltage

40.4 Event Link Operation Setting Procedure

The event link operation procedure is described in the following sections.

40.4.1 DA0 event link operation setting procedure

1. Set the DADPR.DPSEL bit and set the data for D/A conversion in the DADR0 register.
2. Set the ELC_DA0 event signal to be linked to each peripheral module in the ELSR12 register.
3. Set the ELCR.ELCON bit to 1. This step enables event link operation for all modules with the selected event link function.
4. Set the event output source module to activate the event link. After the event is output from the module, the DACR.DAOE0 bit becomes 1, and D/A conversion on channel 0 starts.
5. Set the ELSR12.ELS[8:0] bits to 000h to stop event link operation of the DAC12 channel 0. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

40.4.2 DA1 event link operation setting procedure

1. Set the DADPR.DPSEL bit and set the data for D/A conversion in the DADR1 register.
2. Set the ELC_DA1 event signal to be linked to each peripheral module in the ELSR13 register.
3. Set the ELCR.ELCON bit to 1. This step enables event link operation for all modules with the selected event link function.
4. Set the event output source module to activate the event link. After the event is output from the module, the DACR.DAOE1 bit becomes 1, and the D/A conversion on channel 1 starts.
5. Set the ELSR13.ELS[8:0] bits to 000h to stop event link operation of the DAC12 channel 1. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

40.5 Usage Notes on Event Link Operation

- When the event specified by the ELC_DA0 event signal is generated while a write to the DACR.DAOE0 bit is being performed, the write cycle is stopped, and the generated event takes precedence in setting the bit to 1
- When the event specified by the ELC_DA1 event signal is generated while a write to the DACR.DAOE1 bit is being performed, the write cycle is stopped, and the generated event takes precedence in setting the bit to 1
- Use of the event link function is prohibited when the DAADSCR.DAADST bit is set to 1 to reduce interference between D/A and A/D conversions.

40.6 Usage Notes

40.6.1 Settings for the Module-Stop Function

The Module Stop Control register can enable or disable DAC12 operation. The DAC12 is stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

40.6.2 DAC12 Operation in Module-Stop State

When the MCU enters the module-stop state with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current must be reduced in the module-stop state, disable D/A conversion by setting the DACR.DAOE1 and DACR.DAOE0 bits to 0.

40.6.3 DAC12 Operation in Software Standby Mode

When the MCU enters Software Standby mode with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current must be reduced in Software Standby mode, disable D/A conversion by setting the DACR.DAOE1 and DACR.DAOE0 bits to 0.

40.6.4 Note on Usage when Interference Reduction between D/A and A/D Conversion is Enabled

When the DAADSCR.DAADST bit is 1, enabling interference reduction between D/A and A/D conversion, do not place the ADC14 in the module-stop state. Doing so can halt D/A conversion in addition to A/D conversion.

41. Temperature Sensor (TSN)

41.1 Overview

The on-chip temperature sensor can determine and monitor 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 ADC14 for conversion and can also be used by the end application. Table 41.1 lists the specifications of the temperature sensor and Figure 41.1 shows the block diagram.

Table 41.1 Temperature sensor specifications

Parameter	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the 14-bit A/D converter

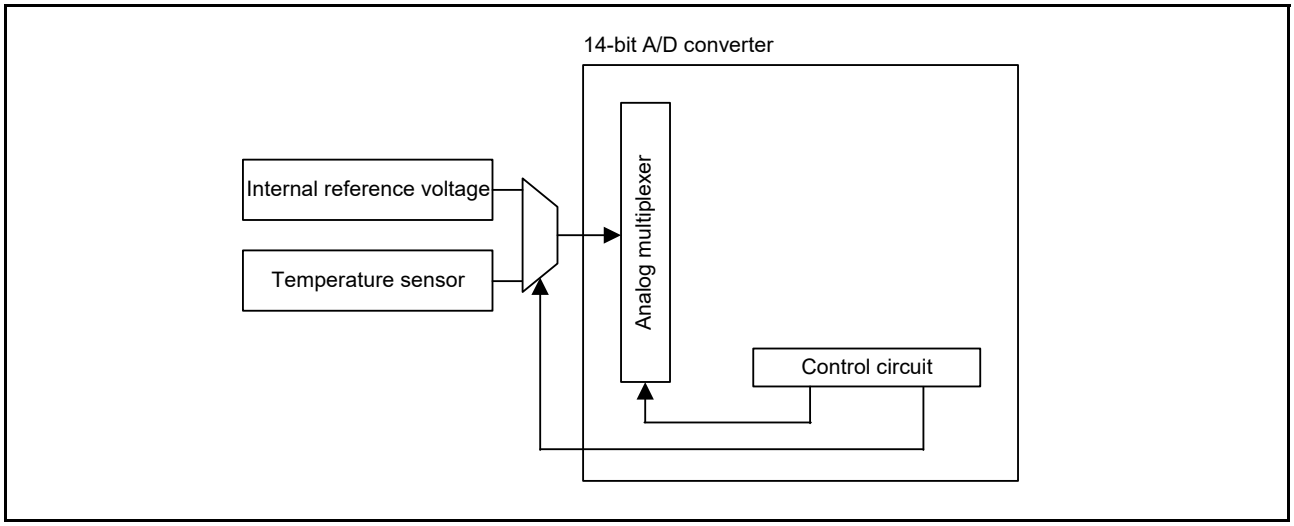
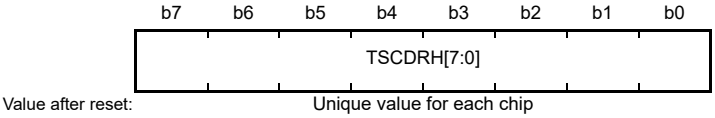


Figure 41.1 Temperature sensor block diagram

41.2 Register Descriptions

41.2.1 Temperature Sensor Calibration Data Register H (TSCDRH)

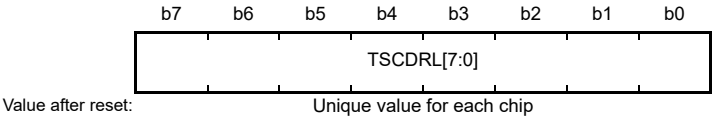
Address(es): TSN.TSCDRH 407E C229h



Bit	Symbol	Bit name	Description	R/W
b7 to b0	TSCDRH[7:0]	Temperature Sensor Calibration Data	The calibration data stores the upper 4 bits of the converted value	R

41.2.2 Temperature Sensor Calibration Data Register L (TSCDRL)

Address(es): TSN.TSCDRL 407E C228h



Bit	Symbol	Bit name	Description	R/W
b7 to b0	TSCDRL[7:0]	Temperature Sensor Calibration Data	The calibration data stores the lower 8 bits of the converted value.	R

At factory shipment, the TSCDRH and TSCDRL registers store temperature sensor calibration data measured for each MCU. Temperature sensor calibration data is a digital value obtained using the ADC14 to convert the voltage output by the temperature sensor under the condition Ta = Tj = 125°C and AVCC0 = 3.3 V. The TSCDRH register stores the upper 4 bits of the converted value, and the TSCDRL register stores the lower 8 bits.

41.3 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 ADC14. To obtain the die temperature, convert this value into the temperature.

41.3.1 Preparation for Using Temperature Sensor

The temperature (T) is proportional to the sensor voltage output (Vs), so temperature is calculated with the following formula:

$$T = (Vs - V1)/slope + T1$$

T: Measured temperature (°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, therefore Renesas recommends measuring two different sample temperatures as follows:

1. Use the ADC14 to measure the voltage V1 output by the temperature sensor at temperature T1.
2. Again using the ADC14, 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 52, Electrical Characteristics, use the ADC14 to measure the voltage V1 output by the temperature sensor at temperature T1, and then calculate the temperature characteristic by using the following formula:

$$T = (Vs - V1)/Slope + T1$$

Note: This method produces less accurate temperatures than measurement at two points.

In the MCU, the TSCDRH and TSCDRL registers store the temperature value (CAL125) of the temperature sensor measured under the condition Ta = Tj = 125°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.

This measured value CAL125 can be calculated as follows:

$$CAL125 = (\text{TSCDRH register value} << 8) + \text{TSCDRL register value}$$

If V1 is calculated from CAL125,

$$V1 = 3.3 \times CAL125/4096 \text{ [V]}$$

Using this value, the measured temperature can be calculated according to the following formula:

$$T = (Vs - V1)/Slope + 125 \text{ [°C]}$$

- T: Measured temperature (°C)
- Vs: Voltage output by the temperature sensor when the temperature is measured (V)
- V1: Voltage output by the temperature sensor when Ta = Tj = 125°C and AVCC0 = 3.3 V (V)
- Slope: Temperature gradient of the temperature sensor ÷ 1,000 (V/°C)

Figure 41.2 shows the error in the measured temperature. The variation range is 3σ.

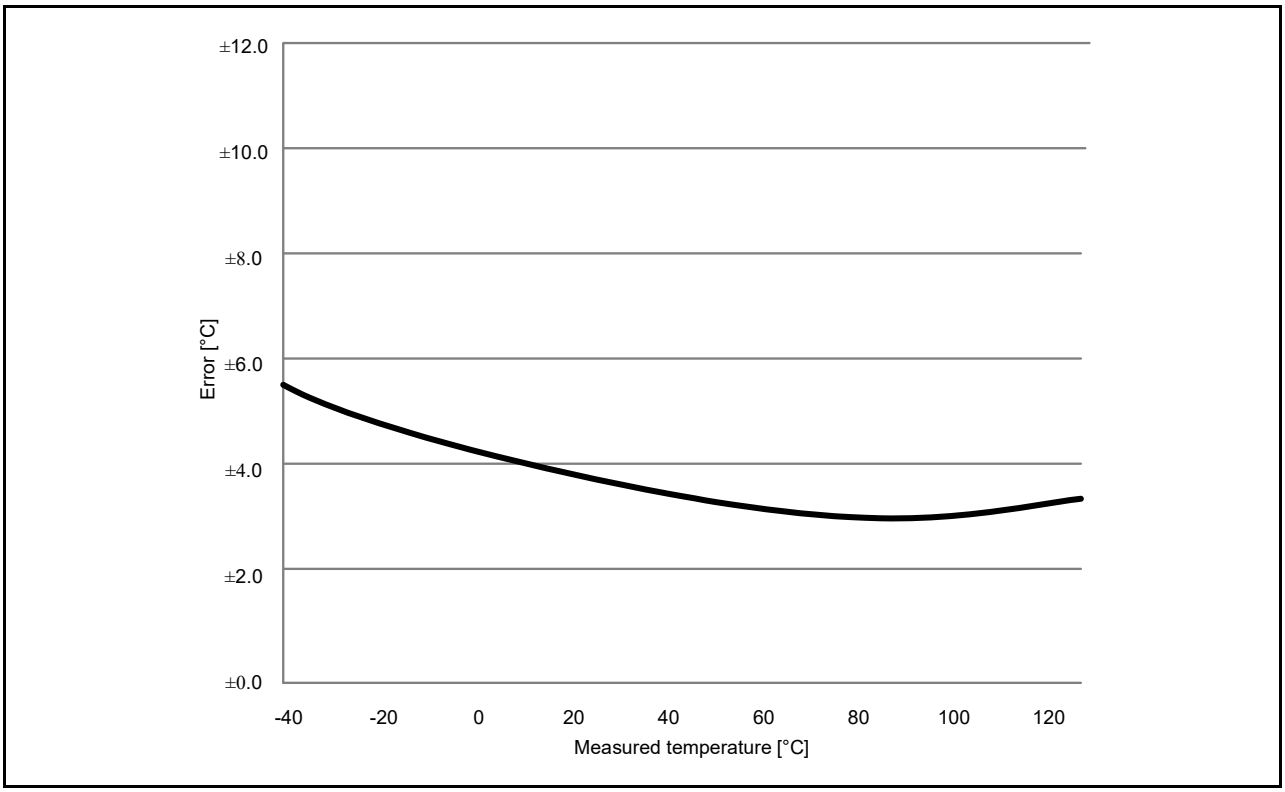


Figure 41.2 Error in the measured temperature (designed values)

41.3.2 Procedure for Using the Temperature Sensor

For details, see section 39, 14-Bit A/D Converter (ADC14).

42. Operational Amplifier (OPAMP)

42.1 Overview

Operational amplifiers can be used to amplify small analog input voltages and output the amplified voltages. The MCU has a total of four differential operational amplifier units with two input pins and one output pin.

The operational amplifiers have the following functions:

- Operational amplifiers can be used to input signals to the respective positive and negative sides of the ACMPHS that provides output signals from the units
- The output signals from all units can be used for the input signals to the A/D converter
- High-speed mode (high-current consumption) and low power mode (slow-speed response) are supported and either mode can be selected based on trade-offs between the response speed and current consumption
- Operation can be started by each trigger from the Asynchronous General purpose Timer (AGT)
- Operation can be stopped by an A/D conversion end trigger.

The number of operational amplifier input and output pins differ depending on the product.

Figure 42.1 shows a block diagram of the operational amplifier and Table 42.2 lists the unit configuration.

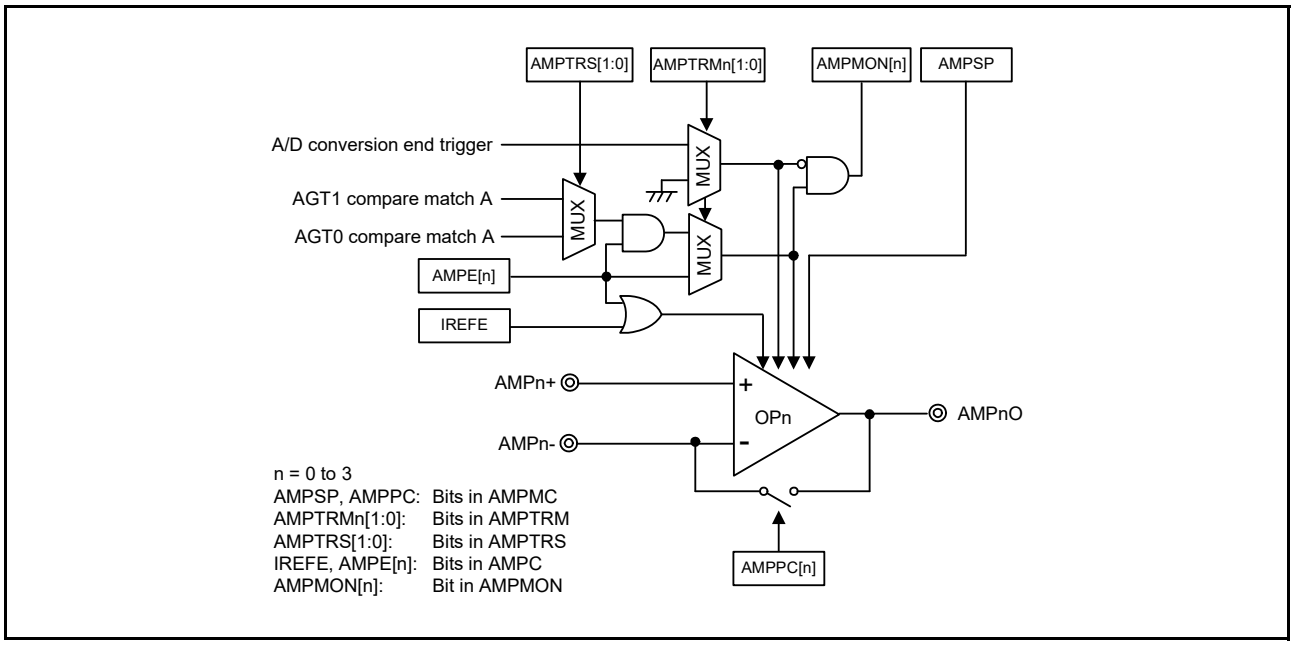


Figure 42.1 Operational amplifier block diagram

Table 42.1 OPAMP unit configuration

Unit	I/O pin	I/O	Function
Unit 0 (Operational amplifier 0)	AMP0+, AMP0-	Input	Input pin of operational amplifier 0 (+, -)
	AMP0O	Output	Output pin of operational amplifier 0
Unit 1 (Operational amplifier 1)	AMP1+, AMP1-	Input	Input pin of operational amplifier 1 (+, -)
	AMP1O	Output	Output pin of operational amplifier 1
Unit 2 (Operational amplifier 2)	AMP2+, AMP2-	Input	Input pin of operational amplifier 2 (+, -)
	AMP2O	Output	Output pin of operational amplifier 2
Unit 3 (Operational amplifier 3)	AMP3+, AMP3-	Input	Input pin of operational amplifier 3 (+, -)
	AMP3O	Output	Output pin of operational amplifier 3

42.2 Register Descriptions

42.2.1 Operational Amplifier Mode Control Register (AMPMC)

Address(es): OPAMP.AMPMC 4008 6008h

	b7	b6	b5	b4	b3	b2	b1	b0
	AMPSP	—	—	—	AMPPC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	AMPPC[3:0]	Operational Amplifier Precharge Control	AMPPC[n] (n = 0 to 3) 0: Precharging of operational amplifier n is stopped 1: Precharging of operational amplifier n is enabled.	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	AMPSP	Operational Amplifier Operation Mode Selection	0: Low power mode (low-speed) 1: High-speed mode.	R/W

Note: Set AMPSP bit while the value of the AMPC register is 00h (operational amplifier and reference current generator are stopped).

Note: Be sure to set bits that are not used in this register to the initial value.

42.2.2 Operational Amplifier Trigger Mode Control Register (AMPTRM)

Address(es): OPAMP.AMPTRM 4008 6009h

	b7	b6	b5	b4	b3	b2	b1	b0
	AMPTRM3[1:0]		AMPTRM2[1:0]		AMPTRM1[1:0]		AMPTRM0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	AMPTRM0[1:0]	OPAMP Function Activation/ Stop Trigger Control n*2	AMPTRMn[1] AMPTRMn[0] (n = 0 to 3) 0 0: Software trigger mode: •The operational amplifier can be activated/ stopped by setting the AMPC register •The operational amplifier cannot be activated by an activation trigger •The operational amplifier cannot be controlled by an A/D conversion end trigger. 0 1: Activation trigger mode: •The operational amplifier can be set to wait for an activation trigger or stopped by setting the AMPC register •The operational amplifier can be activated by an activation trigger*1 •The operational amplifier cannot be controlled by an A/D conversion end trigger. 1 0: Setting prohibited 1 1: Activation and A/D trigger mode: •The operational amplifier can be set to wait for an activation trigger or stopped by setting the AMPC register •The operational amplifier can be activated by an activation trigger*1 •The operational amplifier can be stopped by an A/D conversion end trigger. An A/D conversion end trigger is always generated at the end of A/D conversion.	R/W
b3, b2	AMPTRM1[1:0]			
b5, b4	AMPTRM2[1:0]			
b7, b6	AMPTRM3[1:0]			

Note: An A/D conversion end trigger is always generated at the end of A/D conversion.

Note 1. When using an activation trigger to activate the operational amplifier, first specify settings related to the AGT, set the AMPTRS register, and then use the AMPC register to set the operation control bit of the operational amplifier to be activated to 1 (operational amplifier wait state is enabled).

Note 2. When changing the set values of AMPTRMn[1:0], make sure that the AMPE[n] bit in the AMPC register is 0 (operation amplifier is stopped).

42.2.3 Operational Amplifier Activation Trigger Select Register (AMPTRS)

Address(es): OPAMP.AMPTRS 4008 600Ah

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

Bit	Symbol	Bit name	Description	R/W
b1, b0	AMPTRS[1:0]	Activation Trigger Selection*1	b1 b0 0 0: Operational amplifier n: Operational amplifier activation trigger n (n = 0 to 3) 0 1: Operational amplifier m: Operational amplifier activation trigger 0 (m = 0, 1) Operational amplifier n: Operational amplifier activation trigger 1 (n = 2, 3) 1 0: Setting prohibited 1 1: Operational amplifier n: Operational amplifier activation trigger 0 (n = 0 to 3).	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Be sure to set bits that are not used in this register to the initial value.
Note 1. Do not change the value of the AMPTRS register after setting the AMPTRM register.

42.2.4 Operational Amplifier Control Register (AMPC)

Address(es): OPAMP.AMPC 4008 600Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	IREFE	—	—	—	AMPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	AMPE[3:0]	OPAMP Operation Control	0: Operation amplifier n is stopped. 1: Software trigger mode: Operation of operational amplifier *n* is enabled.*1 Activation trigger mode or activation and A/D trigger mode: Wait until AGT is enabled (n = 0 to 3).	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	IREFE	OPAMP Reference Current Circuit Operation Control	0: Operational amplifier reference current circuit is stopped. 1: Operation of operational amplifier reference current circuit is enabled.	R/W

Note: Be sure to set bits that are not used in this register to the initial value.
Note 1. Operation of the operational amplifier reference current circuit is also enabled regardless of the IREFE bit setting. Be sure to set the bits to 0 for a unit that is not to be used.

Table 42.2 shows operational amplifier activation triggers corresponding to events.

Table 42.2 Operational amplifier activation triggers corresponding to events

Trigger	Event
Operational Amplifier activation trigger 0	AGT1 compare match A
Operational Amplifier activation trigger 1	AGT0 compare match A
Operational Amplifier activation trigger 2	AGT1 compare match A
Operational Amplifier activation trigger 3	AGT0 compare match A

42.2.5 Operational Amplifier Monitor Register (AMPMON)

Address(es): OPAMP.AMPMON 4008 600Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	AMPMON[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	AMPMON[3:0]	Operational Amplifier n Status (n = 0 to 3)	0: Operational amplifier n is stopped 1: Operational amplifier n is operating.	R
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note: This register is used to asynchronously reflect whether each operational amplifier is operating/stopped. To determine the operational amplifier state, read this register continuously to determine when the bit state changes. When an activation trigger or A/D conversion end trigger synchronized with the clock or a software trigger in the other interrupt routine is used to control the operational amplifier, the timing to operate/stop the operational amplifier can be estimated, such as for checking normal operation. In this case, read this register after one CPU/peripheral clock cycle when the corresponding trigger or interrupt affecting the operational amplifier state occurs. Be sure to set bits that are not used in this register to the initial value.

42.3 Operation

42.3.1 State Transitions

Figure 42.2 shows state transitions when the operational amplifier and reference current circuit are activated or stopped using the operational amplifier control circuit.

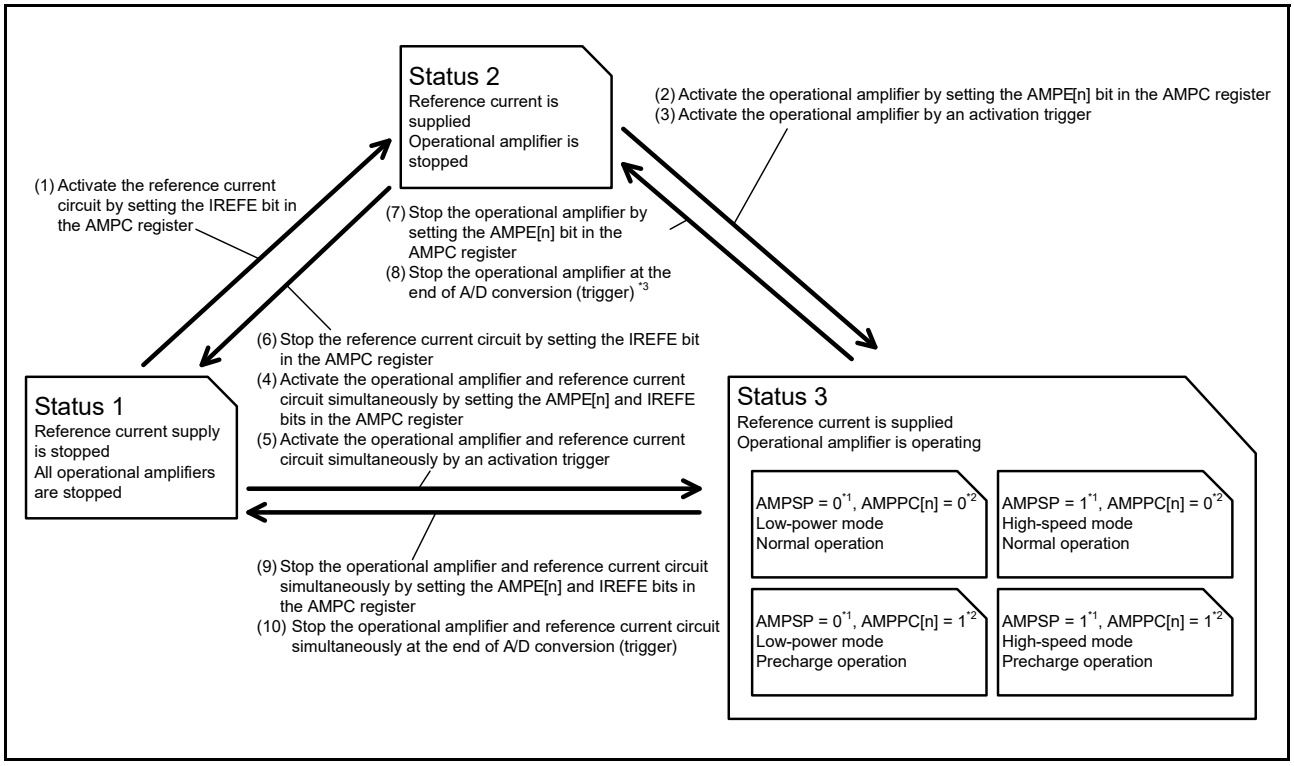


Figure 42.2 Operational amplifier state transitions

Note 1. Set the AMPSP bit in the AMPMC register and the AMPTRS and AMPTRM registers in status 1.
Note 2. Set the AMPPC[n] bit in the AMPMC register in status 3.
Note 3. To stop only the operational amplifier at the end of A/D conversion, it is necessary to preset operation of the reference current circuit to be enabled (operate the operational amplifier by status 2).

A stabilization wait time is necessary after supply of the reference current and operation of the operational amplifier are set before each operation actually starts. For details on the stabilization wait time, see section 52, Electrical Characteristics.

The operational amplifier cannot be activated/stopped continuously in steps (2) → (8), (2) → (10), (3) → (10), and (4) → (10).

An activation trigger and end of A/D conversion can be used to activate/stop only the operational amplifier that is preset to be used by setting the AMPTRM register.

42.3.2 Operational Amplifier Control Operation

Figure 42.3 to Figure 42.6 show operational amplifier control operation.

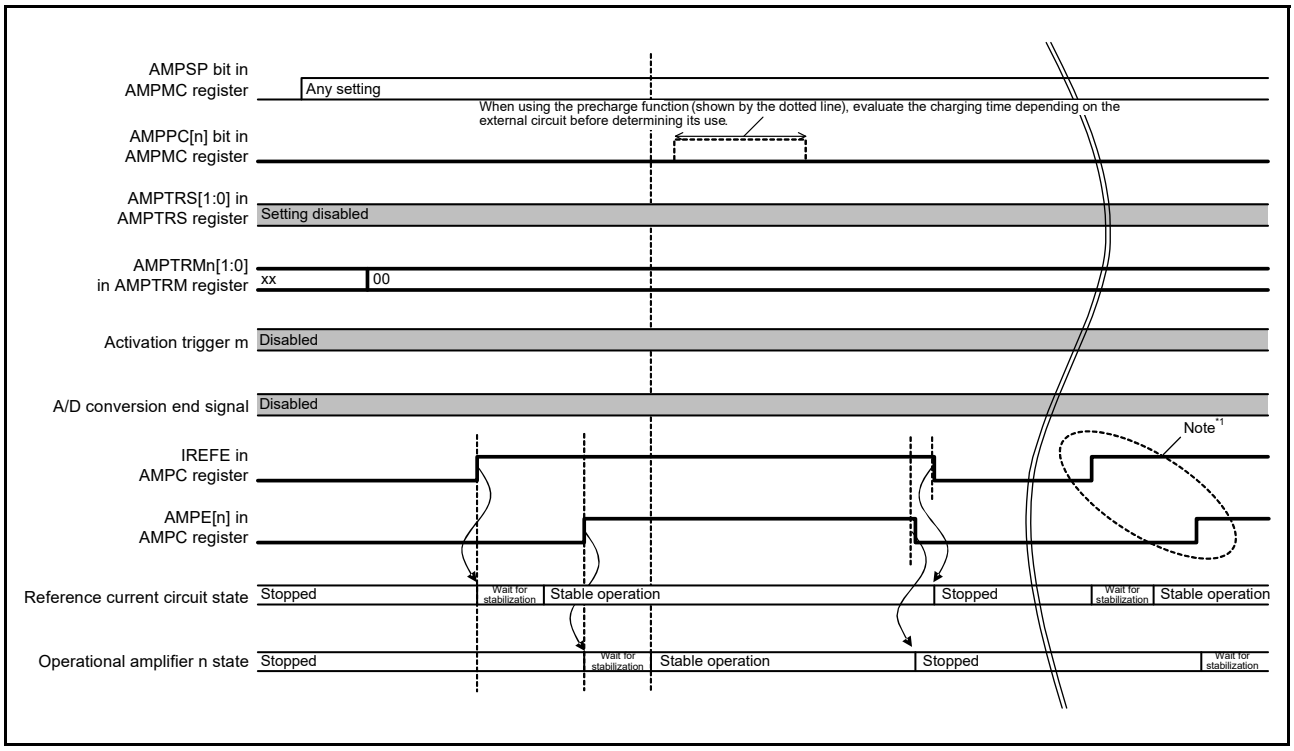


Figure 42.3 Operational amplifier control operation in software trigger mode is used for control when the reference current circuit and operational amplifier are activated/stopped by software trigger mode

Note 1. When operating/stopping the operational amplifier continuously, set the IREFE and AMPE[n] bits again as in the first setting after the operational amplifier is stopped.

Note: n: Unit number (n = 0 to 3)
m: An activation trigger used to control operational amplifier unit n selected by the AMPTRS register

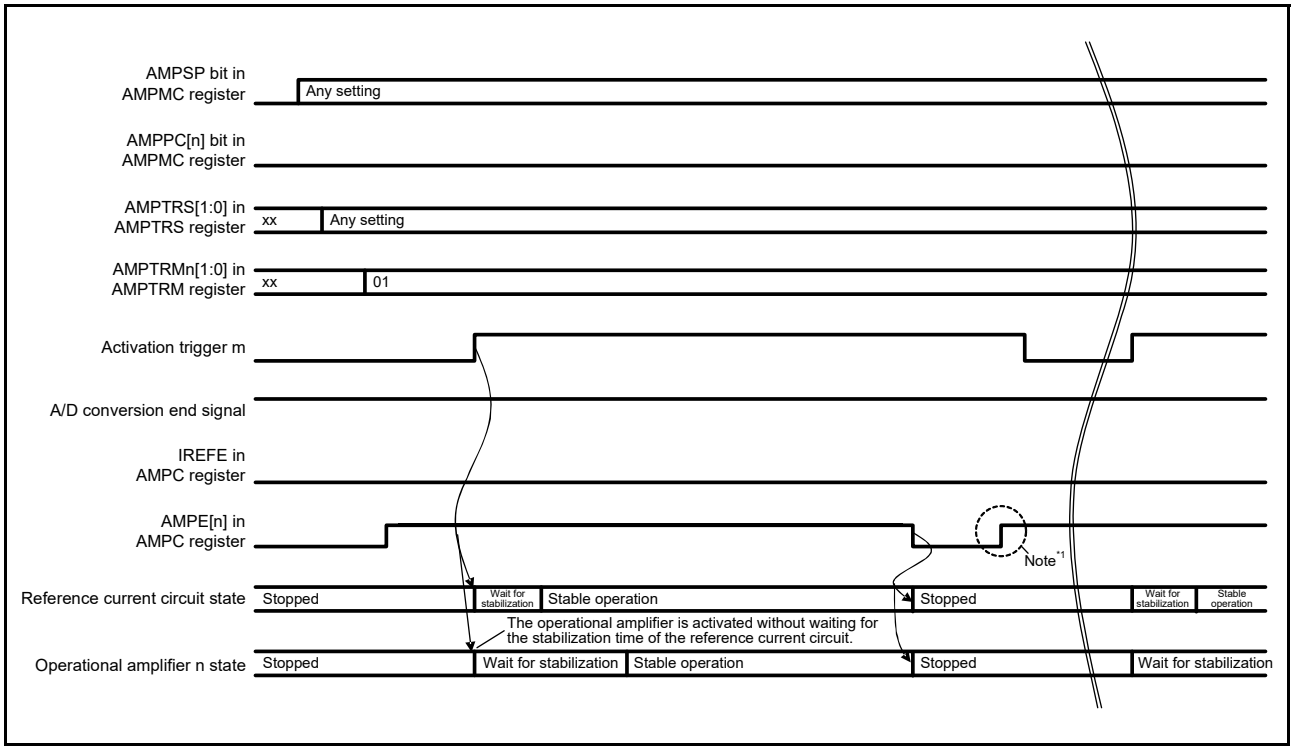


Figure 42.4 Operational amplifier control operation when activation trigger mode is used for activation, when the reference current circuit and operational amplifier are activated by an activation trigger and stopped by setting the AMPC register

Note 1. When operating/stopping the operational amplifier continuously, use the AMPE[n] bit again as in the first setting, and set the operational amplifier to wait for an activation trigger after it is stopped.

Note: n: Unit number (n = 0 to 3)
m: An activation trigger used to control operational amplifier unit n selected by the AMPTRS register. Set the AGT function.

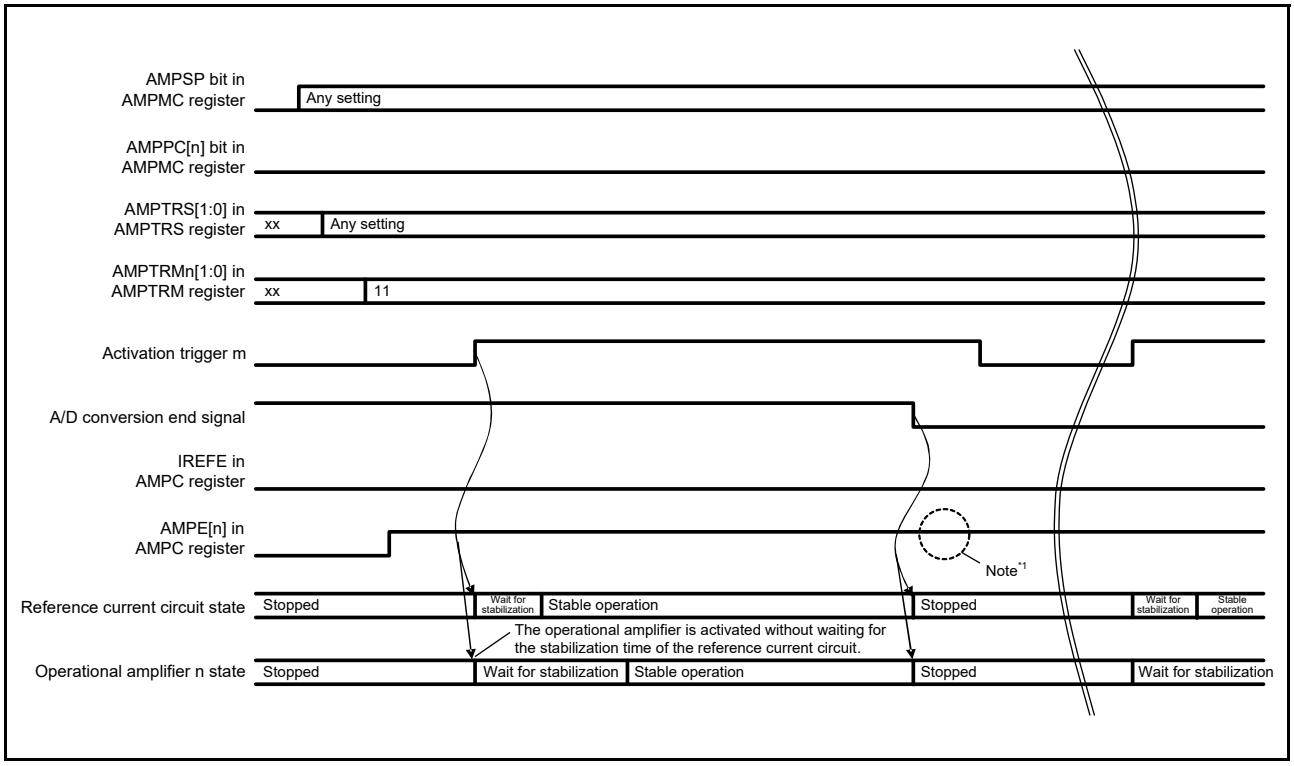


Figure 42.5 Operational amplifier control operation in activation and A/D trigger mode (1) when the reference current circuit and operational amplifier are activated by an activation trigger and stopped by an A/D conversion end (trigger)

Note 1. When operating/stopping the operational amplifier continuously, it is not necessary to set the registers again because the operational amplifier waits for an activation trigger after it is stopped.

Note: n: Unit number (n = 0 to 3)
m: An activation trigger used to control operational amplifier unit n selected by the AMPTRS register
Set the AGT function.

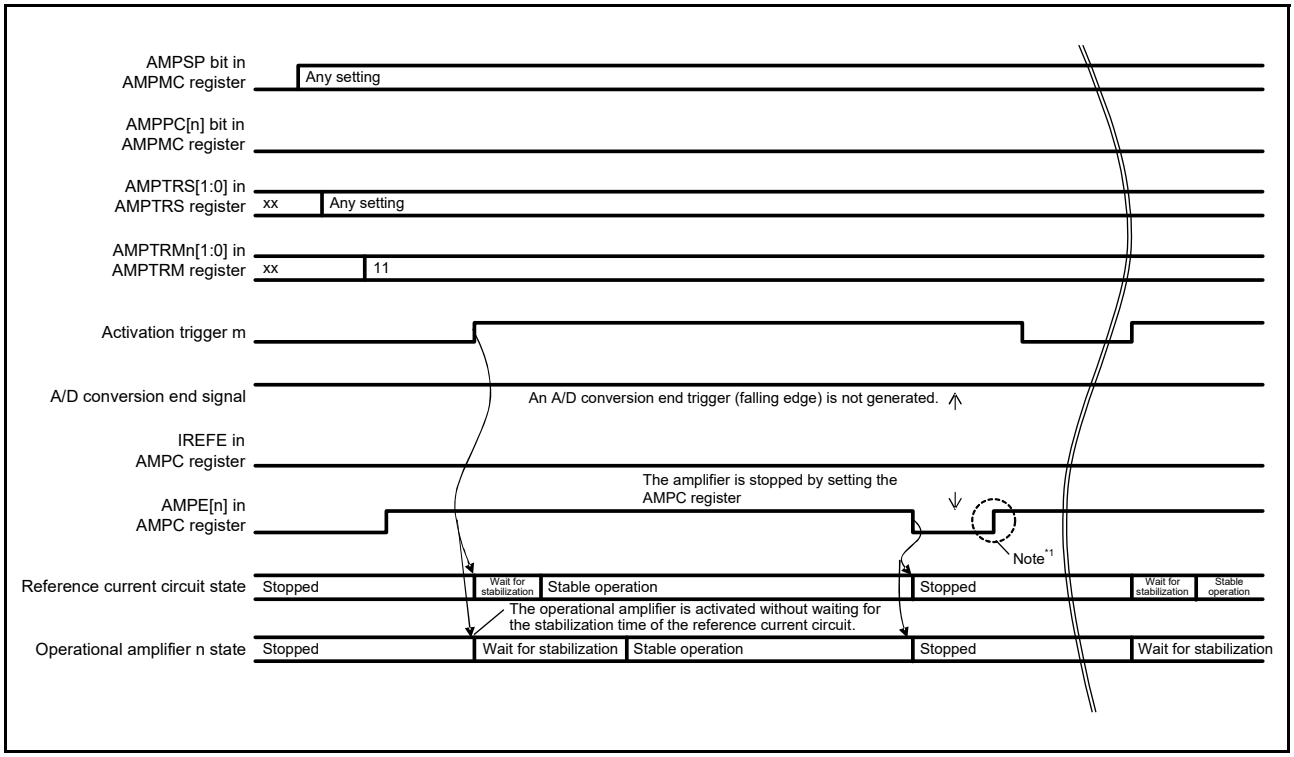


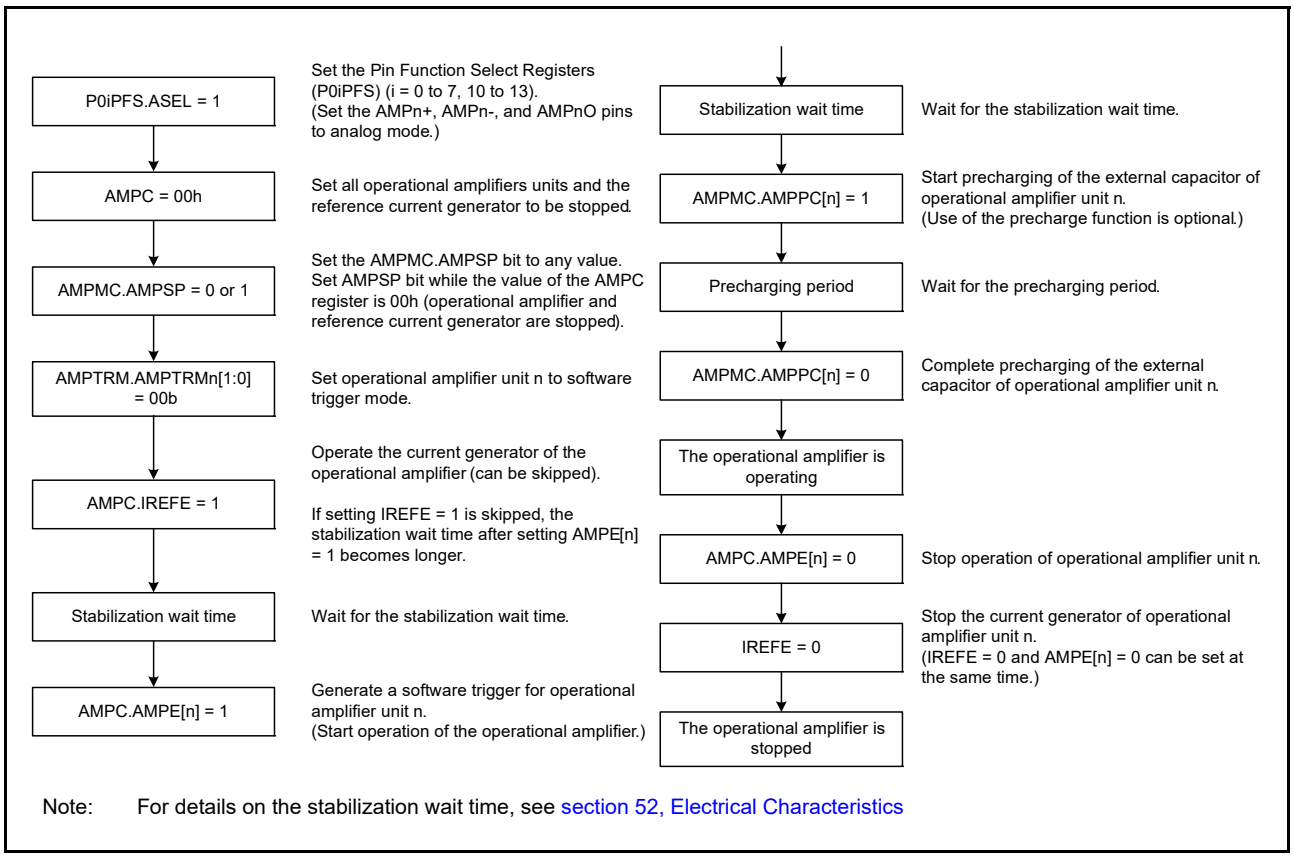
Figure 42.6 Operational amplifier control operation in activation and A/D trigger mode (2) when the reference current circuit and operational amplifier are stopped by setting the AMPC register to be activated by an activation trigger and stopped by an A/D conversion end (trigger)

Note 1. When operating/stopping the operational amplifier continuously, use the AMPE[n] bit again as in the first setting, and set the operational amplifier to wait for an activation trigger after it is stopped.

Note: n: Unit number (n = 0 to 3)
m: An activation trigger used to control operational amplifier unit n selected by the AMPTRS register
Set the AGT function. See section 42.4, Software Trigger Mode for the procedure to activate the operational amplifier with an activation trigger.

42.4 Software Trigger Mode

This section describes the procedure to activate and stop the operational amplifier using a software trigger. An example of each register setting is shown in Figure 42.7.



Note: For details on the stabilization wait time, see section 52, Electrical Characteristics

Figure 42.7 Procedure to start and stop OPAMP in software trigger mode

42.5 Activation Trigger Mode

This section describes the procedure to activate the operational amplifier using an activation trigger and to stop the amplifier with software. An example of each register setting is shown in Figure 42.8.

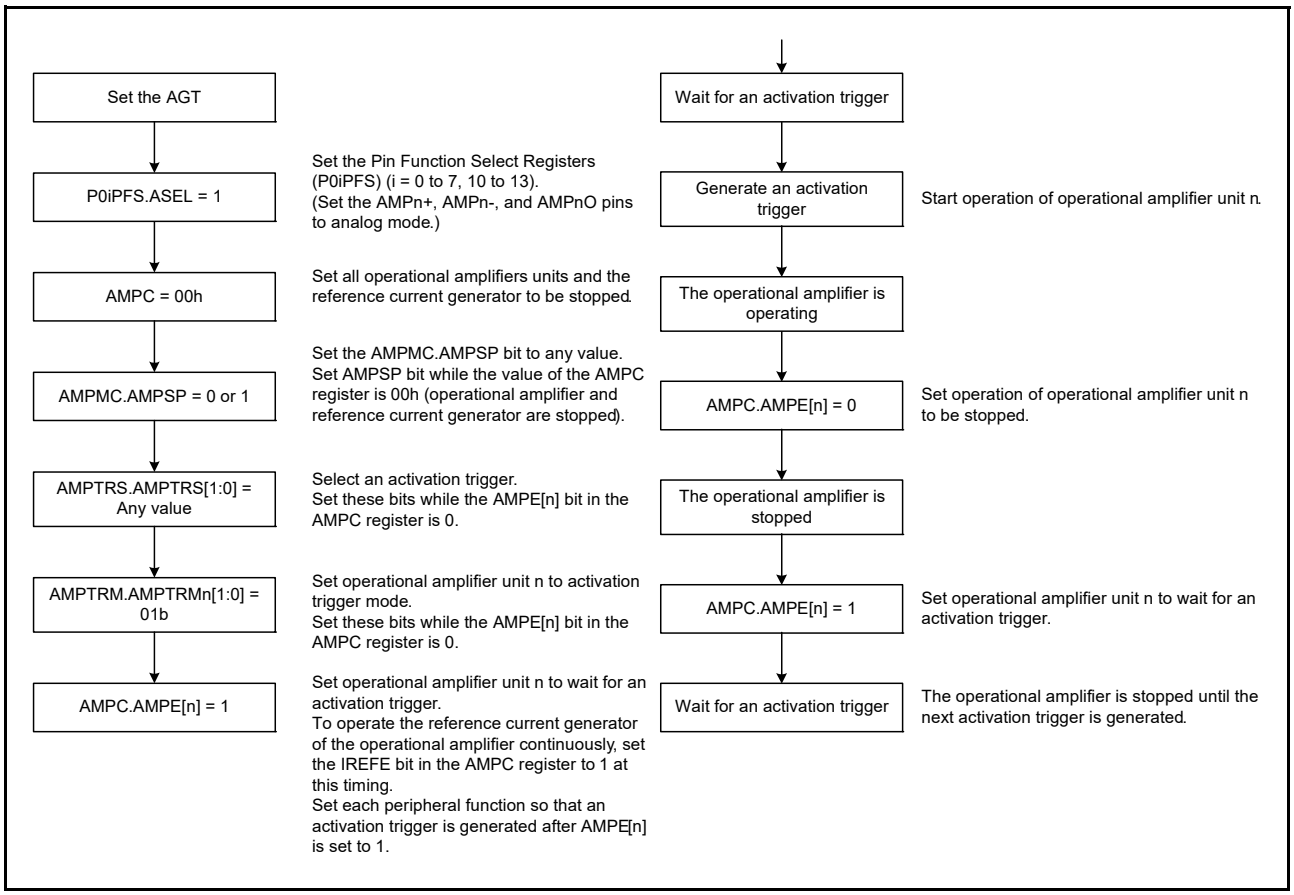


Figure 42.8 Procedure to start and stop OPAMP in activation trigger mode

42.6 Activation and A/D Trigger Mode

This section describes the procedure to activate the operational amplifier using an activation trigger and to stop the amplifier with an A/D conversion end trigger. An example of each register setting is shown in Figure 42.9.

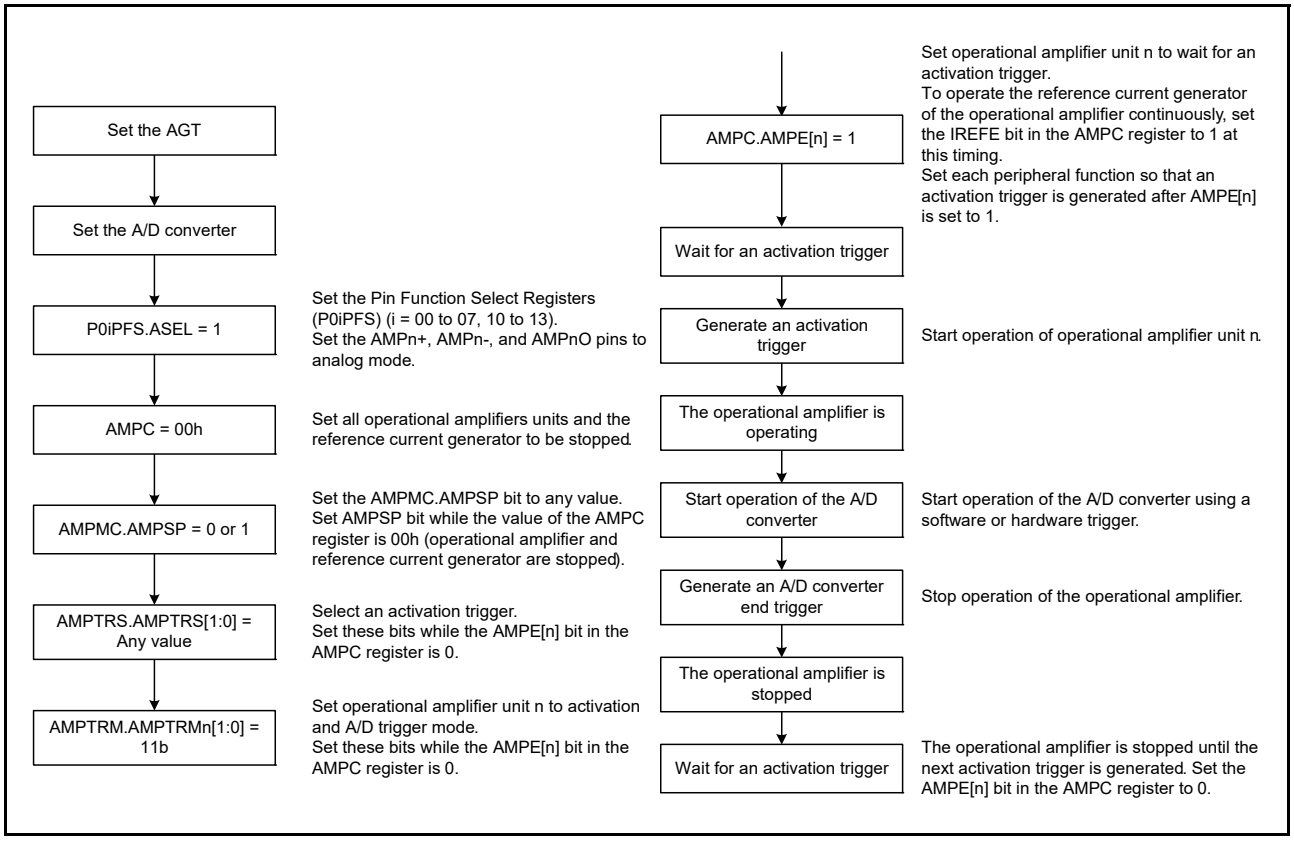


Figure 42.9 Procedure to activate the operational amplifier using an activation trigger and to stop the operational amplifier with an A/D conversion end trigger

42.7 Usage Notes

In addition to the AMPC register settings, the operational amplifier function can be activated by an activation trigger and stopped at the end of A/D conversion. The reference current circuit can be stopped at the end of A/D conversion. Application sequences must prevent these asynchronous triggers from causing conflicts between the activation and stop control.

Do not perform A/D conversion on pins that are used for the positive and negative input of the operational amplifier because these pins are multiplexed with analog input for the A/D converter.

43. High-Speed Analog Comparator (ACMPHS)

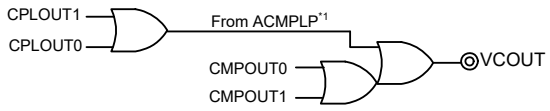
43.1 Overview

The High-Speed Analog Comparator (ACMPHS) can be used to compare a test voltage with a reference voltage and to provide a digital output based on the result of conversion. Both the test voltage and the reference voltage can be provided to the ACMPHS from internal sources (D/A converter output and internal reference voltage) and an external source. Such flexibility is useful in applications that require go/no-go comparisons to be performed between analog signals without necessarily requiring A/D conversion.

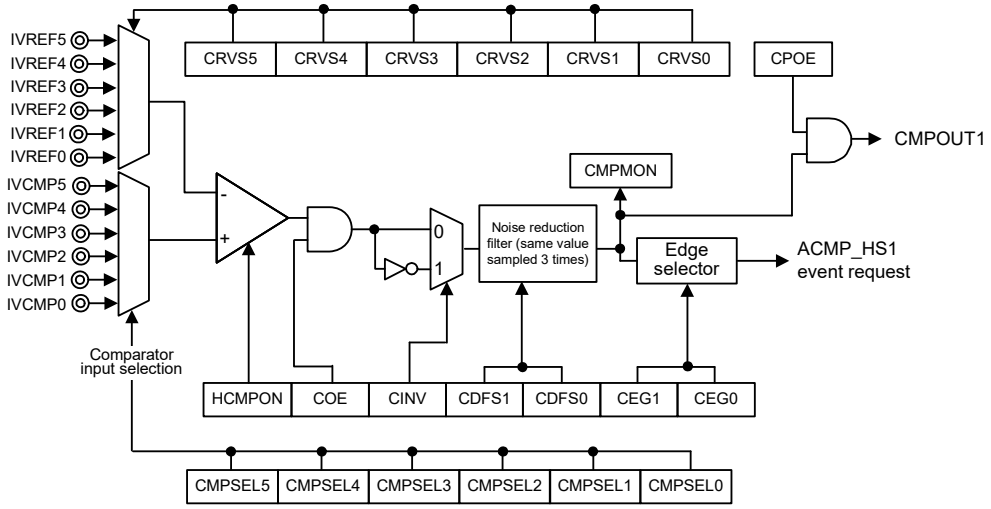
Table 43.1 lists the ACMPHS features and Figure 43.1 shows the block diagram. Table 43.2 shows the input source configuration.

Table 43.1 ACMPHS features

Parameter	Description
Number of channels	2 (ACMPHS0 and ACMPHS1)
Analog input voltage	• ACMPHS1 Output from internal D/A converter Input from internal A/D converter input pin (one selectable) • ACMPHS0 Output from internal D/A converter Internal reference voltage (Vref) Input from internal A/D converter input pin (one selectable).
Reference voltage	• ACMPHS1 Output from internal D/A converter Input from internal A/D converter input pin (one selectable) • ACMPHS0 Output from internal D/A converter Internal reference voltage (Vref) Input from internal A/D converter input pin (one selectable).
ACMPHS 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 three sampling frequencies • Not using the filter function is selectable.

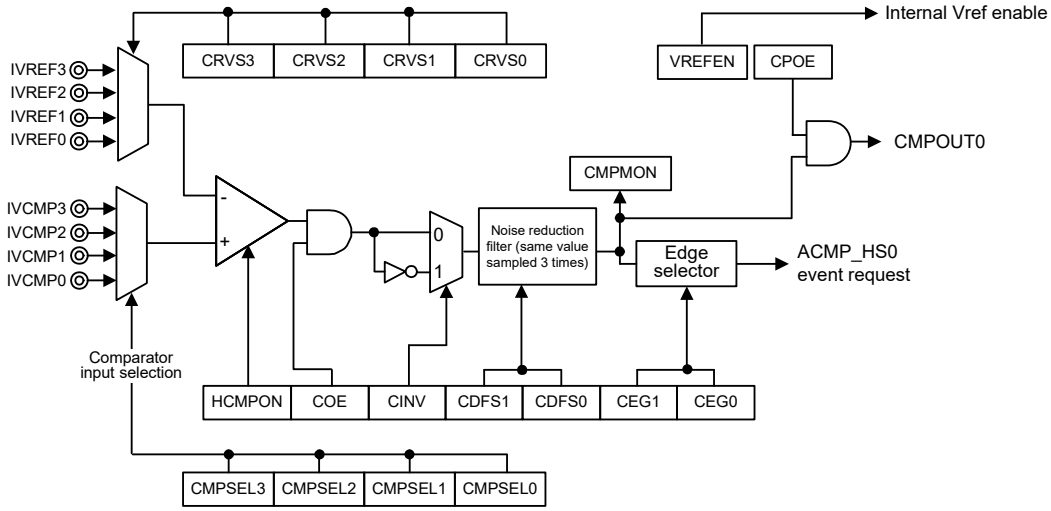


• ACMPHS1



CMPSEL5 to CMPSEL0 : Bits in CMPSEL0 register
CRVS5 to CRVS0 : Bits in CMPSEL1 register
CPOE : Bit in CPIOC register
CMPMON : Bit in CMPMON register
HCMPON, COE, CINV, CDFS1, CDFS0, CEG1, CEG0: Bits in CMPCTL

• ACMPHS0



CMPSEL3 to CMPSEL0 : Bits in CMPSEL0 register
CRVS3 to CRVS0 : Bits in CMPSEL1 register
CPOE, VREFEN : Bits in CPIOC register
CMPMON : Bit in CMPMON register
HCMPON, COE, CINV, CDFS1, CDFS0, CEG1, CEG0: Bits in CMPCTL register

Note: ACMPHS0, ACMPHS1, and ACMPLP results are bundled VCOUT.

Figure 43.1 ACMPHS block diagram

Table 43.2 ACMPHS input source configuration

Comparator	Reference voltage input source						Analog voltage input source						Output pin
	IVREF5	IVREF4	IVREF3	IVREF2	IVREF1	IVREF0	IVCMP5	IVCMP4	IVCMP3	IVCMP2	IVCMP1	IVCMP0	
ACMPHS0	not exist	not exist	Vref*2	DA0*1	AN006	AN005	not exist	not exist	Vref*2	DA1*3	AN007/ AMP3O	AN004/ AMP2O	VCOUT*4
ACMPHS1	DA0*1	AN006	AN003/ AMP1O	AN002/ AMP0O	AN001	AN000	DA1*3	AN007/ AMP3O	AN003/ AMP1O	AN002/ AMP0O	AN001	AN000	

Note 1. When D/A converter 0 output (DA0) is not used, the signal can be used as AN014 analog input.
Note 2. Internal voltage reference.
Note 3. When D/A converter 1 output (DA1) is not used, the signal can be used as AN015 analog input.
Note 4. ACMPHS0 and ACMPHS1 compare outputs are bundled with the VCOUT pin.

43.2 Register Descriptions

43.2.1 Comparator Control Register (CMPCTL)

Address(es): ACMPHS0.CMPCTL 4008 5000h, ACMPHS1.CMPCTL 4008 5C00h

	b7	b6	b5	b4	b3	b2	b1	b0
	HCMPON	CDFS[1:0]		CEG1	CEG0	—	COE	CINV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CINV	Comparator Output Polarity Selection*1, *2	0: Comparator output not inverted 1: Comparator output inverted.	R/W
b1	COE	Comparator Output Enable	0: Comparator output disabled (the output signal is low level) 1: Comparator output enabled.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4, b3	CEG1/ CEG0	Selection of Valid Edge (Edge Selector)	b4 b3 0 0: No edge selection 0 1: Rising edge selection 1 0: Falling edge selection 1 1: Both edge selection.	R/W
b6, b5	CDFS[1:0]	Noise Filter Selection*1, *2, *3	b6 b5 0 0: Noise filter not used 0 1: Noise filter sampling frequency is PCLKB/2^3 1 0: Noise filter sampling frequency is PCLKB/2^4 1 1: Noise filter sampling frequency is PCLKB/2^5.	R/W
b7	HCMPON	Comparator Operation Control*4	0: Operation stopped (the comparator outputs a low-level signal) 1: Operation enabled (input to the comparator pins is enabled).	R/W

Note 1. Change CDFS[1:0] and CINV bits only after disabling the ACMPHS output (COE = 0).
Note 2. If bits CDFS[1:0] and CINV are changed, an ACMPHS interrupt request and an ELC event may be generated. Change these bits only after setting the ELSRn register to 0 (the ACMPHS output is not linked). After changing these bits, initialize the IR flag bit in the IELSRn register to 0 (interrupt status flag cleared).
Note 3. If bits CDFS[1:0] are changed from 00b (noise filter not used) to a value other than 00b (noise filter used), perform sampling four times and update the filter output, and then use the ACMPHS interrupt request or the ELC event.
Note 4. The operation stabilization wait time is required for the ACMPHS operation after enabling ACMPHS operation (HCMPON = 1).
Operation stabilization wait time for ACMPHS0 and ACMPHS1: 1 µs.

The CMPCTL register controls the ACMPHS operation, enables or disables the ACMPHS output, selects the noise filter, selects the valid edge of the interrupt signal, and selects the interrupt. A reset clears this register to 00h.

43.2.2 Comparator Input Select Register (CMPSEL0)

Address(es): ACMPHS0.CMPSEL0 4008 5004h, ACMPHS1.CMPSEL0 4008 5C04h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CMPSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	CMPSEL[5:0]	Comparator Input Selection*1	b5 b0 0 0 0 0 0 0: No input 0 0 0 0 0 1: IVCMP0 selected*2 0 0 0 0 1 0: IVCMP1 selected*2 0 0 0 1 0 0: IVCMP2 selected*2 0 0 1 0 0 0: IVCMP3 selected*2 0 1 0 0 0 0: IVCMP4 selected*2 1 0 0 0 0 0: IVCMP5 selected.*2 Other settings are prohibited.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Use the following procedure to change the CMPSEL[5:0] bits. Writing a value other than 0000 0000b while the value of CMPSEL0 register is not 0000 0000b is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.
To change the CMPSEL[5:0] bits:
1. Set the CMPCTL.COE bit to 0.
2. Set the CMPSEL0 register to 0000 0000b.
3. Set a new value in the CMPSEL[5:0] bits, with 1 set in only one of the bits.
4. Wait for the input switching stabilization wait time (200 ns).
5. Set the CMPCTL.COE bit to 1.
6. Clear the IR flag bit in the IELSRn register to clear the interrupt status.

Note 2. For details, see Table 43.2.

43.2.3 Comparator Reference Voltage Select Register (CMPSEL1)

Address(es): ACMPHS0.CMPSEL1 4008 5008h, ACMPHS1.CMPSEL1 4008 5C08h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CRVS[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	CRVS[5:0]	Reference Voltage Selection*1	b5 b0 0 0 0 0 0 0: No reference voltage 0 0 0 0 0 1: IVREF0 selected*2 0 0 0 0 1 0: IVREF1 selected*2 0 0 0 1 0 0: IVREF2 selected*2 0 0 1 0 0 0: IVREF3 selected*2 0 1 0 0 0 0: IVREF4 selected*2 1 0 0 0 0 0: IVREF5 selected.*2 Other settings are prohibited.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Use the following procedure to change the CRVS[5:0] bits. Writing a value other than 0000 0000b while the value of CMPSEL1 register is not 0000 0000b is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.
To change the CRVS[5:0] bits:
1. Set the CMPCTL.COE bit to 0.
2. Set CMPSEL1 register to 0000 0000b.
3. Set a new value to the CRVS[5:0] bits, with 1 set in only one of the bits.
4. Wait for the input switching stabilization wait time (200 ns).
5. Set the CMPCTL.COE bit to 1.
6. Clear IR flag bit in the IELSRn register to clear the interrupt status.

Note 2. For details, see Table 43.2.

43.2.4 Comparator Output Monitor Register (CMPMON)

Address(es): ACMPHS0.CMPMON 4008 500Ch, ACMPHS1.CMPMON 4008 5C0Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPMON
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CMPMON	Comparator Output Monitor*1	0: Comparator output low 1: Comparator output high.	R
b7 to b1	—	Reserved	These bits are read as 0	R

Note 1. When ACMPHS operation is enabled (HCMPON = COE = 1) but the noise filter is not in use (CDFS[1:0] = 00b), write the software so that the CMPMON bit is read twice and the values are only used after the two consecutive values match.

43.2.5 Comparator Output Control Register (CPIOC)

Address(es): ACMPHS0.CPIOC 4008 5010h, ACMPHS1.CPIOC 4008 5C10h

	b7	b6	b5	b4	b3	b2	b1	b0
	VREFEN	—	—	—	—	—	—	CPOE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CPOE	Comparator Output Selection	0: VCOUT pin output of the comparator disabled (the output signal is low level) 1: VCOUT pin output of the comparator enabled.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	VREFEN	Internal Vref Enable*1	0: Internal Vref disabled 1: Internal Vref enabled.	R/W

Note 1. VREFEN exists only in ACMPHS0.CPIOC. When using the internal Vref in ACMPHS0 or ACMPHS1, set the VREFEN bit in ACMPHS0.CPIOC to 1. Bit [7] in ACMPHS1.CPIOC should be 0 regardless of whether or not the internal Vref is used.

43.3 Operation

The ACMPHS compares a reference voltage to an analog input voltage. Operation is not guaranteed when the values of registers are changed during ACMPHS operation. Table 43.3 shows the procedures for setting the registers associated with ACMPHS associated.

Table 43.3 Procedure for setting registers associated with ACMPHS

Step	Register	Bit	Setting
1	Associated MSTPCRD register	MSTPD28, MSTPD30	0: Enable input clock supply
2	Associated pin function control register (PFS)	ASEL	1: Select the function of pins IVREF and IVCMP
3	ACMPHS0.CPIOC	VREFEN	1: When using Internal Vref
4	Associated D/A converter		When using D/A converter, configure the corresponding D/A converter channel as needed
5	ACMPHS0.CMPSEL0 ACMPHS0.CMPSEL1 ACMPHS1.CMPSEL0 ACMPHS1.CMPSEL1	CMPSEL0 to CMPSEL3 CRVS0 to CRVS3 CMPSEL0 to CMPSEL5 CRVS0 to CRVS5	Select the ACMPHSn input, with 1 set in only one of the bits
6	CMPCTL	CDFS[1:0], CEG1, CEG0, and CINV	Set the ACMPHSn control
		HCMPON	1: Enable the ACMPHSn operation
7	Waiting for the ACMPHS stabilization time (min. 1 µs).		
8	CMPCTL	COE	1: Enable the ACMPHSn output
9	CPIOC	CPOE	1: Set the VCOUT output
	Associated pin function control register (PFS)	PSEL, PMR	Select the VCOUT port function
10	IELSRn	IR, IELS[8:0]	When using an interrupt, select the interrupt status flag and ICU event link select*1
11	ELSRn	ELS[8:0]	When using an ELC, select the event link select*2
12	Operation started		
13	CMPCTL	COE	0: When changing IVREF or IVCMP (disable ACMPHSn output)
14	ACMPHS0.CMPSEL1 ACMPHS1.CMPSEL1	CRVS0 to CRVS3, CRVS0 to CRVS5	Use the following procedure to modify CMPSEL1 bits: Set bits CMPSEL1 to 0000 0000b. Set a new value to bits CMPSEL1, with 1 set in only one of the bits.
15	ACMPHS0.CMPSEL0 ACMPHS1.CMPSEL0	CMPSEL0 to CMPSEL3, CMPSEL0 to CMPSEL5	Use the following procedure to modify CMPSEL0 bits: Set bits CMPSEL0 to 0000 0000b. Set a new value to bits CMPSEL0, with 1 set in only one of the bits.
16	Waiting for the ACMPHSn switching stabilization time (min. 200 ns)		
17	CMPCTL	COE	1: Enable the ACMPHSn output
18	Operation restarted		

Note 1. After ACMPHSn is set, initialize the interrupt flag to prevent an interrupt from occurring until the operation becomes stable.

Note 2. After ACMPHSn is set, initialize the event link select to prevent an interrupt from occurring until the operation becomes stable.

Figure 43.2 shows an example of ACMPHS operation. The VCOUT output becomes 1 when the analog input voltage is higher than the ACMPHS reference input voltage, and the VCOUT output becomes 0 when the analog input voltage is lower than the reference voltage. When the ACMPHS output changes, an interrupt request and an ELC event are output.

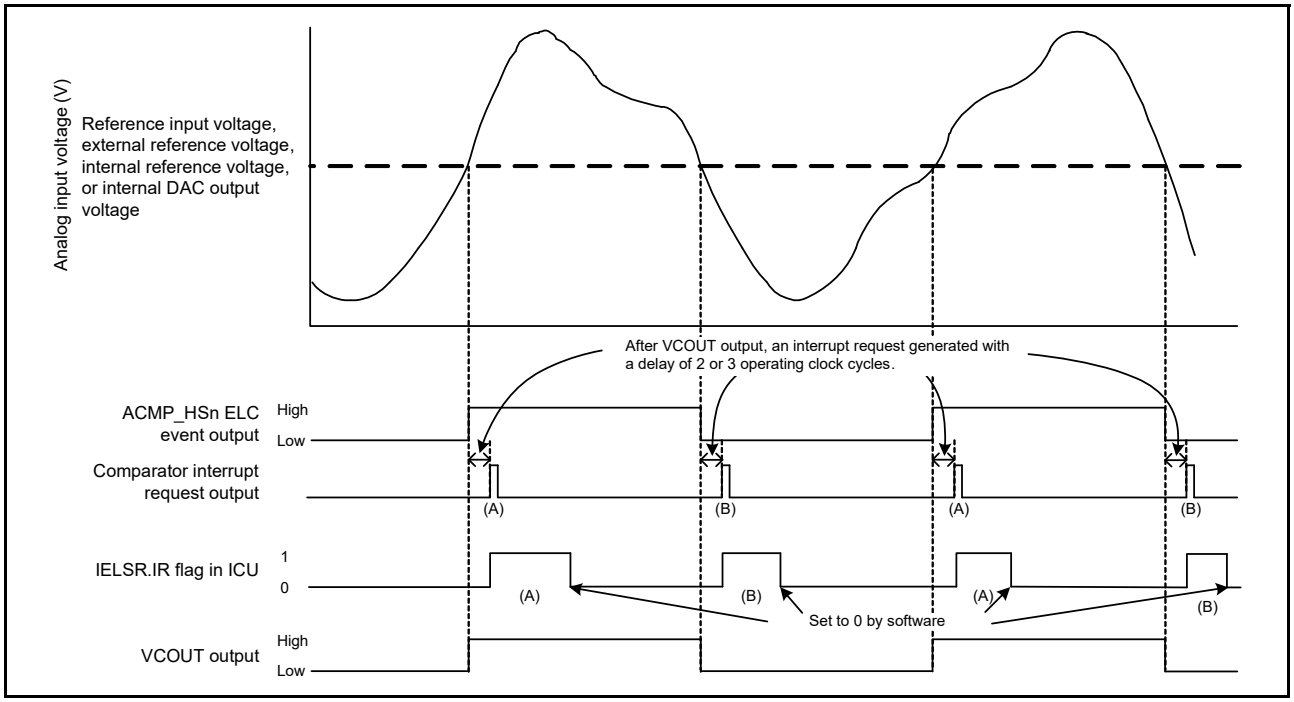


Figure 43.2 ACMPHS operation example

Figure 43.2 applies when CPOE = 1 (pin output enabled), CDFS[1:0] = 00b (filter not used), and CEG1 = CEG0 = 1 (both-edge selection). When CINV = 0, CEG0 = 1, and CEG1 = 0 (rising-edge selection for non-inversion output signal from the ACMPHS), IELSR.IR flag changes as shown by (A) only. When CINV = 0, CEG0 = 0, and CEG1 = 1 (falling-edge selection for non-inversion output signal from the ACMPHS), IR flag changes as shown by (B) only. When CPOE = 1, VCOUT directly outputs the ELC event output.

43.4 Noise Filter

The ACMPHS contains a noise filter. The sampling clock can be selected in the CMPCTL.CDFS[1:0] bits. The ACMPHS signal is sampled every sampling clock, and if the same value is sampled three times, the noise filter output at the next sampling clock cycle is used as the ACMPHS output.

Figure 43.3 shows the configuration of the noise filter and edge detector and Figure 43.4 shows an example of noise filter and interrupt operation. When using an interrupt and ELC is in Software Standby mode, set the CMPCTL.CDFS[1:0] bits to 00b (noise filter not used).

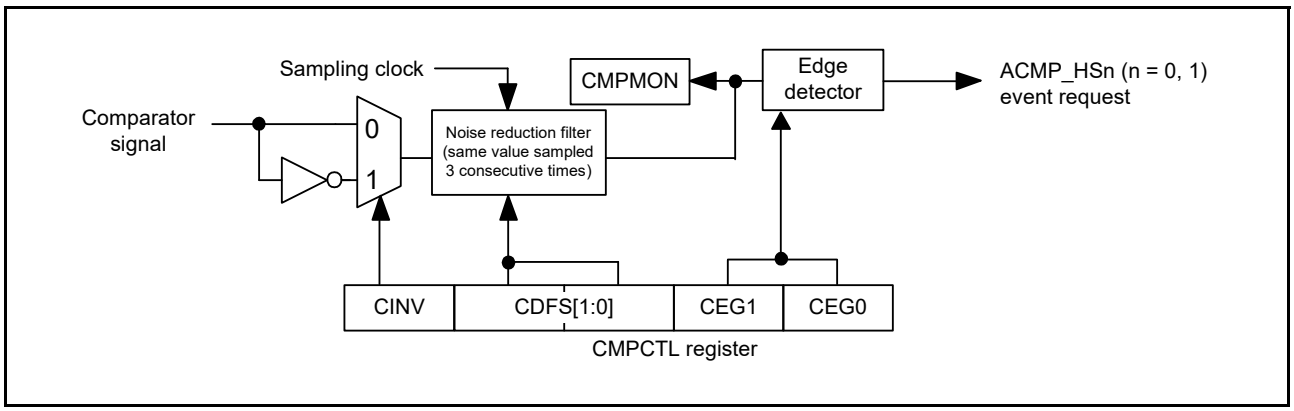


Figure 43.3 Noise filter and edge detection configuration

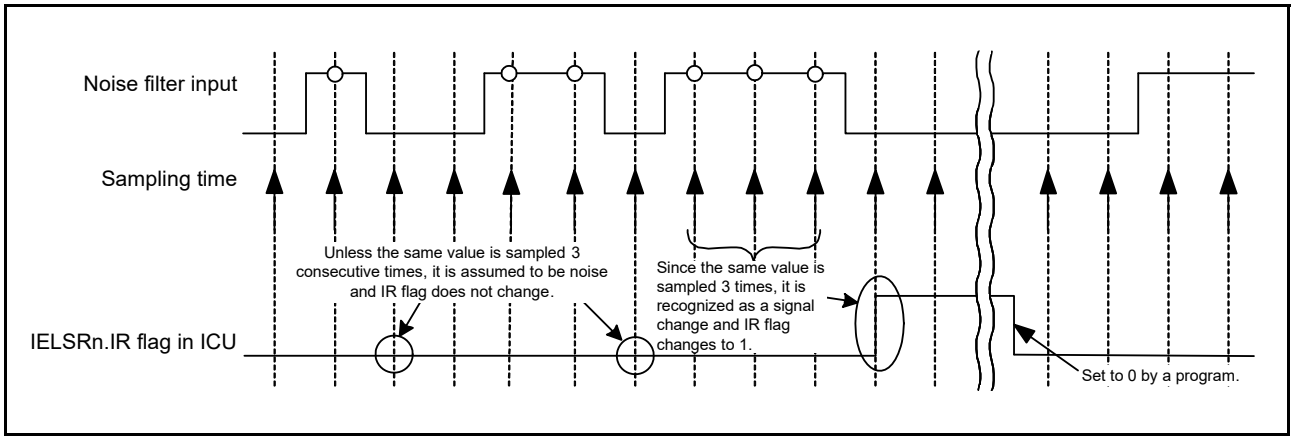


Figure 43.4 Noise filter and interrupt operation example

The operation example in Figure 43.4 applies when the CMPCTL.CDFS[1:0] bits are 01b, 10b, or 11b (noise filter used).

43.5 ACMPHS Interrupts

ACMPHS generates two interrupt requests from sources, ACMPHS0 and ACMPHS1. To use an ACMPHS interrupt, select it in the IELSR register in the Interrupt Controller Unit (ICU). Set at least one of the bits CMPCTL.CEG0 or CMPCTL.CEG1 to 1 (a value other than 00b (no edge selection)).

For details on the register setting related to ACMPHS interrupt request, refer to section 43.2.1, Comparator Control Register (CMPCTL).

43.6 ACMPHS Output to the Event Link Controller (ELC)

The ELC uses the ACMPHS interrupt request signal as an ELC event signal, enabling link operation for the preset module. To use the ACMPHS ELC event, select them in the ELSR register in the ELC. When using the ELC event request, set at least one of the CMPCTL.CEG0 bit and CMPCTL.CEG1 bit to 1 (to a value other than 00b for no edge selection).

43.7 ACMPHS Pin Output

The comparison result from the ACMPHS can be output to external pins. Use the CMPCTL.CINV and CPIOC.CPOE bits to set the output polarity (non-inverted output or inverted output) and enable/disable output. To output the ACMPHS comparison result to the VCOUT output pin, set the associated port mn pin function control register (PmnPFS) in the I/O register.

43.8 Usage Notes

43.8.1 Settings for the Module-Stop Function

ACMPHS operation can be disabled or enabled using the Module Stop Control Register. The ACMPHS is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

43.8.2 Relationship with 14-bit A/D converter

Constraints apply on the simultaneous use of ACMPHS analog input and ADC14 analog input. For details, see section 39.8.13, Relationship between A/D converter, OPAMP, ACMPHS, and ACMPLP.

44. Low Power Analog Comparator (ACMPLP)

The Low Power Analog Comparator (ACMPLP) compares a reference input voltage and an analog input voltage. Comparator channels ACMPLP0 and ACMPLP1 are independent of each other.

44.1 Overview

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 either an input to the CMPREFi (i = 0, 1) pin or from the internal reference voltage (Vref) generated internally in the MCU.

The ACMPLP response speed can be set before starting an operation. Setting high-speed mode decreases the response delay time, but increases current consumption. Setting low-speed mode increases the response delay time, but decreases current consumption.

Table 44.1 lists the features of the ACMPLP, Figure 44.1 shows a block diagram of the ACMPLP when the window function is disabled. Figure 44.2 shows the block diagram of the ACMPLP when the window function is enabled. Table 44.2 lists the I/O pins of the ACMPLP.

Table 44.1 ACMPLP features

Parameter	Description
Number of channels	2 (ACMPLP0 and ACMPLP1)
Analog input voltage	Input from CMPINi (i = 0, 1) pin
Reference voltage	• Internal reference voltage (Vref) • Input from CMPREFi (i = 0, 1) pin.
Comparator output	• Comparison result • Generation of ELC event output • Monitor output from register.
Interrupt request signal	• Interrupt request generated on detecting a valid edge from comparison result • Rising edge, falling edge, or both edges can be selected.

Selectable functions	• Noise filter function
	  • One of three sampling frequencies can be selected or noise filter function can by bypassed.
	• Window function
	  • Window function can be enabled or disabled.
	• Response speed
	  • High-speed mode or low-speed mode can be selected.

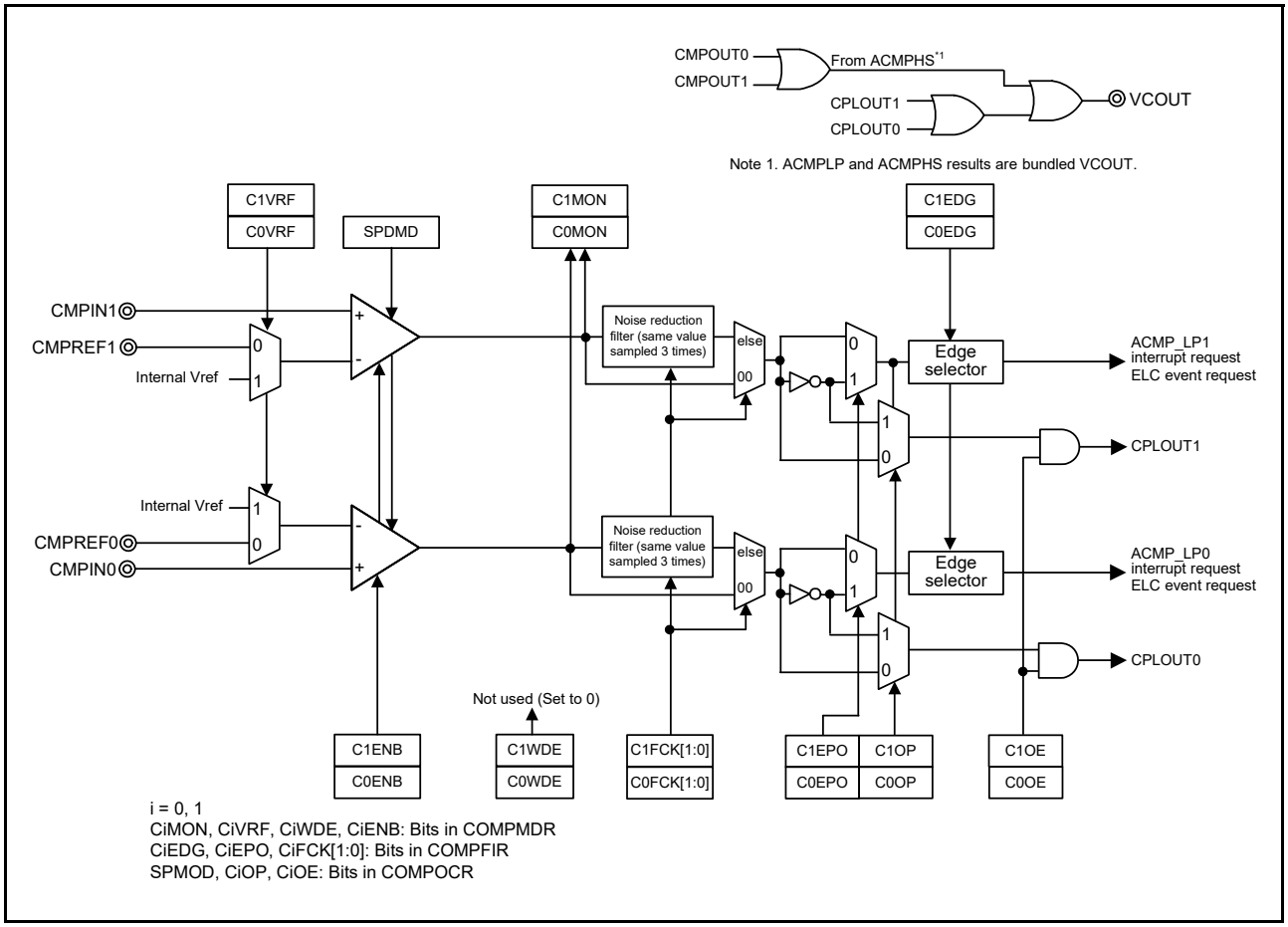


Figure 44.1 ACMPLP block diagram when window function is disabled

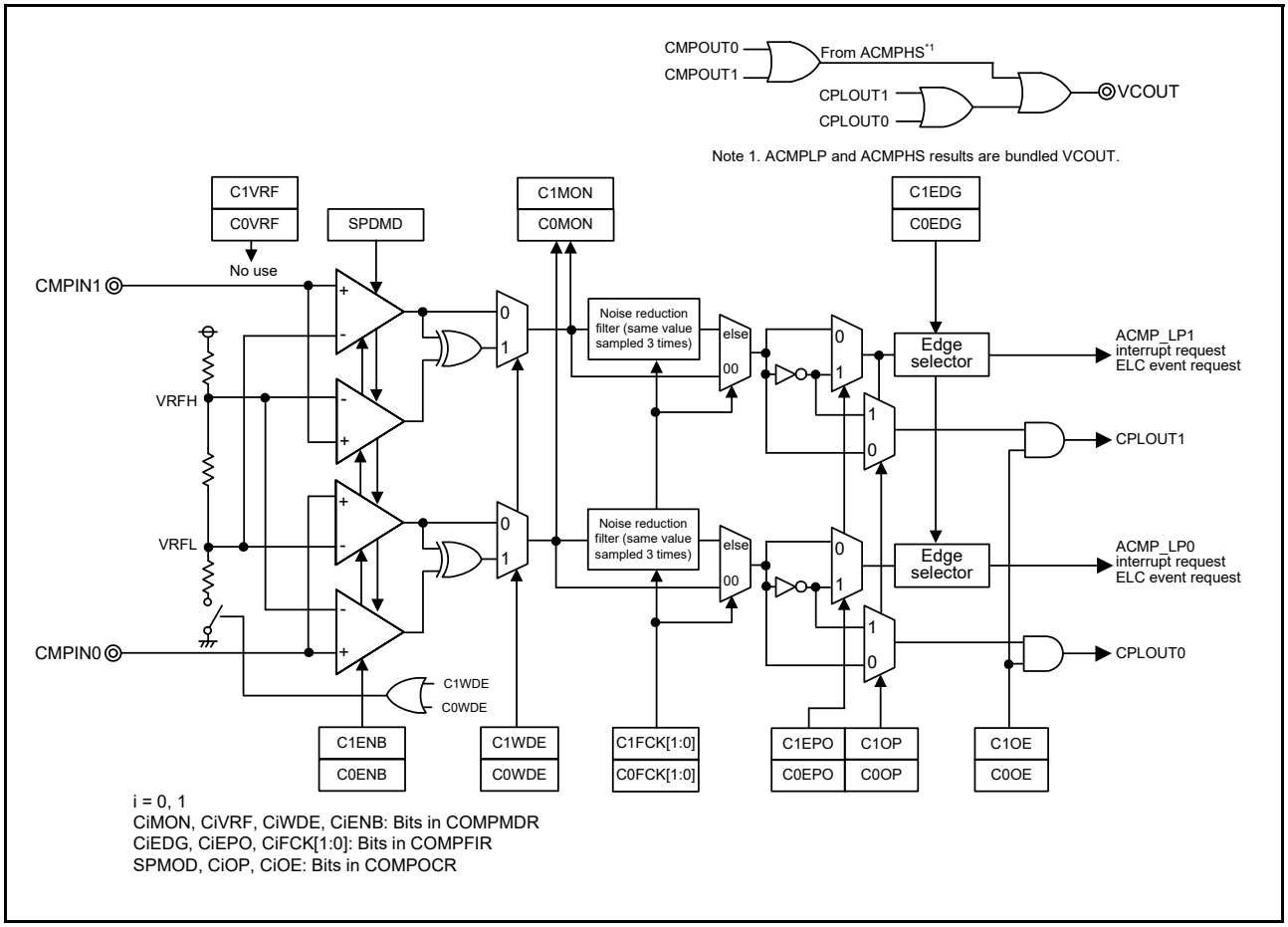


Figure 44.2 ACMPLP block diagram when window function is enabled

Table 44.2 Comparator pin configuration

Comparator	Reference voltage input pin	Analog voltage input pin	Output pin
ACMPLP0	CMPREF0	CMPIN0	VCOUT*1
ACMPLP1	CMPREF1	CMPIN1	

Note 1. ACMPLP0 and ACMPLP1 compare outputs are bundled on the VCOUT pin.

44.2 Register Descriptions

44.2.1 ACMPLP Mode Setting Register (COMPMDR)

Address(es): ACMPLP.COMPMDR 4008 5E00h

	b7	b6	b5	b4	b3	b2	b1	b0
	C1MON	C1VRF	C1WDE	C1ENB	C0MON	C0VRF	C0WDE	C0ENB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	C0ENB	ACMPLP0 Operation Enable	0: Disable comparator channel ACMPLP0 1: Enable comparator channel ACMPLP0.	R/W
b1	C0WDE	ACMPLP0 Window Function Mode Enable*1,*2	0: Disable window function for ACMPLP0 1: Enable window function for ACMPLP0.	R/W
b2	C0VRF	ACMPLP0 Reference Voltage Selection	0: Select CMPREF0 input as ACMPLP0 reference voltage 1: Select internal reference voltage (Vref) as ACMPLP0 reference voltage.	R/W
b3	C0MON	ACMPLP0 Monitor Flag*3	When the window function is disabled: 0: CMPIN0 < CMPREF0, CMPIN0 < internal reference voltage, or ACMPLP0 operation disabled 1: CMPIN0 > CMPREF0, or CMPIN0 > internal reference voltage. When the window function is enabled: 0: CMPIN0 < VRFL, CMPIN0 > VRFH, or ACMPLP0 operation disabled 1: VRFL < CMPIN0 < VRFH.	R
b4	C1ENB	ACMPLP1 Operation Enable	0: Disable comparator channel ACMPLP1 1: Enable comparator channel ACMPLP1.	R/W
b5	C1WDE	ACMPLP1 Window Function Mode Enable*1,*2	0: Disable window function for ACMPLP1 1: Enable window function for ACMPLP1.	R/W
b6	C1VRF	ACMPLP1 Reference Voltage Selection	0: Select CMPREF1 input as ACMPLP1 reference voltage 1: Select internal reference voltage (Vref) as ACMPLP1 reference voltage.	R/W
b7	C1MON	ACMPLP1 Monitor Flag*3	When the window function is disabled: 0: CMPIN1 < CMPREF1, CMPIN1 < internal reference voltage, or ACMPLP1 operation disabled 1: CMPIN1 > CMPREF1, or CMPIN1 > internal reference voltage. When the window function is enabled: 0: CMPIN1 < VRFL, CMPIN1 > VRFH, or ACMPLP1 operation disabled 1: VRFL < CMPIN1 < VRFH.	R

Note 1. Window function mode cannot be set when low-speed mode is selected (the SPDMD bit in the COMPOCR register is 0).
Note 2. In window function mode, the reference voltage in the comparator is selected regardless of the setting of this bit.
Note 3. The initial value is 0 immediately after a reset is released. However, the value is undefined when C0ENB is set to 0 and C1ENB is set to 0 after operation of the comparator is enabled once.

44.2.2 ACMPLP Filter Control Register (COMPFIR)

Address(es): ACMPLP.COMPFIR 4008 5E01h

	b7	b6	b5	b4	b3	b2	b1	b0
	C1EDG	C1EPO	C1FCK[1:0]		C0EDG	C0EPO	C0FCK[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	C0FCK[1:0]	ACMPLP0 Filter Select*1	b1 b0 0 0: No Sampling (bypass) 0 1: Sampling at PCLKB 1 0: Sampling at PCLKB/8 1 1: Sampling at PCLKB/32.	R/W
b2	C0EPO	ACMPLP0 Edge Polarity Switching*1	0: Interrupt and ELC event request at rising edge 1: Interrupt and ELC event request at falling edge.	R/W
b3	C0EDG	ACMPLP0 Edge Detection Selection*1	0: Interrupt and ELC event request by one-edge detection 1: Interrupt and ELC event request by both-edge detection.	R/W
b5, b4	C1FCK[1:0]	ACMPLP1 Filter Select*1	b5 b4 0 0: No Sampling (bypass) 0 1: Sampling at PCLKB 1 0: Sampling at PCLKB/8 1 1: Sampling at PCLKB/32.	R/W
b6	C1EPO	ACMPLP1 Edge Polarity Switching*1	0: Interrupt and ELC event request at rising edge 1: Interrupt and ELC event request at falling edge.	R/W
b7	C1EDG	ACMPLP1 Edge Detection Selection*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 might be generated. Change these bits only after setting event link deselected. Also, be sure to clear the associated interrupt request flag.

44.2.3 ACMPLP Output Control Register (COMPOCR)

Address(es): ACMPLP.COMPOCR 4008 5E02h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPDMD	C1OP	C1OE	—	—	C0OP	C0OE	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R
b1	C0OE	ACMPLP0 VCOUT Pin Output Enable*1	0: Disabled 1: Enabled.	R/W
b2	C0OP	ACMPLP0 VCOUT Output Polarity Selection*1	0: Non-inverted 1: Inverted.	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b5	C1OE	ACMPLP1 VCOUT Pin Output Enable*1	0: Disabled 1: Enabled.	R/W
b6	C1OP	ACMPLP1 VCOUT Output Polarity Selection*1	0: Non-inverted 1: Inverted.	R/W
b7	SPDMD	ACMPLP0/ACMPLP1 Speed Selection*2	0: Select comparator low-speed mode 1: Select comparator high-speed mode.	R/W

Note 1. ACMPLP0 and ACMPLP1 result outputs are bundled on the VCOUT pin.
Note 2. Set the CiENB bit (i = 0, 1) in the COMPMDR register to 0 before rewriting the SPDMD bit.

44.3 Operation

ACMPLP0 and ACMPLP1 operate independently, and their operations are the same. Operation is not guaranteed when the values of their associated registers are changed during comparator operation. Table 44.3 shows the procedure for setting the ACMPLP associated registers.

Table 44.3 Procedure for setting the ACMPLP associated registers (i = 0, 1)

Step No.	Register	Bit	Setting		
1	MSTPCRD	MSTPD29	0: Input clock supply		
2	Corresponding Port mn Pin Function Select Register (PmnPFS)	ASEL	1: Select the function of pins CMPREFi and CMPINi		
3	COMPOCR	SPDMD	Select the comparator response speed (0: Low-speed mode / 1: High-speed mode)*1		
4	COMPMDR	CiWDE	0: Disable window function mode		1: Enable window function mode*2
		CiVRF	0: Reference = CMPREFi input	1: Reference = Internal reference voltage	Window comparator operation (reference = VRFL and VRFH*3)
		CiENB	1: Operation enabled		
5	Waiting for the comparator stabilization time T_{cmp} (min. 100 µs).				
6	COMPFIR	CiFCK[1:0]	Select whether the digital filter is 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)		
7	COMPOCR	CiOP, CiOE	Set the VCOUT output (select the polarity and set output enabled or disabled)		
	Corresponding Port mn Pin Function Select Register (PmnPFS)	PSEL, PMR	Select the VCOUT port function		
8	IELSRn	IR, IELS[8:0]	When using an interrupt: select the interrupt status flag, ICU event link select*3		
9	ELSRn	ELS[8:0]	When using an ELC: Select the Event Link Select*4		
10	Operation started				

Note 1. ACMPLP0 and ACMPLP1 cannot be set independently.
Note 2. Can only be set in high-speed mode (SPDMD = 1).
Note 3. After the setting of the comparator, a spurious interrupt might occur until operation becomes stable, so initialize the interrupt flag.
Note 4. After the setting of the comparator, a spurious interrupt might occur until operation becomes stable, so initialize the event link select.

Figure 44.3 shows an operating example of the ACMPLPi (i = 0, 1) when the window function is disabled. The reference input voltage (CMPREFi) or internal reference voltage (Vref) and the analog input voltage (CMPINi) are compared. If the analog input voltage is higher than the reference input voltage, the COMPMDR.CiMON bit is set to 1. If the analog input voltage is lower than the reference input voltage, the CiMON bit is set to 0.

ACMPLPi outputs an interrupt to ICU. For details on the interrupt, see section 44.5, ACMPLP Interrupts. ACMPLPi also outputs an event signal to the ELC to activate other modules. For details on the ELC, see section 44.6, ELC Event Output. Do not change the values of the registers during comparison.

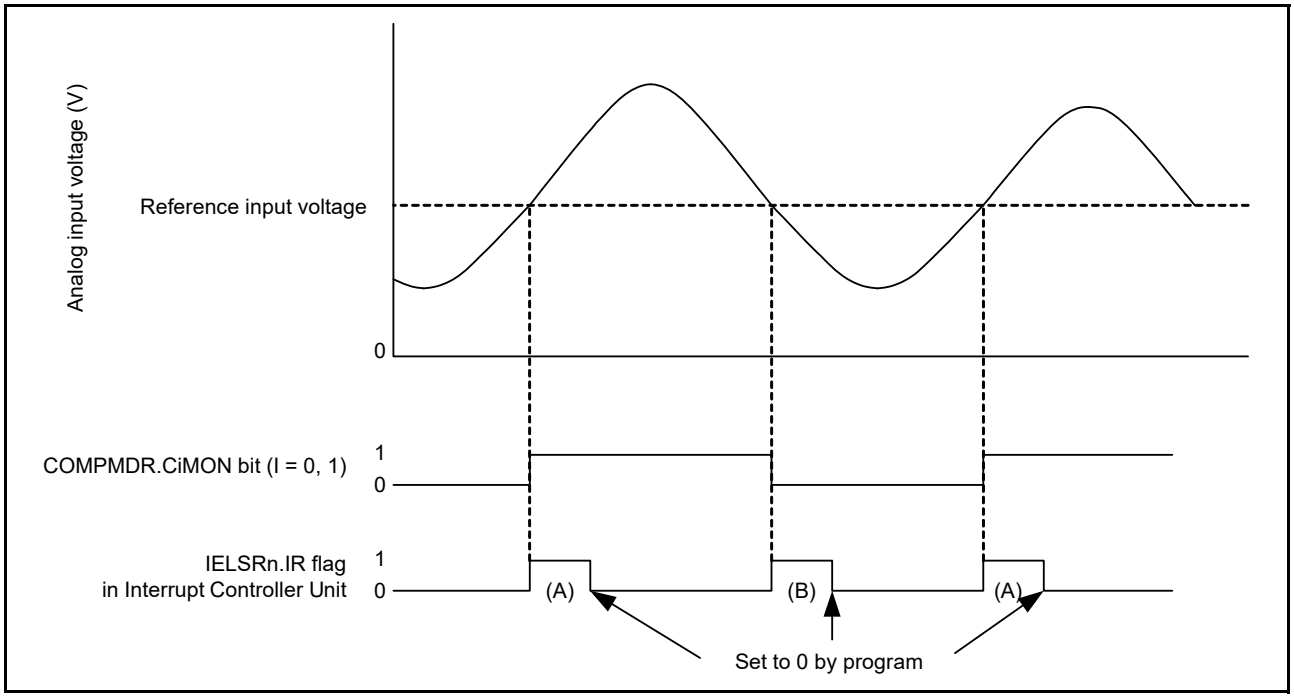


Figure 44.3 Operating example of ACMPLPi (i = 0, 1) when window function is disabled

Figure 44.3 applies when the following conditions are met:

- 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.

Figure 44.4 shows an operation example of ACMPLPi (i = 0, 1) when the window function is enabled. The internal Vref (VRFL/VRFH) and the analog input voltage are compared. The CiMON bit is set to 1 when VRFL < the analog input voltage < VRFH, and the CiMON bit is set to 0 when the analog input voltage < VRFL, or VRFH < the analog input voltage.

ACMPLPi outputs an interrupt to ICU. For details on the interrupt, see section 44.5, ACMPLP Interrupts. ACMPLPi also outputs an event signal to the ELC to activate other modules. For details on the ELC, see section 44.6, ELC Event Output. Do not change the values of the registers during comparison.

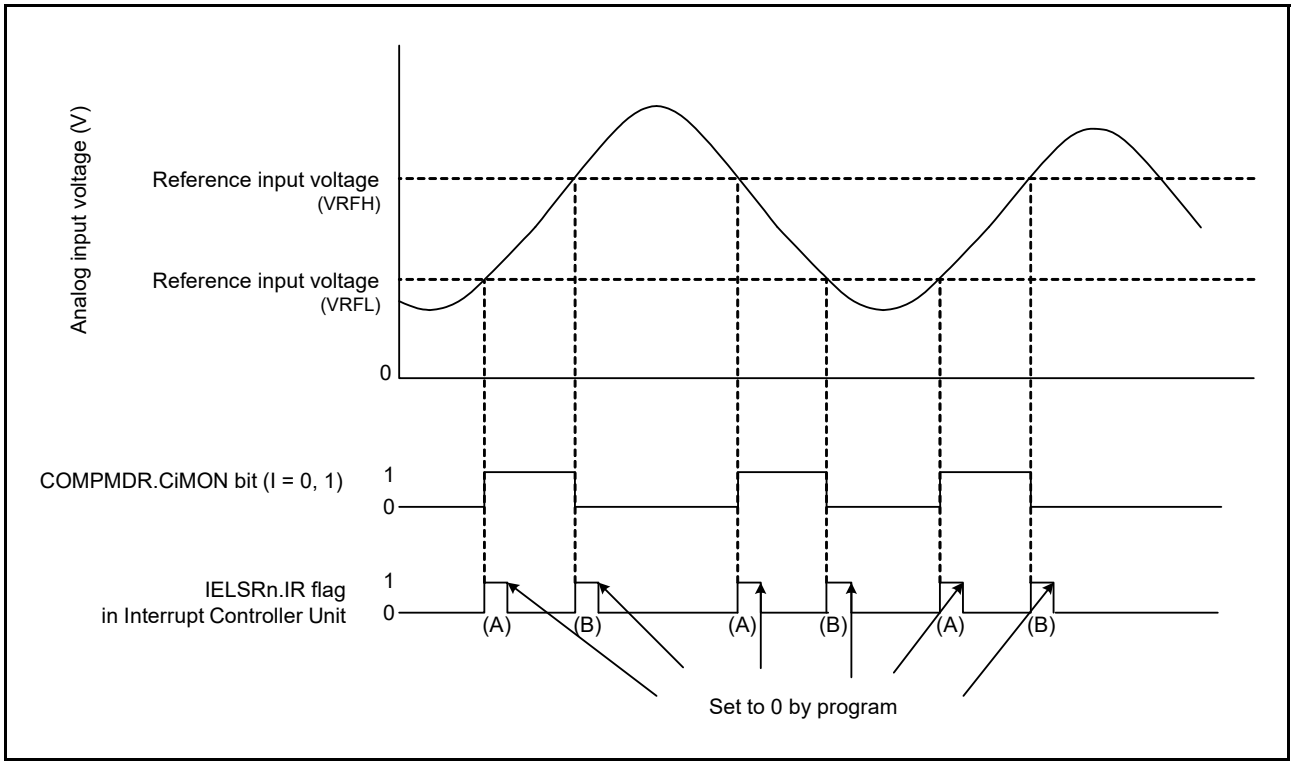


Figure 44.4 Operating example of ACMPLPi (i = 0, 1) when window function is enabled

Figure 44.4 applies when the following conditions are met:

- 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.

44.4 Noise Filter

Figure 44.5 shows the configuration of the ACMPLPi noise filter, and Figure 44.6 shows an operating example of the ACMPLPi noise filter.

The sampling clock can be selected in the COMPFIR.CiFCK[1:0] bits. The ACMP_LPi signal (internal signal) output from ACMPLPi 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 and ELC in Software Standby mode, set the COMPFIR.CiFCK[1:0] bits to 00b (bypass).

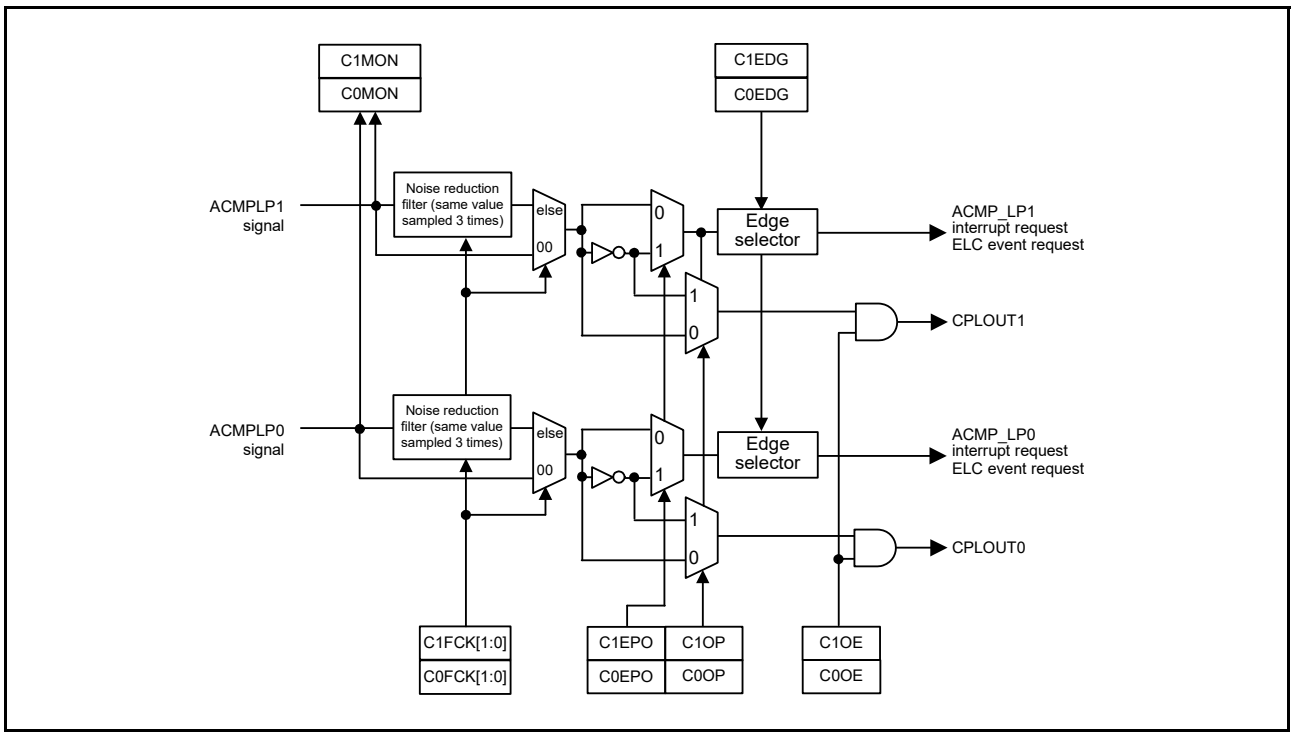


Figure 44.5 Noise filter and edge detection configuration

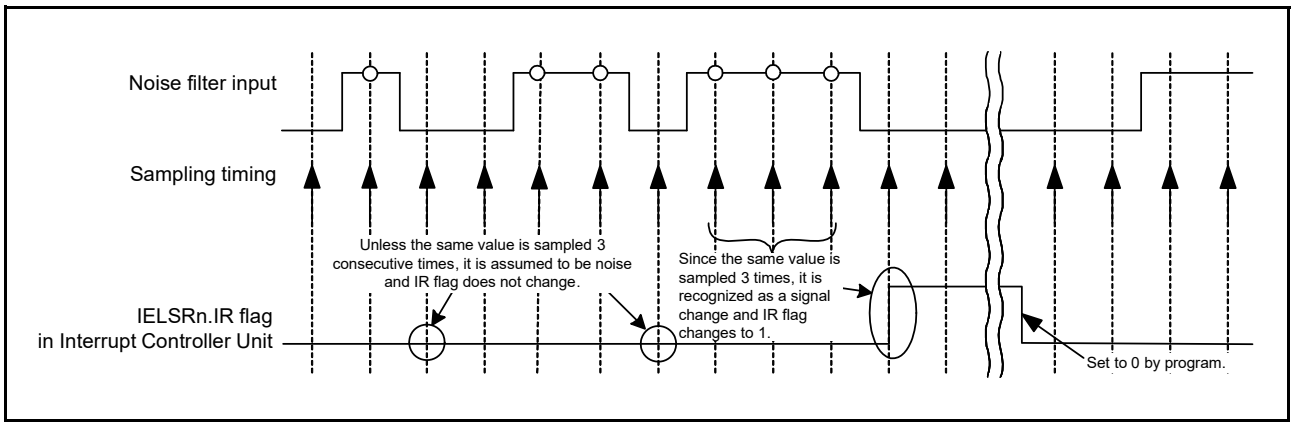


Figure 44.6 Noise filter and interrupt operation example

44.5 ACMPLP Interrupts

The ACMPLP generates two interrupt requests from sources ACMPLP0 and ACMPLP1. The ACMPLPi (i = 0 and 1) interrupt can be used by selecting it in the IELSRn register in the ICU.

To use the ACMPLPi interrupt, select either single-edge detection or both-edge detection using the COMPFIR.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 three different sampling clocks, as selected in the COMPFIR.CiFCK[1:0] bits. Set the COMPFIR.CiFCK[1:0] bits to 01b, 10b, or 11b to select the respective sampling clock.

To use the ACMPLP0 interrupt request to release Software Standby mode or Snooze mode, set COMPFIR.C0FCK[1:0] to 00b to bypass the ACMPLP0 noise filter. The ACMPLP1 interrupt request cannot be used to release Software Standby mode or Snooze mode.

44.6 ELC Event Output

The ELC uses the ACMPLPi interrupt request signal as an ELC event signal, enabling link operation for the preset module. To use the ELC events of the ACMPLPi, select them in the ELSRn register in the ELC. When using the ACMPLPi ELC event request, set the COMPFIR.CiFCK[1:0] bits to 01b, 10b, or 11b.

44.7 Interrupt Handling and ELC Linking

ACMPLPi 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 ACMPLPi to the ELC can be selected as a single-edge detection or both-edge detection by setting the COMPFIR.CiEDG bit. When the single-edge detection is selected, the polarity can be selected by the CiEPO bit.

44.8 Comparator Pin Output

The comparison result from ACMPLPi can be output to external pins. Use the COMPOCR.CiOP and CiOE bits to set the output polarity (non-inverted output or inverted output) and to enable or disable the comparison output. For the register settings and associated comparator output, see section 44.2.3, ACMPLP Output Control Register (COMPOCR).

To output the ACMPLP comparison result to the VCOUT output pin by the CPLOUTi, set the corresponding Port mn Pin Function Select Register (PmnPFS) in the I/O register.

44.9 Usage Notes

44.9.1 Settings for the Module-Stop Function

The Module Stop Control Register can enable or disable ACMPLP operation. The ACMPLP is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 11, Low Power Modes.

44.9.2 Relationship with A/D converter

Constraints apply on the simultaneous use of ACMPHS analog input and A/D converter analog input. For details, see section 39.8.13, Relationship between A/D converter, OPAMP, ACMPHS, and ACMPLP.

45. Capacitive Touch Sensing Unit (CTSU)

45.1 Overview

The Capacitive Touch Sensing Unit (CTSU) measures the electrostatic capacitance of the touch sensor. Changes in the electrostatic capacitance are determined by software that enables the CTSU to detect whether a finger is in contact with the touch sensor. The electrode surface of the touch sensor is usually enclosed with an electrical insulator so that a finger does not come into direct contact with the electrode.

As Figure 45.1 shows, electrostatic capacitance (parasitic capacitance) exists between the electrode and the surrounding insulators. Because the human body is an electrical conductor, when a finger is placed close to the electrode, the electrostatic capacitance increases.

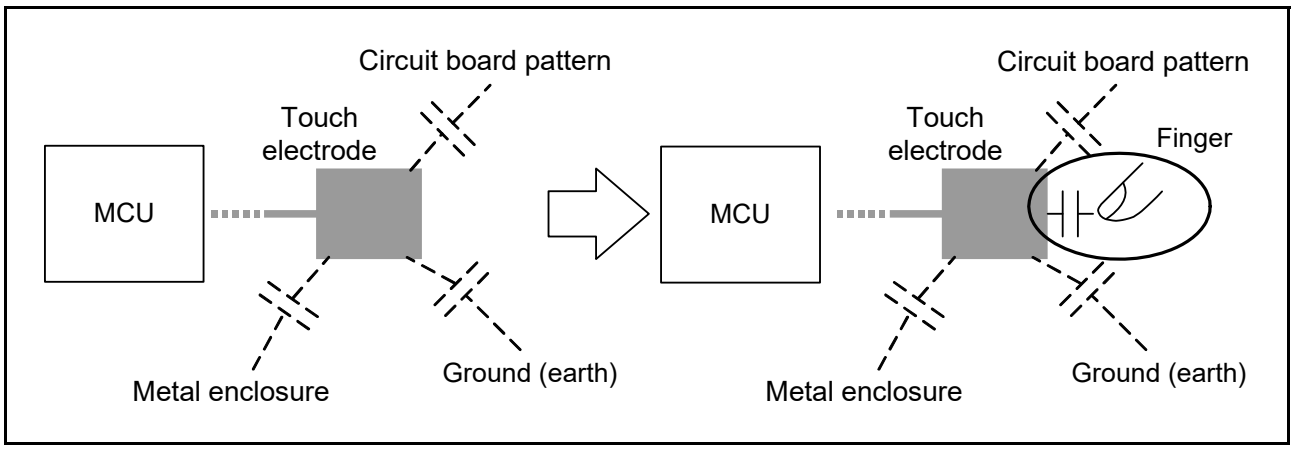


Figure 45.1 Increased electrostatic capacitance because of the presence of a finger

Electrostatic capacitance is detected by the self-capacitance and mutual-capacitance methods. In the self-capacitance method, the CTSU detects electrostatic capacitance generated between a finger and a single electrode. In the mutual-capacitance method, two electrodes are used as a transmit electrode and a receive electrode, and the CTSU detects the change in the electrostatic capacitance generated between the two when a finger is placed close to them.

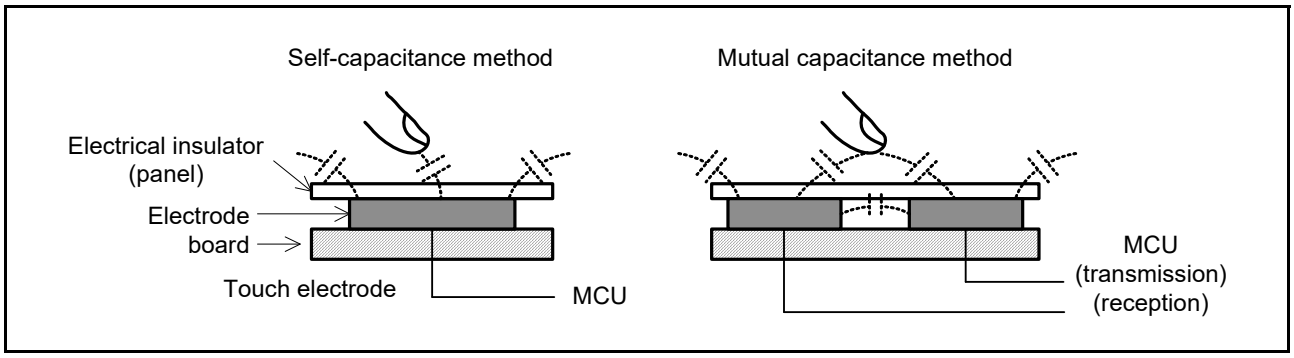


Figure 45.2 Self-capacitance and mutual-capacitance methods

Electrostatic capacitance is measured by counting a clock signal whose frequency changes according to the amount of charged or discharged current, for a specified period. For details on the measurement principles of the CTSU, see section 45.3.1, Principles of Measurement Operation. Table 45.1 lists the CTSU specifications and Figure 45.3 shows the block diagram.

Table 45.1 CTSU specifications

Parameter		Description
Operating clocks		PCLKB, PCLKB/2, or PCLKB/4
Pins	Electrostatic capacitance measurement	31 channels (TS00, TS01, TS03 to TS22, TS26 to TS27, TS29 to TS35)
	TSCAP	Low Pass Filter (LPF) connection pin
Measurement modes	Self-capacitance single scan mode	Electrostatic capacitance is measured on one channel using the self-capacitance method
	Self-capacitance multi-scan mode	Electrostatic capacitance is measured successively on multiple channels using the self-capacitance method
	Mutual-capacitance full-scan mode	Electrostatic capacitance is measured successively on multiple channels using the mutual-capacitance method
Noise prevention		Synchronous noise prevention, high-pass noise prevention
Measurement start conditions		• Software trigger • External trigger (ELC_CTSU from the Event Link Controller (ELC))

As Figure 45.3 shows, the CTSU consists of the following components:

- Status control block
- Trigger control block
- Clock control block
- Channel control block
- Port control block
- Sensor drive pulse generator
- Measurement block
- Interrupt block
- I/O registers.

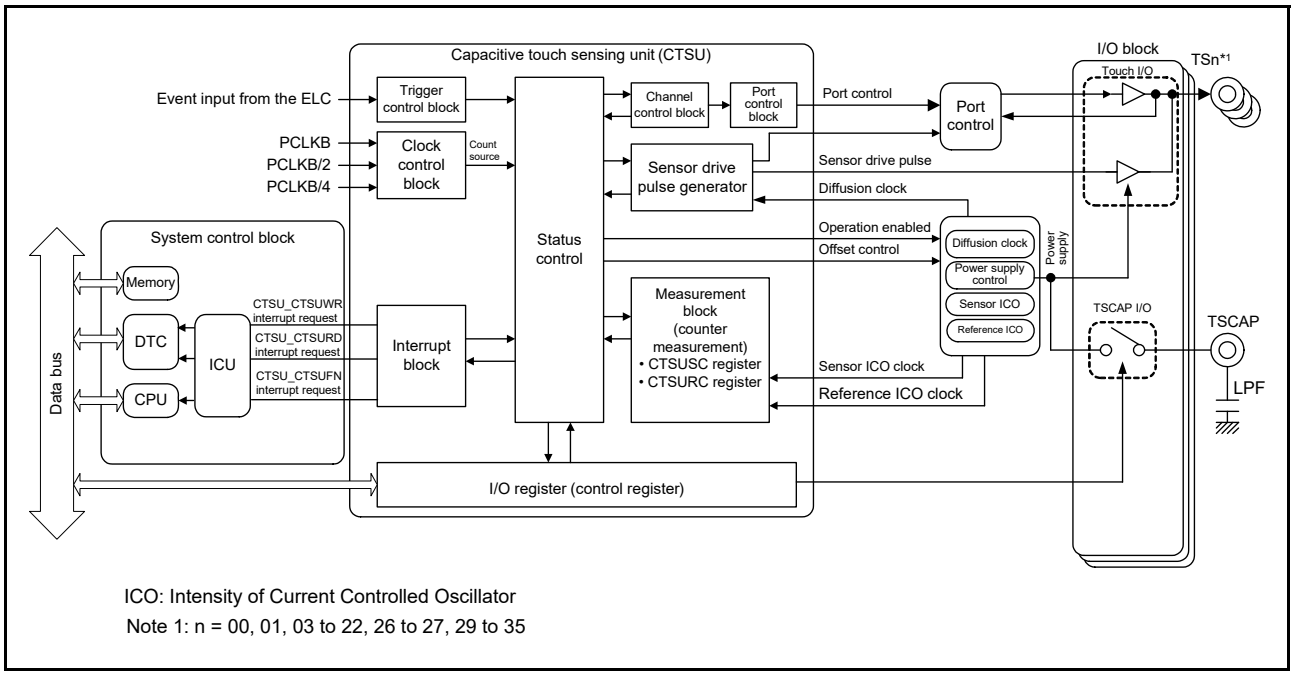


Figure 45.3 CTSU block diagram

Table 45.2 CTSU pin configuration

Pin name	I/O	Function
TS00, TS01, TS03 to TS22, TS26 to TS27, TS29 to TS35	Input	Electrostatic capacitive measurement pins (touch pins)
TSCAP	-	LPF connection pin

45.2 Register Descriptions

45.2.1 CTSU Control Register 0 (CTSUCR0)

Address(es): CTSU.CTSUCR0 4008 1000h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CTSUINIT	—	CTSUSNZ	CTSUCAP	CTSUSTRT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CTSUSTRT	CTSU Measurement Operation Start	0: Stop measurement operation*1 1: Start measurement operation.	R/W
b1	CTSUCAP	CTSU Measurement Operation Start Trigger Select	0: Software trigger 1: External trigger.	R/W
b2	CTSUSNZ	CTSU Wait State Power-Saving Enable	This bit sets the power-saving function during a wait state: 0: Disable power-saving function during wait state 1: Enable power-saving function during wait state.	R/W
b3	—	Reserved	This bit read as 0. The write value should be 0.	R/W
b4	CTSUINIT	CTSU Control Block Initialization	Writing 1 to this bit initializes the CTSU control block and the CTSUSC, CTSURC, CTSUMCH0, CTSUMCH1, and CTSUST registers. This bit is read as 0.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the CTSU is not used, fix the value of this bit to 0.

Only set the CTSUCAP and CTSUSNZ bits when the CTSUSTRT bit is 0. These bits can be set at the same time when measurement operation starts.

CTSUSTRT bit (CTSU Measurement Operation Start)

The CTSUSTRT bit specifies whether CTSU operation starts or stops. When the CTSUCAP bit is 0, measurement starts when the software writes 1 to the CTSUSTRT bit (software trigger), and stops when the hardware clears the CTSUSTRT bit to 0. When the CTSUCAP bit is 1, the CTSU waits for an external trigger by writing 1 to the CTSUSTRT bit, and measurement starts on the rising edge of the external trigger. When measurement is stopped, the CTSU waits for the next external trigger and operation continues.

Table 45.3 lists the CTSU states.

Table 45.3 CTSU states

CTSUSTRT bit	CTSUCAP bit	CTSU state
0	0	Stopped
0	1	Stopped
1	0	Measurement in progress
1	1	Measurement in progress and waiting for an external trigger*1

Note 1. The state can be read from the CTSUST.CTSUSTC[2:0] flags as follows:
During measurement: CTSUST.CTSUSTC[2:0] flags ≠ 000b
While waiting for an external trigger: CTSUST.CTSUSTC[2:0] flags = 000b

If the software sets the CTSUSTRT bit to 1 when the bit is already 1, the write is ignored and operation continues. To force operation to stop through the software when the CTSUSTRT bit is 1, set the CTSUSTRT bit to 0 and the CTSUINIT bit to 1 at the same time.

CTSUCAP bit (CTSU Measurement Operation Start Trigger Select)

The CTSUCAP bit specifies the measurement start condition. For details, see CTSUSTRT bit (CTSU Measurement Operation Start).

CTSUSNZ bit (CTSU Wait State Power-Saving Enable)

The CTSUSNZ bit enables or disables power-saving operation during a wait state. It can also suspend the CTSU power supply, which decreases power consumption during the wait state. In the suspended state, the CTSU power supply is turned off while the external TSCAP is still charged.

Table 45.4 shows the CTSU power supply state control.

Table 45.4 CTSU power supply state control

CTSUCR1.CTSUPON bit	CTSUSNZ bit	CTSUCAP bit	CTSUSTRT bit	CTSU power supply state
0	0	0	0	Stopped
1	0	-	-	Operating
1	1	0	0	Suspended

Note: Settings other than those listed in the table are prohibited.

To start measurement from the suspended state, set the CTSUSNZ bit to 0, and then set the CTSUSTRT bit to 1. To suspend the module after measurement stops, set the CTSUSNZ bit to 1.

CTSUINIT bit (CTSU Control Block Initialization)

Write 1 to the CTSUINIT bit to initialize the control registers. To force the current operation to stop, set the CTSUSTRT bit to 0 and the CTSUINIT bit to 1 at the same time. This stops the operation and initialize the internal control registers.

Do not write 1 to the CTSUINIT bit when the CTSUSTRT bit is 1.

45.2.2 CTSU Control Register 1 (CTSUCR1)

Address(es): CTSU.CTSUCR1 4008 1001h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUMD[1:0]		CTSUCLK[1:0]		CTSUATUNE1	CTSUATUNE0	CTSUCSW	CTSUPON
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	CTSUPON	CTSU Power Supply Enable	This bit controls the CTSU power supply: 0: Power off 1: Power on.	R/W
b1	CTSUCSW	CTSU LPF Capacitance Charging Control	This bit controls charging of the LPF capacitance connected to the TSCAP pin: 0: Turn off capacitance switch 1: Turn on capacitance switch.	R/W
b2	CTSUATUNE0	CTSU Power Supply Operating Mode Setting	VCC ≥ 2.4 V 0: Normal operating mode 1: Low-voltage operating mode. VCC < 2.4 V 0: Setting prohibited 1: Low-voltage operating mode.	R/W
b3	CTSUATUNE1	CTSU Power Supply Capacity Adjustment	0: Normal output 1: High-current output.	R/W

Bit	Symbol	Bit name	Description	R/W
b5, b4	CTSUCLK[1:0]	CTSU Operating Clock Select	These bits select the operating clock: b5 b4 0 0: PCLKB 0 1: PCLKB/2 (PCLKB divided by 2) 1 0: PCLKB/4 (PCLKB divided by 4) 1 1: Setting prohibited.	R/W
b7, b6	CTSUMD[1:0]	CTSU Measurement Mode Select	These bits select the measurement mode: b7 b6 0 0: Self-capacitance single scan mode 0 1: Self-capacitance multi-scan mode 1 0: Setting prohibited 1 1: Mutual capacitance full-scan mode.	R/W

Only set the CTSUCR1 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUPON bit (CTSU Power Supply Enable)

The CTSUPON bit controls the power supply to the CTSU. Set the CTSUPON and CTSUCSW bits to the same value.

CTSUCSW bit (CTSU LPF Capacitance Charging Control)

The CTSUCSW bit controls charging of the LPF capacitor connected to the TSCAP pin by turning the capacitance switch on or off. After the capacitance switch is turned on, wait until the capacitance connected to the TSCAP pin is charged for the specified time before starting measurement by setting CTSUCR0.CTSUSTRT to 1. Before starting measurement, use an I/O port to output low to the TSCAP pin, and discharge the existing LPF capacitance. Set the CTSUPON and CTSUCSW bits to the same value.

CTSUATUNE0 bit (CTSU Power Supply Operating Mode Setting)

The CTSUATUNE0 bit sets the power supply operating mode. Set this bit to the lower limit of VCC to operate the CTSU. As an example, when performing touch measurement in a system where VCC varies depending on battery operation, set this bit to 1 regardless of the initial VCC voltage. The VCC voltage range is 2 to 3 V.

CTSUATUNE1 bit (CTSU Power Supply Capacity Adjustment)

The CTSUATUNE1 bit sets the capacity of the CTSU power supply. In general, set this bit to 0.

CTSUCLK[1:0] bits (CTSU Operating Clock Select)

The CTSUCLK[1:0] bits select the operating clock.

CTSUMD[1:0] bits (CTSU Measurement Mode Select)

The CTSUMD bits set the measurement mode. For details, see section 45.3.2, Measurement Modes.

45.2.3 CTSU Synchronous Noise Reduction Setting Register (CTSUSDPRS)

Address(es): CTSU.CTSUSDPRS 4008 1002h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	CTSUSOFF	CTSUPRMODE[1:0]		CTSUPRRATIO[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b3 to b0	CTSUPRRATIO[3:0]	CTSU Measurement Time and Pulse Count Adjustment	These bits set the measurement time and the measurement pulse count. The recommended setting is 3 (0011b)	R/W
b5, b4	CTSUPRMODE[1:0]	CTSU Base Period and Pulse Count Setting	These bits set the base pulse count: b5 b4 0 0: 510 pulses 0 1: 126 pulses 1 0: 62 pulses (recommended setting value) 1 1: Setting prohibited.	R/W
b6	CTSUSOFF	CTSU High-Pass Noise Reduction Function Off Setting	This bit controls spectrum diffusion, which can be used to reduce high-pass noise: 0: Turn spectrum diffusion on 1: Turn spectrum diffusion off.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Only set the CTSUSDPRS register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUPRRATIO[3:0] bits (CTSU Measurement Time and Pulse Count Adjustment)

The CTSUPRRATIO[3:0] bits set the measurement time and the number of measurement pulses using the following formulas, where the number of base pulses is determined by the CTSUPRMODE[1:0] setting:

Measurement pulse count = base pulse count × (CTSUPRRATIO[3:0] bits + 1)
Measurement time = (base pulse count × (CTSUPRRATIO[3:0] bits + 1) + base pulse count - 2) × 0.25 × base clock cycle

Note: For details on the base clock cycle, see section 45.2.21, CTSU Sensor Offset Register 1 (CTSUSO1).

CTSUPRMODE[1:0] bits (CTSU Base Period and Pulse Count Setting)

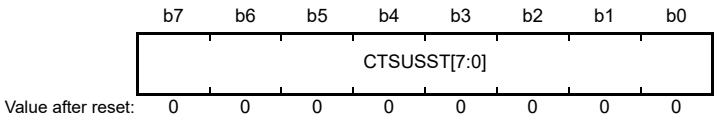
The CTSUPRMODE[1:0] bits select the number of base pulses that occur during measurement.

CTSUSOFF bit (CTSU High-Pass Noise Reduction Function Off Setting)

The CTSUSOFF bit turns on or off the function for reducing high-pass noise. Set this bit to 1 to turn the function off.

45.2.4 CTSU Sensor Stabilization Wait Control Register (CTSUSST)

Address(es): CTSU.CTSUSST 4008 1003h



Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUSST[7:0]	CTSU Sensor Stabilization Wait Control	Set the value of these bits to 00010000b.	R/W

Only set the CTSUSST register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUSST[7:0] bits (CTSU Sensor Stabilization Wait Control)

The CTSUSST[7:0] bits set the stabilization wait time for the TSCAP pin voltage. Always set these bits to 00010000b. If these bits are not set, the TSCAP voltage becomes unstable at the start of measurement, and the CTSU is unable to obtain correct touch measurement results.

45.2.5 CTSU Measurement Channel Register 0 (CTSUMCH0)

Address(es): CTSU.CTSUMCH0 4008 1004h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CTSUMCH0[5:0]					
Value after reset:	0	0	1	1	1	1	1	1

Bit	Symbol	Bit name	Description	R/W
b5 to b0	CTSUMCH0[5:0]	CTSU Measurement Channel 0	In self-capacitance single scan mode, these bits set a channel to be measured: b5 b0 0 0 0 000: TS00 0 0 0 001: TS01 0 0 0 011: TS03 0 0 0 100: TS04 0 0 0 101: TS05 0 0 0 110: TS06 0 0 0 111: TS07 0 0 1 000: TS08 0 0 1 001: TS09 0 0 1 010: TS10 0 0 1 011: TS11 0 0 1 100: TS12 0 0 1 101: TS13 0 0 1 110: TS14 0 0 1 111: TS15 0 1 0 000: TS16 0 1 0 001: TS17 0 1 0 010: TS18 0 1 0 011: TS19 0 1 0 100: TS20 0 1 0 101: TS21 0 1 0 110: TS22 0 1 1 010: TS26 0 1 1 011: TS27 0 1 1 101: TS29 0 1 1 110: TS30 0 1 1 111: TS31 1 0 0 000: TS32 1 0 0 001: TS33 1 0 0 010: TS34 1 0 0 011: TS35. Other than those specified, the starting measurement operation when CTSUCR0.CTSUSTRT = 1 is prohibited after these bits are set.	R/W*1

Bit	Symbol	Bit name	Description	R/W
			In other measurement modes, these bits indicate the channel that is currently being measured: b5 b0 0 0 0 000: TS00 0 0 0 001: TS01 0 0 0 011: TS03 0 0 0 100: TS04 0 0 0 101: TS05 0 0 0 110: TS06 0 0 0 111: TS07 0 0 1 000: TS08 0 0 1 001: TS09 0 0 1 010: TS10 0 0 1 011: TS11 0 0 1 100: TS12 0 0 1 101: TS13 0 0 1 110: TS14 0 0 1 111: TS15 0 1 0 000: TS16 0 1 0 001: TS17 0 1 0 010: TS18 0 1 0 011: TS19 0 1 0 100: TS20 0 1 0 101: TS21 0 1 0 110: TS22 0 1 1 010: TS26 0 1 1 011: TS27 0 1 1 101: TS29 0 1 1 110: TS30 0 1 1 111: TS31 1 0 0 000: TS32 1 0 0 001: TS33 1 0 0 010: TS34 1 0 0 011: TS35 1 1 1 111: Measure is being stopped.	
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is enabled only in self-capacitance single scan mode (CTSUCR1.CTSUMD[1:0] bits = 00b).

Only set the CTSUMCH0 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUMCH0[5:0] bits (CTSU Measurement Channel 0)

In self-capacitance single scan mode, the CTSUMCH0[5:0] bits set the channel to be measured. In this mode, only specify enabled channels (000000b, 000001b, 000011b to 010110b, 011010b, 011011b, and 011101b to 100011b). In other modes, these indicate the receive channel that is being measured and writing to these bits has no effect.

45.2.6 CTSU Measurement Channel Register 1 (CTSUMCH1)

Address(es): CTSU.CTSUMCH1 4008 1005h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CTSUMCH1[5:0]					
Value after reset:	0	0	1	1	1	1	1	1

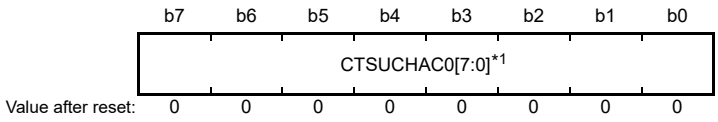
Bit	Symbol	Bit name	Description	R/W
b5 to b0	CTSUMCH1[5:0]	CTSU Measurement Channel 1	b5 b0 0 0 0 000: TS00 0 0 0 001: TS01 0 0 0 011: TS03 0 0 0 100: TS04 0 0 0 101: TS05 0 0 0 110: TS06 0 0 0 111: TS07 0 0 1 000: TS08 0 0 1 001: TS09 0 0 1 010: TS10 0 0 1 011: TS11 0 0 1 100: TS12 0 0 1 101: TS13 0 0 1 110: TS14 0 0 1 111: TS15 0 1 0 000: TS16 0 1 0 001: TS17 0 1 0 010: TS18 0 1 0 011: TS19 0 1 0 100: TS20 0 1 0 101: TS21 0 1 0 110: TS22 0 1 1 010: TS26 0 1 1 011: TS27 0 1 1 101: TS29 0 1 1 110: TS30 0 1 1 111: TS31 1 0 0 000: TS32 1 0 0 001: TS33 1 0 0 010: TS34 1 0 0 011: TS35 1 1 1 111: Measurement is being stopped.	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CTSUMCH1[5:0] bits (CTSU Measurement Channel 1)

In full-scan mode, the CTSUMCH1[5:0] bits indicate the transmit channel that is being measured. The value of these bits is 111111b when measurement is stopped, or when in self-capacitance single scan or multi-scan mode.

45.2.7 CTSU Channel Enable Control Register 0 (CTSUCHAC0)

Address(es): CTSU.CTSUCHAC0 4008 1006h



Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHAC0[7:0]	CTSU Channel Enable Control 0	These bits select whether the associated TS pin is measured: 0: Do not measure 1: Measure. These bits specify the TS00, TS01, and TS03 to TS07 pins.	R/W

Note 1. The MCU does not support the TS02 pin, and therefore b2 (CTSUCHAC0[2]) is read as 0. The write value should be 0.

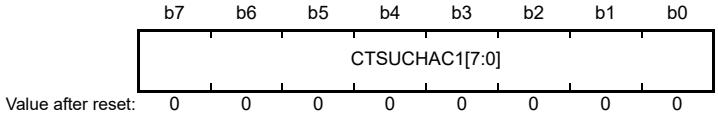
Only set the CTSUCHAC0 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC0[7:0] bits (CTSU Channel Enable Control 0)

The CTSUCHAC0[7:0] bits select the receive and transmit pins whose electrostatic capacitance is to be measured. CTSUCHAC0[0] is associated with TS00 and CTSUCHAC0[7] with TS07.

45.2.8 CTSU Channel Enable Control Register 1 (CTSUCHAC1)

Address(es): CTSU.CTSUCHAC1 4008 1007h



Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHAC1[7:0]	CTSU Channel Enable Control 1	These bits select whether the associated TS pin is measured. 0: Do not measure 1: Measure. These bits specify the TS08 to TS15 pins.	R/W

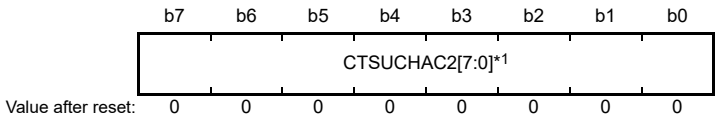
Only set the CTSUCHAC1 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC1[7:0] bits (CTSU Channel Enable Control 1)

The CTSUCHAC1[7:0] bits select the receive and transmit pins whose electrostatic capacitance is to be measured. CTSUCHAC1[0] is associated with TS08 and CTSUCHAC1[7] with TS15.

45.2.9 CTSU Channel Enable Control Register 2 (CTSUCHAC2)

Address(es): CTSU.CTSUCHAC2 4008 1008h



Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHAC2[7:0] *1	CTSU Channel Enable Control 2	These bits select whether the associated TS pin is measured. 0: Do not measure 1: Measure. These bits specify the TS16 to TS22 pins.	R/W

Note 1. The MCU does not support the TS23 pin, and therefore CTSUCHAC2[7] is read as 0.

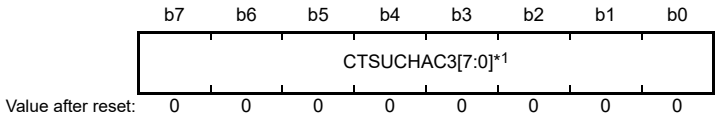
Only set the CTSUCHAC2 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC2[7:0]*1 bits (CTSU Channel Enable Control 2)

The CTSUCHAC2[7:0] bits select the receive and transmit pins for which electrostatic capacitance is to be measured. CTSUCHAC2[0] is associated with TS16 and CTSUCHAC2[7] with TS23.

45.2.10 CTSU Channel Enable Control Register 3 (CTSUCHAC3)

Address(es): CTSU.CTSUCHAC3 4008 1009h



Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHAC3[7:0] *1	CTSU Channel Enable Control 3	These bits select whether the associated TS pins is measured: 0: Do not measure 1: Measure. These bits specify the TS26, TS27, and TS29 to TS31 pins.	R/W

Note 1. The MCU does not support TS24 pin, and therefore CTSUCHAC3[0] is read as 0. The write value should be 0.
The MCU does not support TS25 pin, and therefore CTSUCHAC3[1] is read as 0. The write value should be 0.
The MCU does not support TS28 pin, and therefore CTSUCHAC3[4] is read as 0. The write value should be 0.

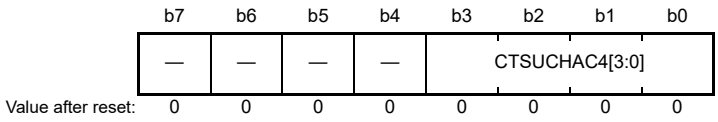
Only set the CTSUCHAC3 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC3[7:0]*1 bits (CTSU Channel Enable Control 3)

The CTSUCHAC3[7:0] bits select the receive and transmit pins for which electrostatic capacitance is to be measured. CTSUCHAC3[0] is associated with TS24 and CTSUCHAC3[7] with TS31.

45.2.11 CTSU Channel Enable Control Register 4 (CTSUCHAC4)

Address(es): CTSU.CTSUCHAC4 4008 100Ah



Bit	Symbol	Bit name	Description	R/W
b3 to b0	CTSUCHAC4[3:0]	CTSU Channel Enable Control 4	These bits select whether the associated TS pin is measured: 0: Do not measure 1: Measure. These bits specify the TS32 to TS35 pins.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

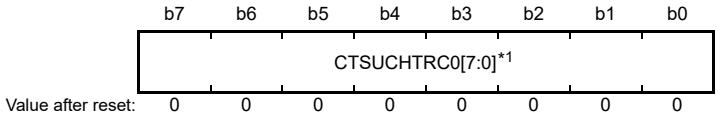
Only set the CTSUCHAC4 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC4[3:0] bits (CTSU Channel Enable Control 4)

The CTSUCHAC4[3:0] bits select the receive and transmit pins for which electrostatic capacitance is to be measured. CTSUCHAC4[0] is associated with TS32 and CTSUCHAC4[3] with TS35.

45.2.12 CTSU Channel Transmit/Receive Control Register 0 (CTSUCHTRC0)

Address(es): CTSU.CTSUCHTRC0 4008 100Bh



Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHTRC0[7:0]	CTSU Channel Transmit/Receive Control 0	0: Reception 1: Transmission. These bits specify the TS00, TS01, and TS03 to TS07 pins.	R/W

Note 1. The MCU does not support TS02 pin, and therefore CTSUCHTRC0[2] is read as 0. The write value should be 0.

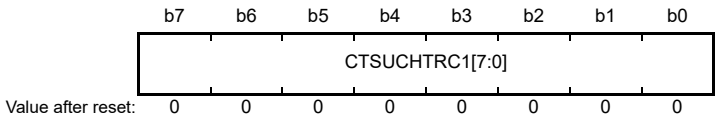
Only set the CTSUCHTRC0 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC0[7:0] bits (CTSU Channel Transmit/Receive Control 0)

In full-scan mode, the CTSUCHTRC0[7:0] bits assign the associated TS pins to reception or transmission. The setting of these bits is ignored in self-capacitance single scan and multi-scan modes. CTSUCHTRC0[0] is associated with TS00 and CTSUCHTRC0[7] with TS07.

45.2.13 CTSU Channel Transmit/Receive Control Register 1 (CTSUCHTRC1)

Address(es): CTSU.CTSUCHTRC1 4008 100Ch



Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHTRC1[7:0]	CTSU Channel Transmit/Receive Control 1	0: Reception 1: Transmission. These bits specify the TS08 to TS15 pins.	R/W

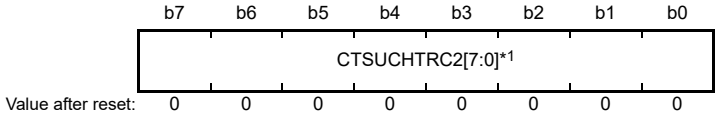
Only set the CTSUCHTRC1 bit when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC1[7:0] bits (CTSU Channel Transmit/Receive Control 1)

In full-scan mode, the CTSUCHTRC1[7:0] bits allocate reception or transmission to the associated TS pins. The setting is ignored in self-capacitance single scan and multi-scan modes. CTSUCHTRC1[0] is associated with TS08 and CTSUCHTRC1[7] with TS15.

45.2.14 CTSU Channel Transmit/Receive Control Register 2 (CTSUCHTRC2)

Address(es): CTSU.CTSUCHTRC2 4008 100Dh



Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHTRC2[7:0]*1	CTSU Channel Transmit/Receive Control 2	0: Reception 1: Transmission. These bits specify the TS16 to TS22 pins.	R/W

Note 1. The MCU does not support TS23 pin, and therefore CTSUCHTRC2[7] is read as 0. The write value should be 0.

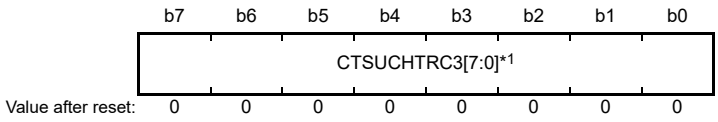
Only set the CTSUCHTRC2 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC2[7:0]*1 bits (CTSU Channel Transmit/Receive Control 2)

In full-scan mode, the CTSUCHTRC2[7:0] bits allocate reception or transmission to the associated TS pins. The setting of these bits is ignored in self-capacitance single scan and multi-scan modes. CTSUCHTRC2[0] is associated with TS16 and CTSUCHTRC2[7] with TS23.

45.2.15 CTSU Channel Transmit/Receive Control Register 3 (CTSUCHTRC3)

Address(es): CTSU.CTSUCHTRC3 4008 100Eh



Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSUCHTRC3[7:0]*1	CTSU Channel Transmit/Receive Control 3	0: Reception 1: Transmission. These bits specify the TS26, TS27, and TS29 to TS31 pins.	R/W

Note 1. The MCU does not support TS24 pin, therefore CTSUCHTRC3[0] is read as 0. The write value should be 0.
The MCU does not support TS25 pin, therefore CTSUCHTRC3[1] is read as 0. The write value should be 0.
The MCU does not support TS28 pin, therefore CTSUCHTRC3[4] is read as 0. The write value should be 0.

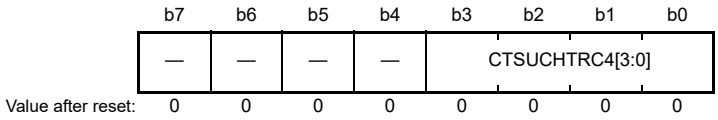
Only set the CTSUCHTRC3 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC3[7:0]*1 bits (CTSU Channel Transmit/Receive Control 3)

In full-scan mode, the CTSUCHTRC3[7:0] bits allocate reception or transmission to the associated TS pins. The setting of these bits is ignored in self-capacitance single scan and multi-scan modes. CTSUCHTRC3[0] is associated with TS24 and CTSUCHTRC3[7] with TS31.

45.2.16 CTSU Channel Transmit/Receive Control Register 4 (CTSUCHTRC4)

Address(es): CTSU.CTSUCHTRC4 4008 100Fh



Bit	Symbol	Bit name	Description	R/W
b3 to b0	CTSUCHTRC4[3:0]	CTSU Channel Transmit/ Receive Control 4	0: Reception 1: Transmission. These bits specify the TS32 to TS35 pins.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Only set the CTSUCHTRC4 register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC4[3:0] bits (CTSU Channel Transmit/Receive Control 4)

In full-scan mode, the CTSUCHTRC4[3:0] bits allocate reception or transmission to the associated TS pins. The setting of these bits is ignored in self-capacitance single scan and multi-scan modes. CTSUCHTRC4[0] is associated with TS32 and CTSUCHTRC4[3] with TS35.

45.2.17 CTSU High-Pass Noise Reduction Control Register (CTSUDCLKC)

Address(es): CTSU.CTSUDCLKC 4008 1010h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CTSUSSCNT[1:0]		—	—	CTSUSSMOD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	CTSUSSMOD[1:0]	CTSU Diffusion Clock Mode Select	Set these bits to 00b	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	CTSUSSCNT[1:0]	CTSU Diffusion Clock Mode Control	Set these bits to 11b	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Only set the CTSUDCLKC register when the CTSUCR0.CTSUSTRT bit is 0.

CTSUSSMOD[1:0] bits (CTSU Diffusion Clock Mode Select)

The CTSUSSMOD[1:0] bits set the mode of the spectrum diffusion clock for high-pass noise reduction. When using the high-pass noise reduction function, always fix these bits to 00b. If these bits are not set, the CTSU is unable to effectively reduce high-pass noise.

CTSUSSCNT[1:0] bits (CTSU Diffusion Clock Mode Control)

The CTSUSSCNT[1:0] bits adjust the amount of spectrum diffusion applied to reduce high-pass noise. When using the high-pass noise reduction function, always fix these bits to 11b. If these bits are not set, touch measurement might be performed incorrectly.

45.2.18 CTSU Status Register (CTSUST)

Address(es): CTSU.CTSUST 4008 1011h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUPS	CTSUROVF	CTSUSOVF	CTSUDTSR	—	CTSUSTC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b2 to b0	CTSUSTC[2:0]	CTSU Measurement Status Counter	These counters indicate the current measurement status: b2 b0 0 0 0: Status 0 0 0 1: Status 1 0 1 0: Status 2 0 1 1: Status 3 1 0 0: Status 4 1 0 1: Status 5.	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	CTSUDTSR	CTSU Data Transfer Status Flag	This flag indicates whether the measurement result stored in the sensor counter and the reference counter was read: 0: Read 1: Not read.	R
b5	CTSUSOVF	CTSU Sensor Counter Overflow Flag	This flag indicates an overflow on the sensor counter: 0: No overflow occurred 1: Overflow occurred.	R/W

Bit	Symbol	Bit name	Description	R/W
b6	CTSUROVF	CTSU Reference Counter Overflow Flag	This flag indicates an overflow on the reference counter: 0: No overflow occurred 1: Overflow occurred.	R/W
b7	CTSUPS	CTSU Mutual Capacitance Status Flag	This flag indicates the measurement status in mutual-capacitance full-scan mode: 0: First measurement 1: Second measurement.	R

When using the CTSUCR0.CTSUINIT bit to clear an overflow flag, make sure that the CTSUCR0.CTSUSTRT bit is 0.

CTSUSTC[2:0] flags (CTSU Measurement Status Counter)

The CTSUSTC[2:0] flags are a counter indicating the current measurement status. For details on each status, see section 45.3.2.2, Status Counter.

CTSUDTSR flag (CTSU Data Transfer Status Flag)

The CTSUDTSR flag indicates whether the measurement result stored in the sensor counter and the reference counter was read. This flag is set to 1 when measurement completes and 0 when the reference counter is read by software or the DTC. This flag can also be cleared with the CTSUCR0.CTSUINIT bit.

CTSUSOVF flag (CTSU Sensor Counter Overflow Flag)

The CTSUSOVF flag is set to 1 when the sensor counter, CTSUSC, overflows. On overflow, the counter value reads as FFFFh. Measurement processing continues for the specified period.

No interrupt occurs on an overflow. To determine the channel on which the overflow occurred, read the measurement result of each channel after measurement completes, as signaled by a measurement end interrupt.

This flag is cleared when 0 is written after 1 is read by the software. This flag can also be cleared with the CTSUCR0.CTSUINIT bit.

CTSUROVF flag (CTSU Reference Counter Overflow Flag)

The CTSUROVF flag indicates when the reference counter, CTSUSC, overflows. On overflow, the counter value reads as FFFFh. Measurement processing continues for the specified period.

No interrupt is generated even when an overflow occurs. To determine the channel on which the overflow occurred, read the measurement result of each channel after measurement completes, as signaled by a measurement end interrupt.

This flag is cleared when 0 is written after 1 is read by the software. This flag can also be cleared with the CTSUCR0.CTSUINIT bit.

CTSUPS flag (CTSU Mutual Capacitance Status Flag)

In mutual capacitance full-scan mode, when CTSUCR1.CTSUMD[1:0] bits = 11b, the CTSUPS flag indicates whether the measurement is the first or second of two measurements for each channel. When measurement is stopped or in other measurement modes, this flag is always 0.

45.2.19 CTSU High-Pass Noise Reduction Spectrum Diffusion Control Register (CTSUSSC)

Address(es): CTSU.CTSUSSC 4008 1012h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CTSUSSDIV[3:0]				—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	CTSUSSDIV[3:0]	CTSU Spectrum Diffusion Frequency Division Setting	These bits specify the spectrum diffusion frequency division setting based on the base clock frequency division setting.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CTSUSSDIV[3:0] bits (CTSU Spectrum Diffusion Frequency Division Setting)

The CTSUSSDIV[3:0] bits specify the spectrum diffusion frequency derived from the base clock frequency division setting. To calculate the correct setting for CTSUSSDIV[3:0], see the relationship between base clock frequencies and the settings in Table 45.5.

Table 45.5 Relationship between base clock frequencies and CTSUSSDIV[3:0] bit settings

Base clock frequency fb (MHz)	CTSUSSDIV[3:0] bit setting
4.00 ≤ fb	0000b
2.00 ≤ fb < 4.00	0001b
1.33 ≤ fb < 2.00	0010b
1.00 ≤ fb < 1.33	0011b
0.80 ≤ fb < 1.00	0100b
0.67 ≤ fb < 0.80	0101b
0.57 ≤ fb < 0.67	0110b
0.50 ≤ fb < 0.57	0111b
0.44 ≤ fb < 0.50	1000b
0.40 ≤ fb < 0.44	1001b
0.36 ≤ fb < 0.40	1010b
0.33 ≤ fb < 0.36	1011b
0.31 ≤ fb < 0.33	1100b
0.29 ≤ fb < 0.31	1101b
0.27 ≤ fb < 0.29	1110b
fb < 0.27	1111b

45.2.20 CTSU Sensor Offset Register 0 (CTSUSO0)

Address(es): CTSU.CTSUSO0 4008 1014h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUSNUM[5:0]						CTSUSO[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b9 to b0	CTSUSO[9:0]	CTSU Sensor Offset Adjustment	These bits adjust the electronic capacitance when the electrode is not being touched: b9 b0 0 0 0 0 0 0 0 0 0 0: Current offset is 0 0 0 0 0 0 0 0 0 0 1: Current offset is 1 0 0 0 0 0 0 0 0 1 0: Current offset is 2 : : 1 1 1 1 1 1 1 1 1 0: Current offset is 1022 1 1 1 1 1 1 1 1 1 1: Current offset is maximum.	R/W
b15 to b10	CTSUSNUM[5:0]	CTSU Measurement Count Setting	These bits set the number of measurements.	R/W

CTSUSO[9:0] bits (CTSU Sensor Offset Adjustment)

The CTSUSO[9:0] bits offset the sensor ICO input current generated from electrostatic capacitance during touch measurement when the electrode is not being touched. This prevents the CTSU sensor counter from overflowing. Set the TS pin that is to be measured next after a CTSU_CTSUWR interrupt occurs.

CTSUSNUM[5:0] bits (CTSU Measurement Count Setting)

The CTSUSNUM[5:0] bits specify how many times the measurement pulse count specified in the CTSUSDPRS.CTSUPRRATIO[3:0] and CTSUSDPRS.CTSUPRMODE[1:0] bits is repeated during the measurement time. The measurement pulse count is repeated (CTSUSNUM[5:0] bits + 1) times. Set the TS pin that is to be measured next after a CTSU_CTSUWR interrupt is generated.

45.2.21 CTSU Sensor Offset Register 1 (CTSUSO1)

Address(es): CTSU.CTSUSO1 4008 1016h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	CTSUICOG[1:0]		CTSUSDPA[4:0]					CTSURICOA[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b7 to b0	CTSURICOA[7:0]	CTSU Reference ICO Current Adjustment	These bits adjust the input current of the reference ICO: b7 b0 0 0 0 0 0 0 0 0: Current offset is 0 0 0 0 0 0 0 0 1: Current offset is 1 0 0 0 0 0 0 1 0: Current offset is 2. : : 1 1 1 1 1 1 1 0: Current offset is 254 1 1 1 1 1 1 1 1: Current offset is maximum.	R/W

Bit	Symbol	Bit name	Description	R/W
b12 to b8	CTSUSDPA[4:0]	CTSU Base Clock Setting	These bits are used to generate the base clock: b12 b8 0 0 0 0 0: Operating clock divided by 2*1 0 0 0 0 1: Operating clock divided by 4 0 0 0 1 0: Operating clock divided by 6 0 0 0 1 1: Operating clock divided by 8 0 0 1 0 0: Operating clock divided by 10 0 0 1 0 1: Operating clock divided by 12 0 0 1 1 0: Operating clock divided by 14 0 0 1 1 1: Operating clock divided by 16 0 1 0 0 0: Operating clock divided by 18 0 1 0 0 1: Operating clock divided by 20 0 1 0 1 0: Operating clock divided by 22 0 1 0 1 1: Operating clock divided by 24 0 1 1 0 0: Operating clock divided by 26 0 1 1 0 1: Operating clock divided by 28 0 1 1 1 0: Operating clock divided by 30 0 1 1 1 1: Operating clock divided by 32 1 0 0 0 0: Operating clock divided by 34 1 0 0 0 1: Operating clock divided by 36 1 0 0 1 0: Operating clock divided by 38 1 0 0 1 1: Operating clock divided by 40 1 0 1 0 0: Operating clock divided by 42 1 0 1 0 1: Operating clock divided by 44 1 0 1 1 0: Operating clock divided by 46 1 0 1 1 1: Operating clock divided by 48 1 1 0 0 0: Operating clock divided by 50 1 1 0 0 1: Operating clock divided by 52 1 1 0 1 0: Operating clock divided by 54 1 1 0 1 1: Operating clock divided by 56 1 1 1 0 0: Operating clock divided by 58 1 1 1 0 1: Operating clock divided by 60 1 1 1 1 0: Operating clock divided by 62 1 1 1 1 1: Operating clock divided by 64.	R/W
b14, b13	CTSUICOG[1:0]	CTSU ICO Gain Adjustment	These bits adjust the output frequency gain of the sensor ICO and the reference ICO: b14 b13 0 0: 100% gain 0 1: 66% gain 1 0: 50% gain 1 1: 40% gain.	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Do not set the CTSUSDPA[4:0] bits to 00000b while the high-pass noise reduction function is turned off (CTSUSDPRS.CTSUSOFF bit = 1) in mutual-capacitance full scan mode (CTSUCR1.CTSUMD[1:0] bits = 11b).

After a CTSU_CTSUWR interrupt is generated, write first to the CTSUSSC register, next to the CTSUSO0 register, and then to the CTSUSO1 register. The write to the CTSUSO1 register causes a transition to Status 3 (see Table 45.6 and Table 45.7). Set all the bits in a single operation when writing to the CTSUSO1 register.

CTSURICOA[7:0] bits (CTSU Reference ICO Current Adjustment)

The CTSURICOA[7:0] bits adjust the oscillation frequency using the input current of the reference ICO.

CTSUSDPA[4:0] bits (CTSU Base Clock Setting)

The CTSUSDPA[4:0] bits select a base clock used as the source for the sensor drive pulse by dividing the operating clock. For details on the setting procedure, see section 45.3.2.1, Initial Setting Flow.

CTSUICOG[1:0] bits (CTSU ICO Gain Adjustment)

The CTSUICOG[1:0] bits adjust the output frequency gain of the sensor ICO and the reference ICO. In general, set the bits to 00b for the maximum gain. If changes in the capacitance between when the electrode is touched and when it is not touched greatly exceed the dynamic range of the sensor ICO, adjust the gain appropriately with this setting.

45.2.22 CTSU Sensor Counter (CTSUSC)

Address(es): CTSU.CTSUSC 4008 1018h

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

Bit	Symbol	Bit name	Description	R/W
b15 to b0	CTSUSC[15:0]	CTSU Sensor Counter	These bits indicate the measurement result of the sensor ICO. These bits read FFFFh when an overflow occurs.	R

After a CTSU_CTSURD interrupt is generated, read first from the CTSUSC counter, then from the CTSURC counter.

CTSUSC[15:0] bits (CTSU Sensor Counter)

The CTSUSC[15:0] bits are configured as an increment counter for the sensor ICO. Read these bits after a CTSU_CTSURD interrupt occurs. After the CTSURC counter is read, these bits are cleared immediately before the CTSU measurement status counter value changes to Status 4 (the CTSUST.CTSUSTC[2:0] flags change to 100b) in the next measurement. These bits can also be cleared using the CTSUCR0.CTSUINIT bit.

45.2.23 CTSU Reference Counter (CTSURC)

Address(es): CTSU.CTSURC 4008 101Ah

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

Bit	Symbol	Bit name	Description	R/W
b15 to b0	CTSURC[15:0]	CTSU Reference Counter	These bits indicate the measurement result of the reference ICO. These bits read FFFFh when an overflow occurs.	R

After a CTSU_CTSURD interrupt is generated, read first from the CTSUSC counter, then from the CTSURC counter. Status 3 continues until the CTSURC counter is read, even if the stabilization time specified for Status 3 elapses.

CTSURC[15:0] bits (CTSU Reference Counter)

The CTSURC[15:0] bits are configured as an increment counter for the reference ICO clock. The reference ICO optimizes touch measurement performed by the sensor ICO. There is some deviation depending on the internal sensor ICO and the reference ICO in the CTSU, but both ICOs have almost the same characteristics, including the dynamic range and the current-to-frequency characteristics. The range of current amount that can be set in the reference ICO current adjustment bits is about the same as the dynamic range of both ICOs, and the current amount input to the sensor ICO must be within this dynamic range. To ensure this, use the reference ICO to check the differences between the ICOs and measure the current-to-oscillation frequency characteristics. The reference ICO oscillation frequency can be obtained from the reference ICO counter, and the ICO oscillation frequency for the input current (counter value/ measurement time) can be measured by setting the value in the reference ICO current adjustment bits and measuring the reference ICO counter. The reference ICO counter value measured using the maximum value in the reference ICO current adjustment bits is the maximum value of the ICO dynamic range. The current to the sensor ICO must be offset in the offset adjustment bits so that the sensor ICO counter value does not exceed this value.

Read the CTSURC[15:0] bits after a CTSU_CTSURD interrupt occurs. After these bits are read, they are cleared immediately before the CTSU measurement status counter value changes to Status 4 (the CTSUST.CTSUSTC[2:0] flags change to 100b) in the next measurement. These bits can also be cleared with the CTSUCR0.CTSUINIT bit.

45.2.24 CTSU Error Status Register (CTSUERRS)

Address(es): CTSU.CTSUERRS 4008 101Ch

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

Bit	Symbol	Bit name	Description	R/W
b14 to b0	—	Reserved	These bits are read as 0.	R
b15	CTSUICOMP	TSCAP Voltage Error Monitor	This bit monitors the error status of the TSCAP voltage: 0: Normal TSCAP voltage 1: Abnormal TSCAP voltage.	R

CTSUICOMP bit (TSCAP Voltage Error Monitor)

If the offset current amount set in the CTSUSO1 register exceeds the sensor ICO input current during touch measurement, the TSCAP voltage becomes abnormal and touch measurement cannot be performed correctly. This bit monitors the TSCAP voltage and is set to 1 if the voltage becomes abnormal.

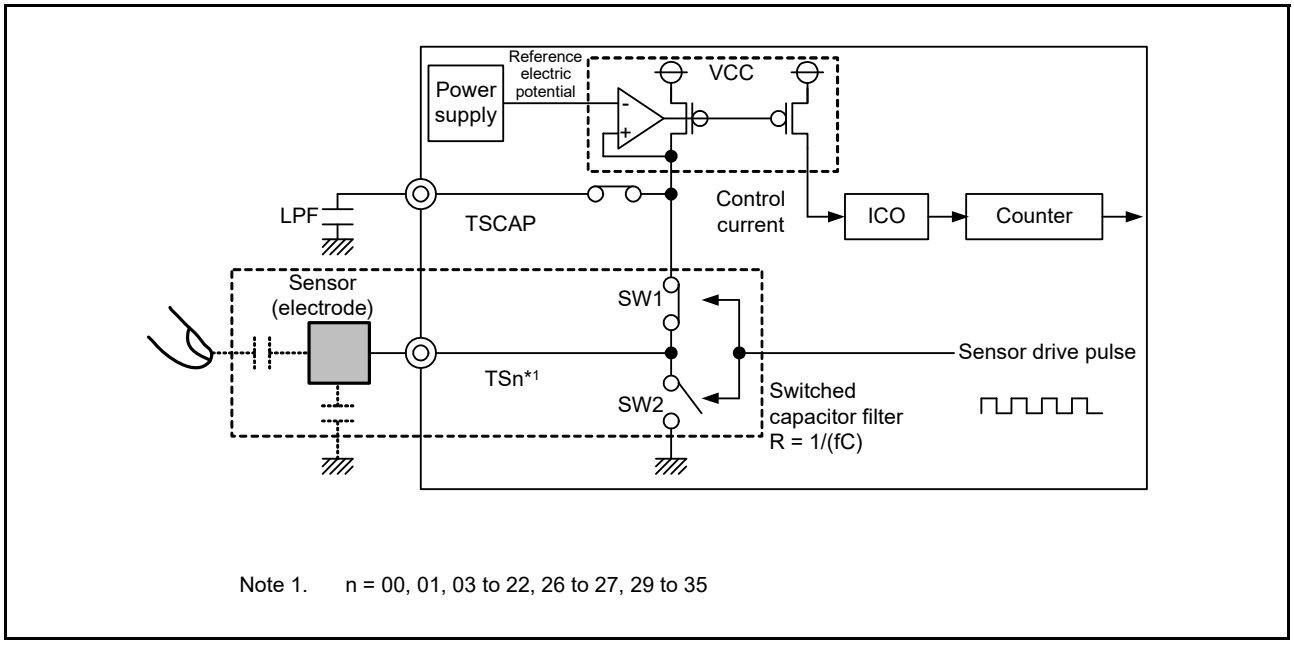
If the TSCAP voltage becomes abnormal, the sensor ICO counter value is undefined, but touch measurement completes normally, therefore it is difficult to detect an abnormality by reading the sensor ICO counter value. If the CTSU reference ICO current adjustment bits (CTSURICOA[7:0]) in the CTSUSO1 register are set to a value other than 0, check this bit when touch measurement completes.

This bit is cleared by writing 0 to the CTSUCR1.CTSUPON bit and turning off the power supply.

45.3 Operation

45.3.1 Principles of Measurement Operation

Figure 45.4 shows the measurement circuit.

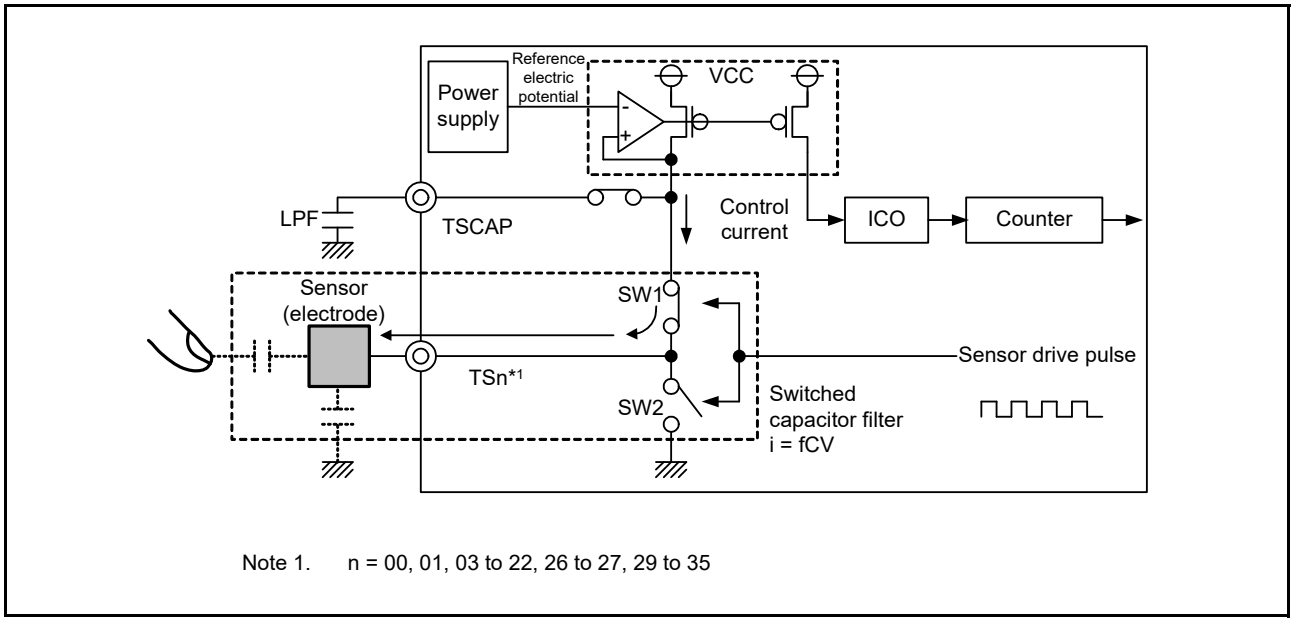


Note 1. n = 00, 01, 03 to 22, 26 to 27, 29 to 35

Figure 45.4 Measurement circuit

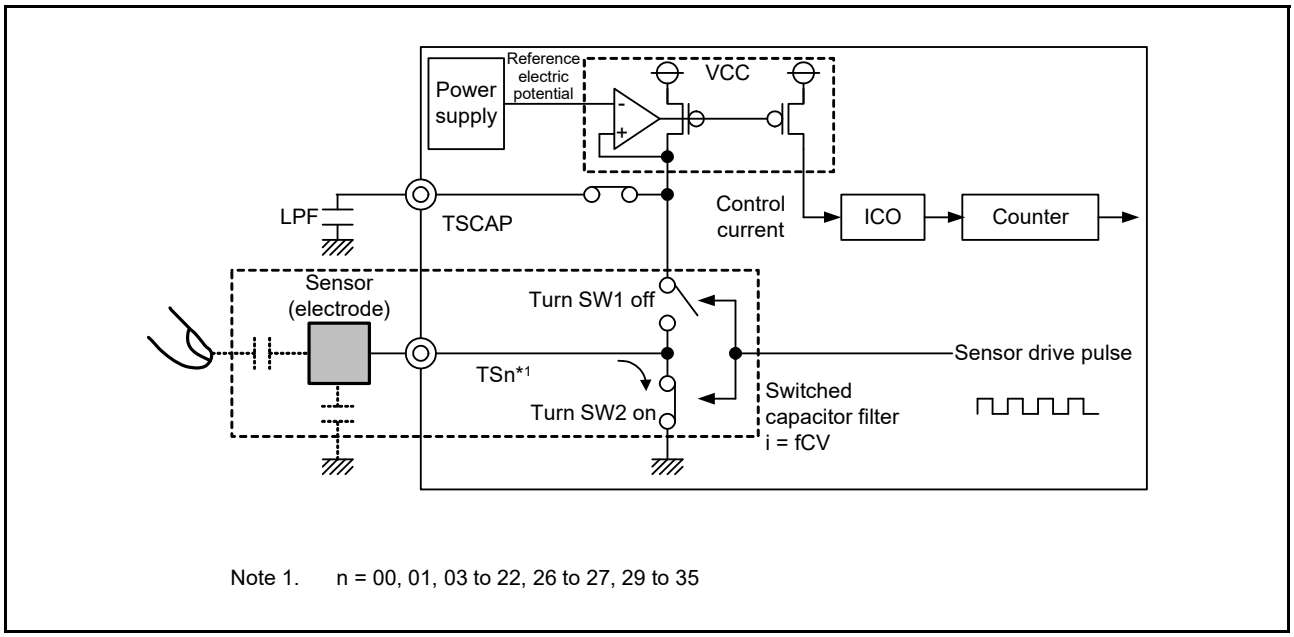
Figure 45.5 to Figure 45.7 explain the electrostatic capacitance measurement operation principles of the CTSU current frequency conversion method. The operation is as follows:

1. The electrostatic capacitance of the electrode is charged by turning SW1 on and SW2 off. See Figure 45.5.
2. The charged capacitance is discharged by turning SW1 off and SW2 on. See Figure 45.6.
3. Current flows to the switched capacitor filter by repeatedly charging and discharging the electrodes as in steps 1. and 2. At this point, if a finger is in close proximity, the capacitance and the flowing current change. A clock is generated by supplying a control current that is proportional to the amount of the current flowing through the switched capacitor filter, from the circuit that generates the TSCAP power supply to the ICO. The counter measures the clock frequency that changes depending on whether a finger is in close proximity. Software uses the value read from the counter to determine contact with a finger (Figure 45.7).



Note 1. n = 00, 01, 03 to 22, 26 to 27, 29 to 35

Figure 45.5 Charging operation



Note 1. n = 00, 01, 03 to 22, 26 to 27, 29 to 35

Figure 45.6 Discharging operation

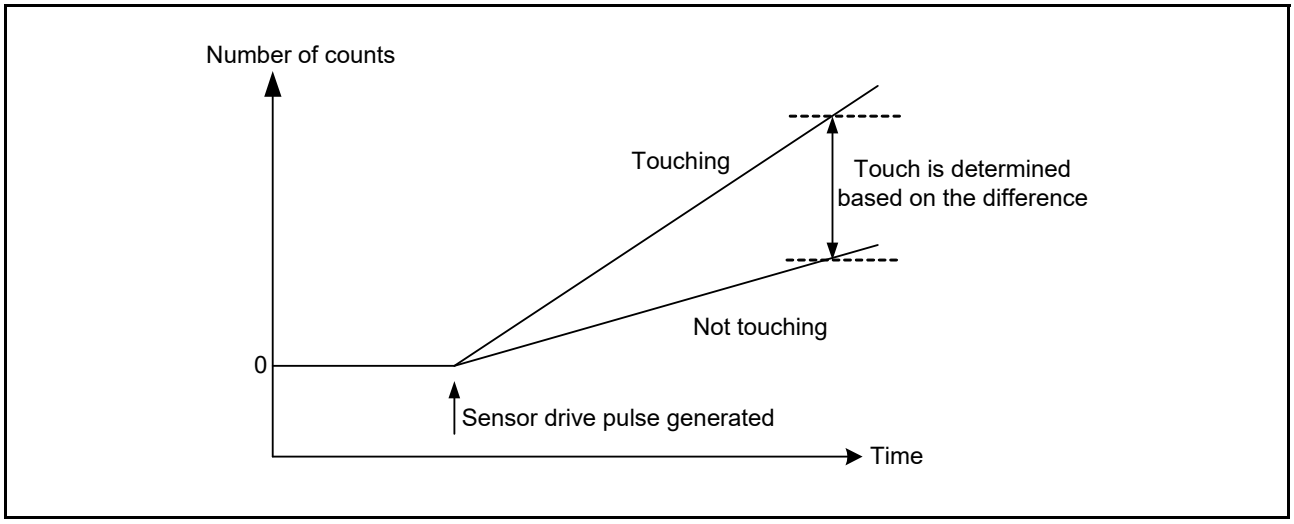


Figure 45.7 Change in measured value when finger is touching and not touching

45.3.2 Measurement Modes

The CTSU supports self-capacitance and mutual-capacitance methods. Figure 45.8 shows these methods.

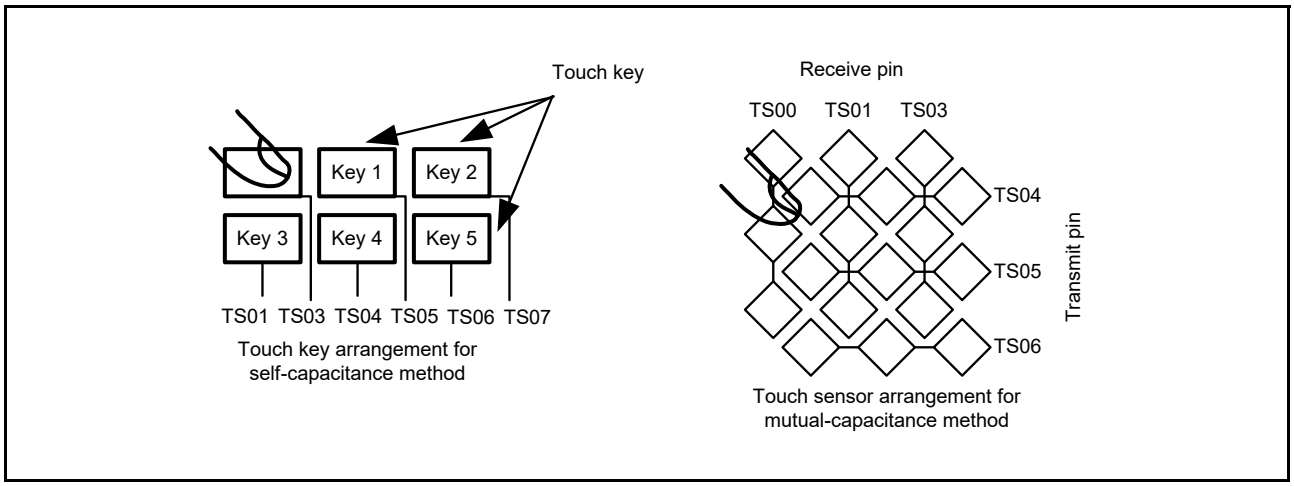


Figure 45.8 Overview of self-capacitance method and mutual-capacitance method

In the self-capacitance method, a single touch pin is allocated to a single touch key to measure individual electrostatic capacitance when a finger is in close proximity. In this method, capacitance can be measured in both single scan and multi-scan modes. In the mutual-capacitance method, the capacitance between two opposing electrodes (transmit and receive pins) is measured.

45.3.2.1 Initial Setting Flow

Figure 45.9 shows the flow for the CTSU initial settings.

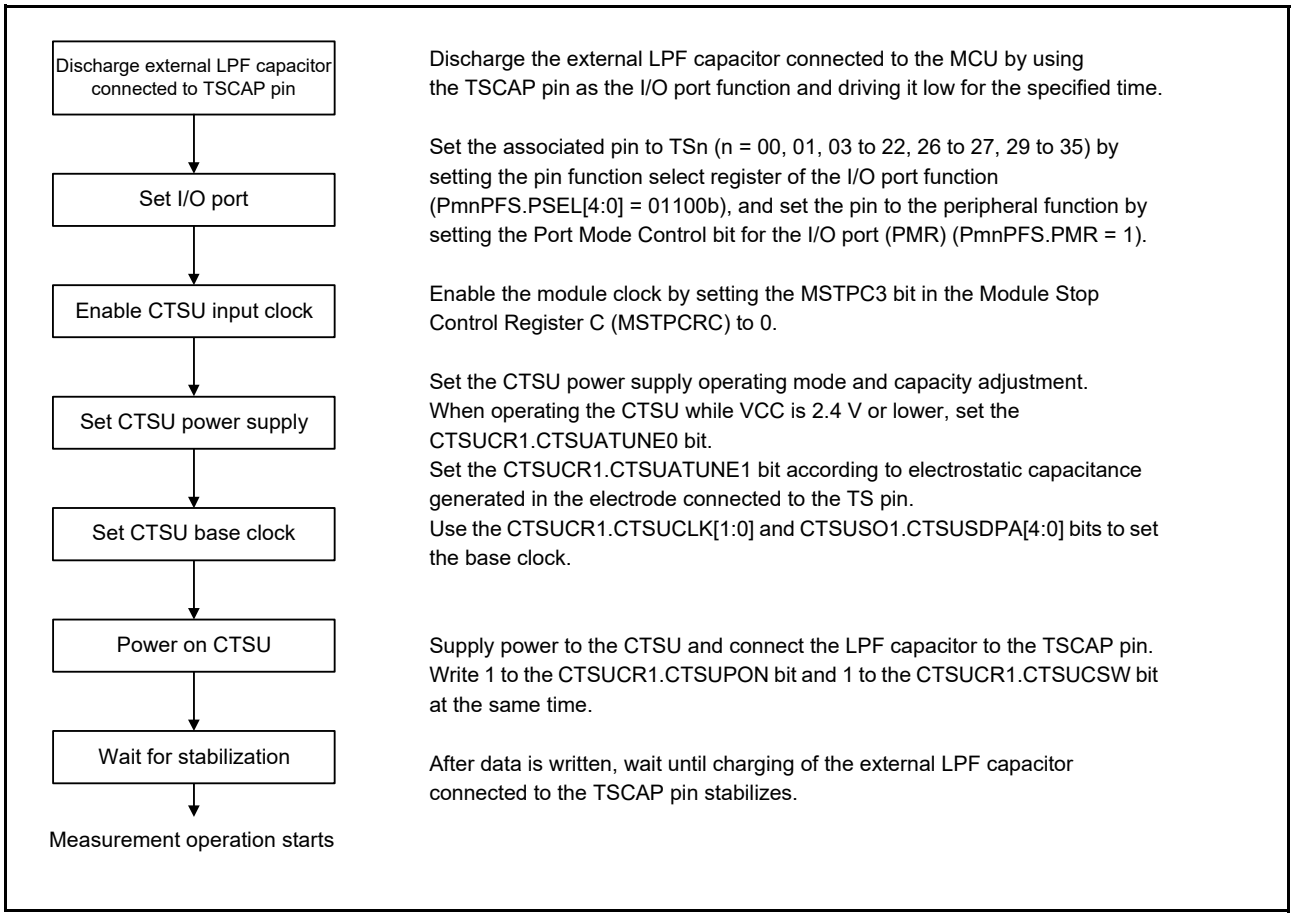


Figure 45.9 CTSU initial setting flow

Figure 45.10 shows the flow for stopping CTSU operation and invoking the standby state.

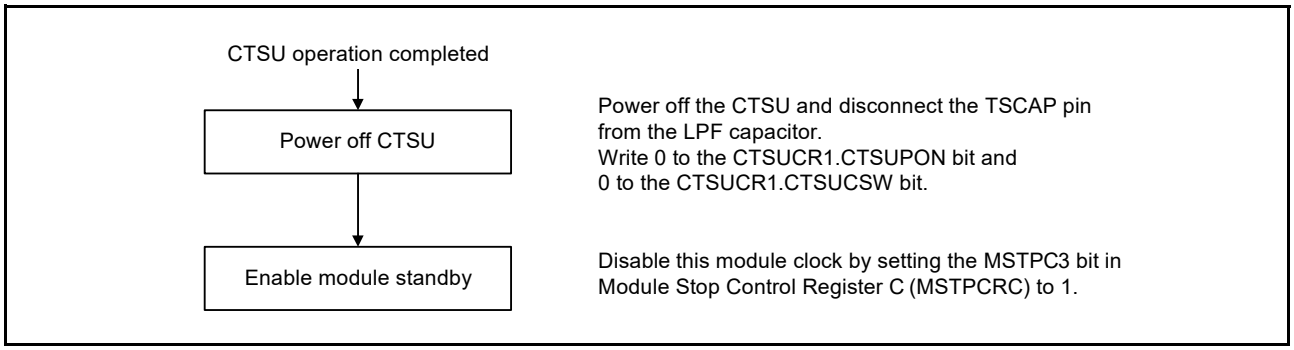


Figure 45.10 CTSU stopping flow

To restart operation, follow the initial setting flow shown in Figure 45.9.

45.3.2.2 Status Counter

The measurement status counter of the CTSU Status Register (CTSUST) indicates the current measurement status. The measurement status applies to all three modes. Figure 45.11 shows the status operation transitions.

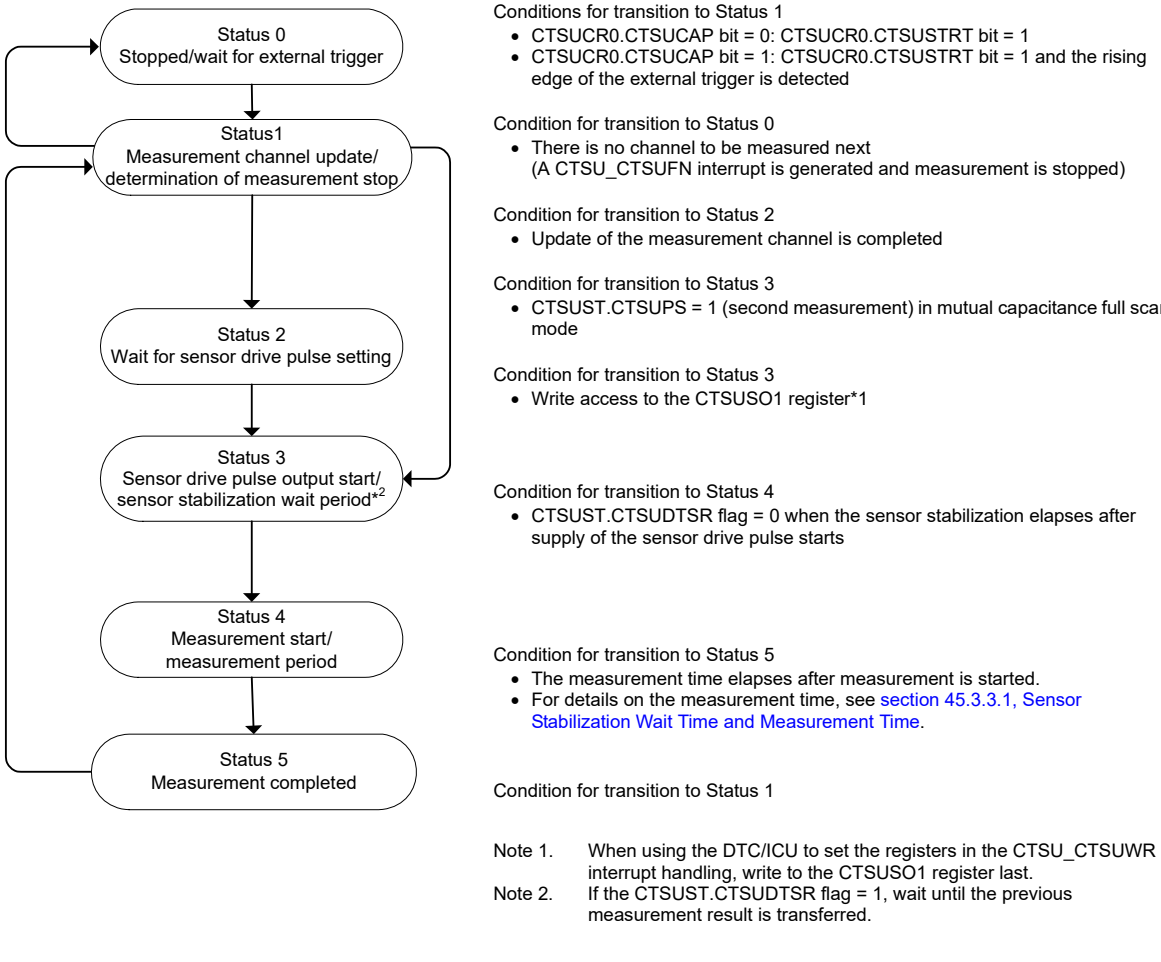


Figure 45.11 Status operation transitions

The status counter transitions to Status 0 when all of the specified measurement channels are measured.

The CTSUCR0.CTSUSTRT bit is set to 0 by hardware when a software trigger is used. When an external trigger is used, the value 1 is retained, and the CTSU waits for the next trigger.

When operation is forced to stop during measurement or the trigger wait state, by a simultaneous 0 write to the CTSUCR0.CTSUSTRT bit and a 1 write to the CTSUCR0.CTSUINIT bit. the status transitions to Status 0 and measurement stops.

In the following situations, there is no channel to be measured:

- No measurement target channel is specified in the CTSUCHAC0 to CTSUCHAC4 registers
- In self-capacitance single scan mode, the channel specified in the CTSUMCH0 register is not a measurement target in the CTSUCHAC0 to CTSUCHAC4 registers
- In full-scan modes, there is no transmit channel or receive channel to be measured based on the combined settings of the CTSUCHAC0 to CTSUCHAC4, and CTSUCHTRC0 to CTSUCHTRC4 registers.

If there is no channel to be measured based on these settings, a CTSU_CTSUFN interrupt is generated immediately after a transition to Status 1, and the counter transitions to Status 0.

45.3.2.3 Self-Capacitance Single Scan Mode Operation

In self-capacitance single scan mode, electrostatic capacitance is measured on one channel. Figure 45.12 shows the software flow and an operation example, and Figure 45.13 shows the timing.

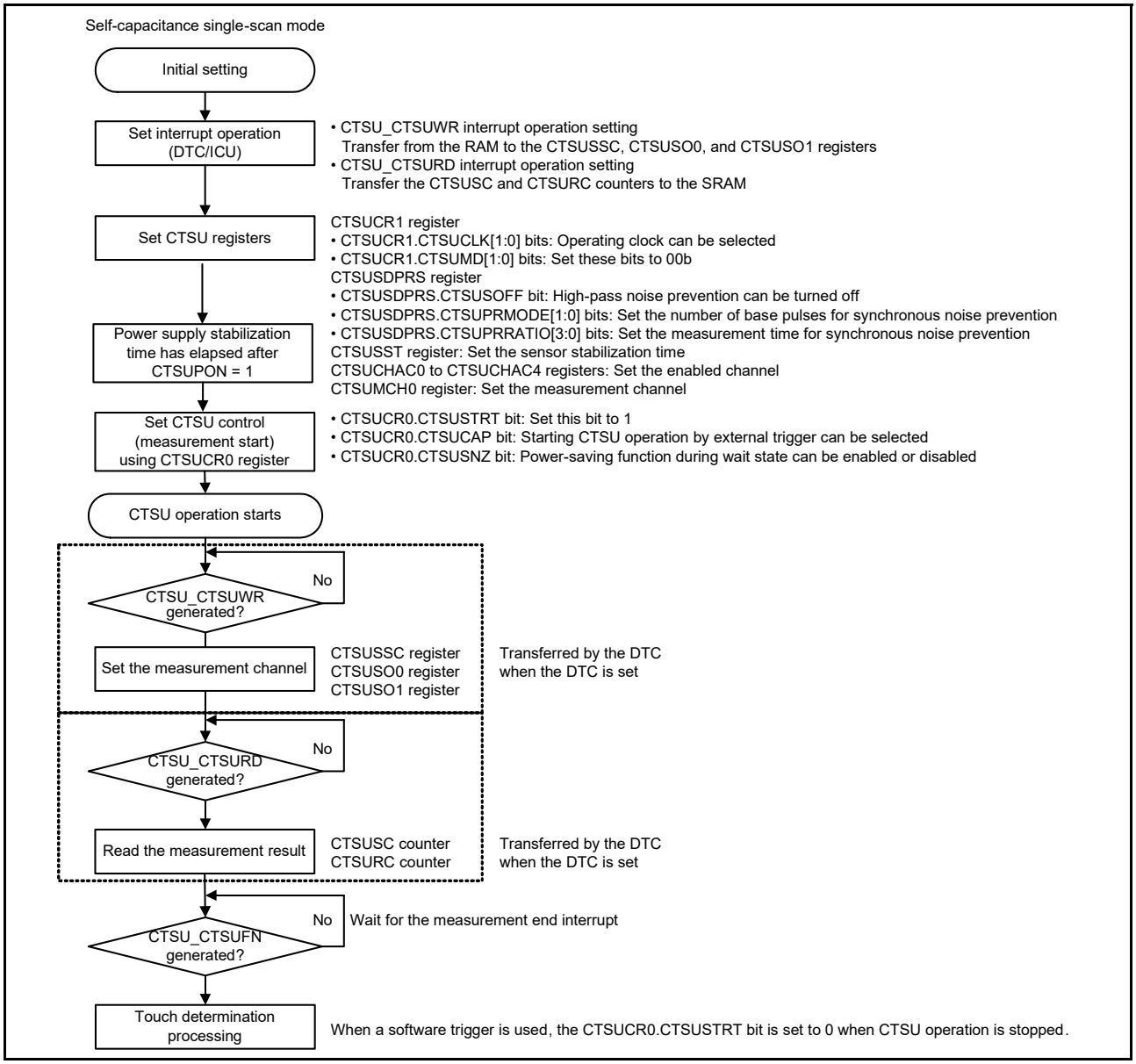


Figure 45.12 Software flow and operation example of self-capacitance single scan mode

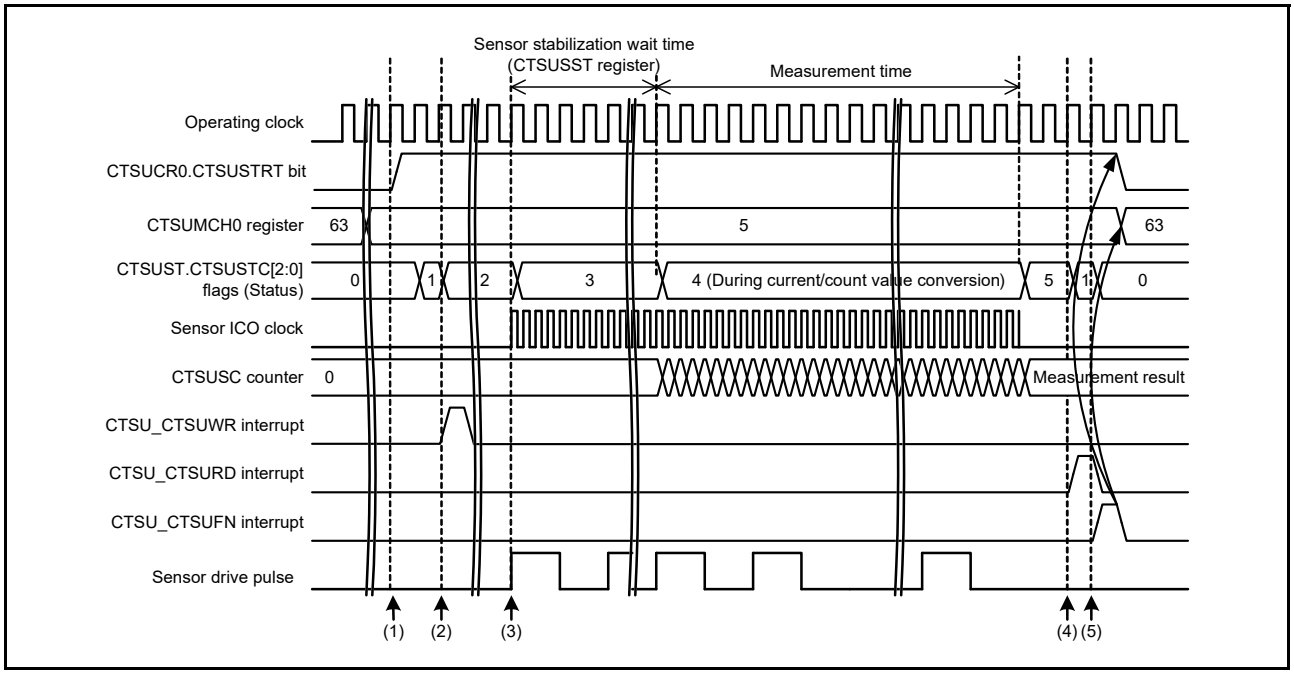


Figure 45.13 Timing of self-capacitance single scan mode when the measurement start condition is a software trigger

The following sequence describes the operation shown in Figure 45.13:

1. After initial settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.
2. After the channel to be measured is determined according to the preset conditions, a request to set the channel (CTSU_CTSUWR) is output.
3. On completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate.
4. After the sensor stabilization wait time and the measurement time elapse, and measurement stops, a measurement result read request (CTSU_CTSURD) is output.
5. A measurement end interrupt (CTSU_CTSUFN) is output and measurement stops (transition to Status 0).

Table 45.6 lists the touch pin states in self-capacitance single scan mode.

Table 45.6 Touch pin states in self-capacitance single scan mode

	Touch pin	
Status	**Measured channel**	**Non-measured channel**
0	Low	Low
1	Low	Low
2	Low	Low
3	Pulse	Low
4	Pulse	Low
5	Low	Low

45.3.2.4 Self-Capacitance Multi-scan Mode Operation

In self-capacitance multi-scan mode, electrostatic capacitance on all channels that are specified as measurement targets in the CTSUCHAC0 to CTSUCHAC4 registers is measured sequentially in ascending order. Figure 45.14 shows the software flow and an operation example, and Figure 45.15 shows the timing.

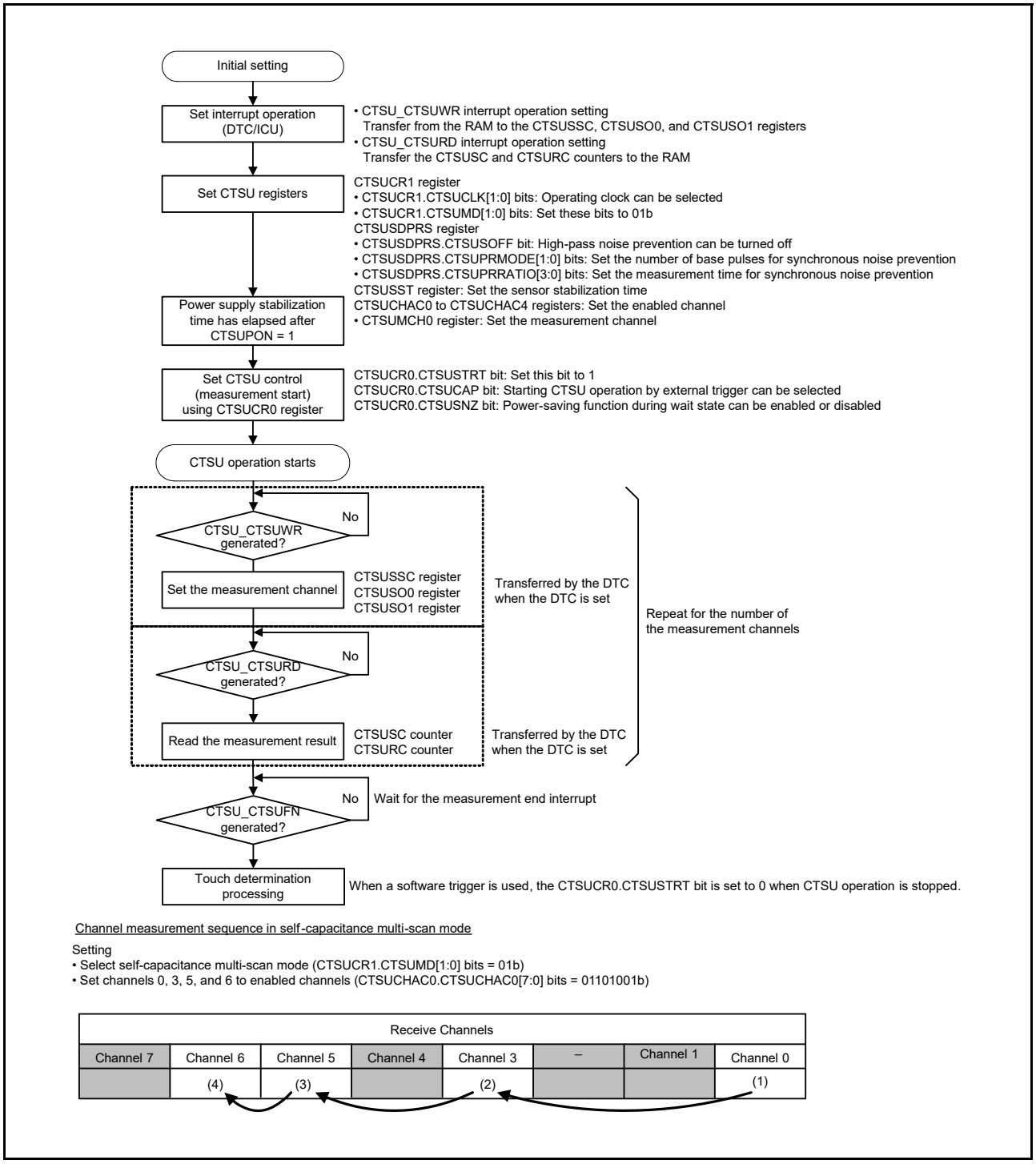


Figure 45.14 Software flow and example operation of self-capacitance multi-scan mode

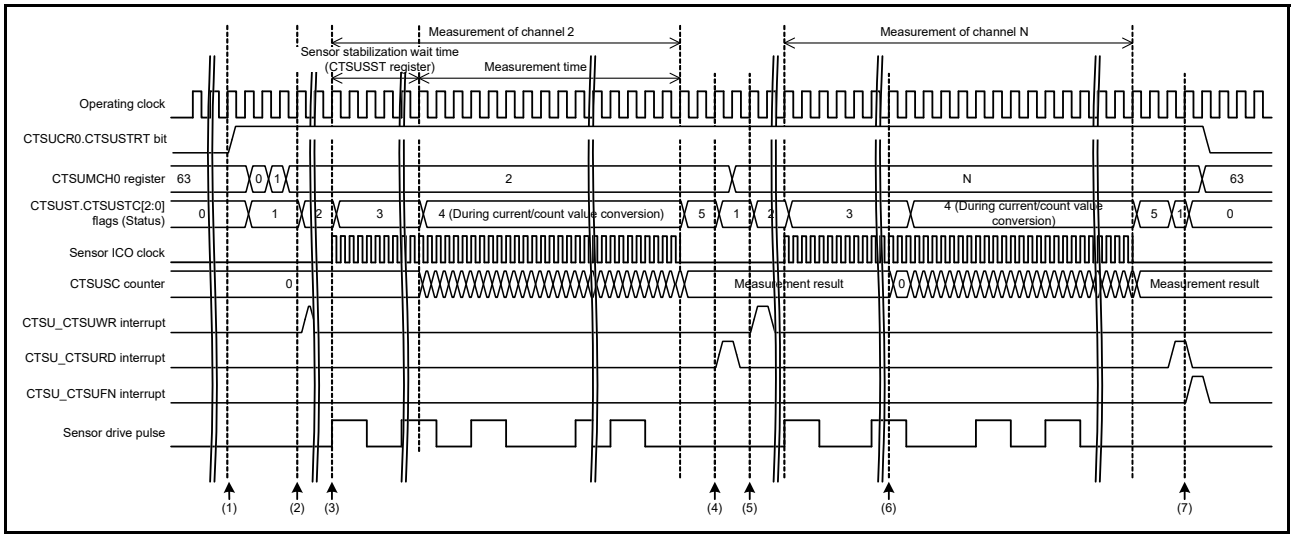


Figure 45.15 Timing of self-capacitance multi-scan mode when the measurement start condition is a software trigger

The following sequence describes the operation shown in Figure 45.15:

1. After initial settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.
2. After the channel to be measured is determined according to the preset conditions, a request to set the channel (CTSU_CTSUWR) is output.
3. On completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate.
4. After the sensor stabilization wait time and the measurement time elapse, and measurement stops, a measurement result read request (CTSU_CTSURD) is output.
5. After the channel to be measured next is determined, a request to set the channel (CTSU_CTSUWR) is output.
6. After the stabilization wait time elapses and when the previous measurement is read, the result is cleared and measurement starts.
7. On completion of all measurement channels, a measurement end interrupt (CTSU_CTSUFN) is output and measurement stops (transition to Status 0).

Table 45.7 lists the touch pin states in self-capacitance multi-scan mode.

Table 45.7 Touch pin states in self-capacitance multi-scan mode

	Touch pin	
Status	**Measured channel**	**Non-measured channel**
0	Low	Low
1	Low	Low
2	Low	Low
3	Pulse	Low
4	Pulse	Low
5	Low	Low

45.3.2.5 Mutual Capacitance Full-Scan Mode Operation

In mutual capacitance full-scan mode, measurement is performed during the high-level period of the sensor drive pulse on the receive channel by applying the edge to the target transmit channel to be measured. A single measurement target is measured twice, on the rising and falling edges. The difference between the data of these two measurements determines whether or not the electrode was touched. This creates higher touch sensitivity.

Electrostatic capacitance is measured sequentially on channels set to transmission or reception in the CTSUCHTRC0 to CTSUCHTRC4 registers, and specified as measurement targets in the CTSUCHAC0 to CTSUCHAC4 registers. The capacitance is measured by combining these signals. Figure 45.16 shows the software flow and an operation example, and Figure 45.17 shows the timing.

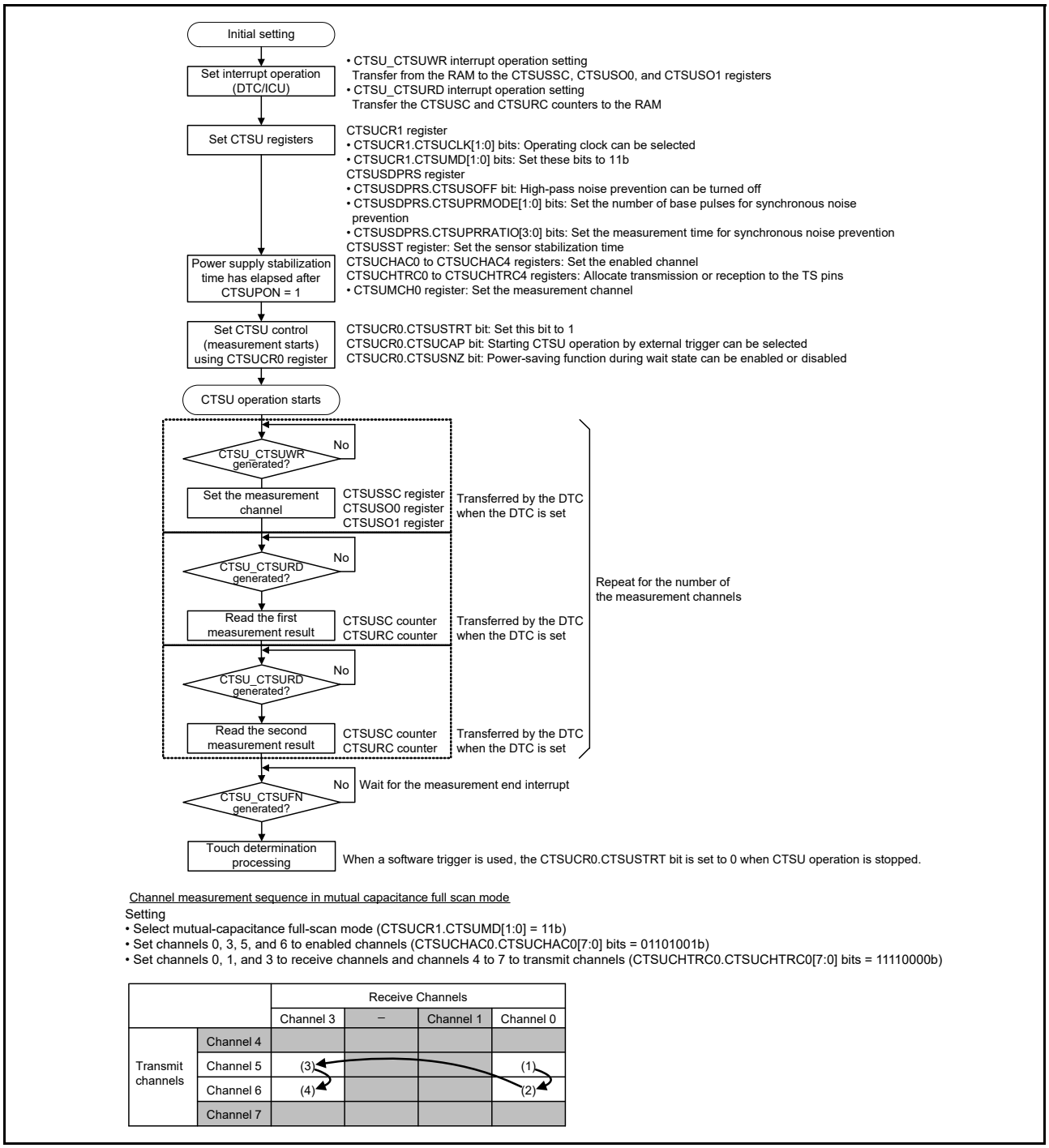


Figure 45.16 Software flow and operation example of mutual capacitance full-scan mode

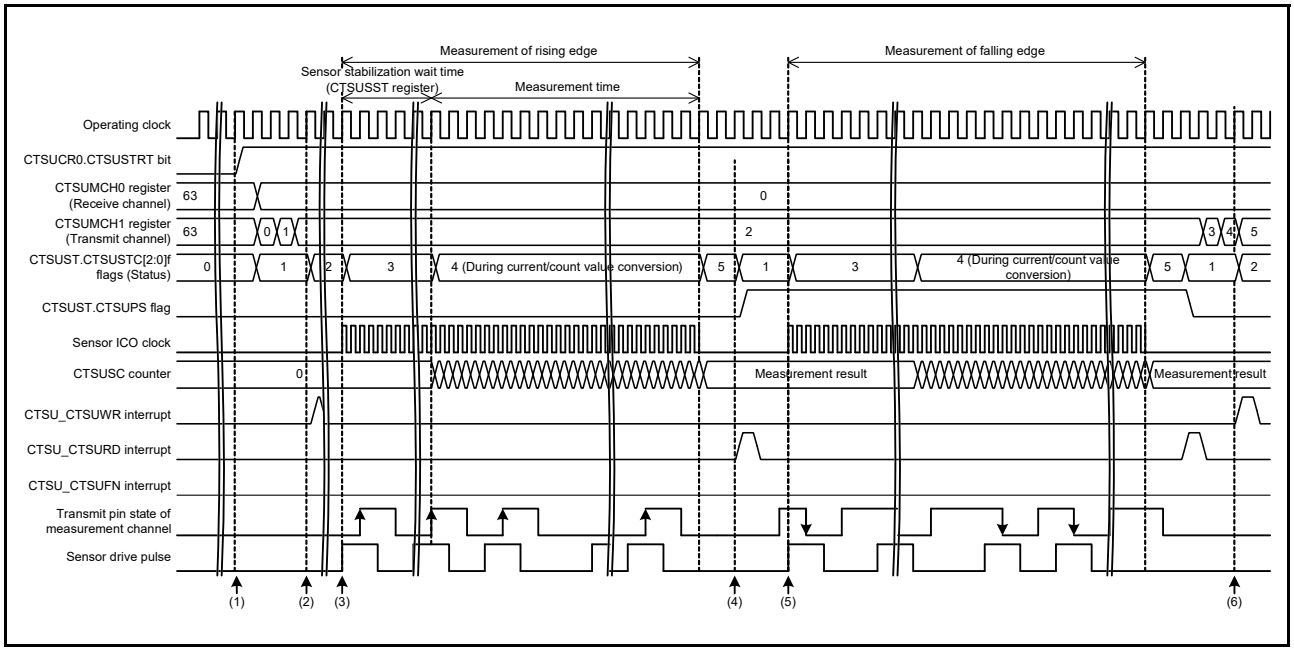


Figure 45.17 Timing of mutual capacitance full-scan mode when the measurement start condition is a software trigger

The following sequence describes the operation shown in Figure 45.17:

1. After initial settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.
2. After the channel to be measured is determined according to the preset conditions, a request for setting the channel (CTSU_CTSUWR) is output.
3. On completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate. At the same time, a pulse detected on the rising edge is output to the transmit pin on the measurement channel during the high-level period of the sensor drive pulse.
4. After the sensor stabilization wait time and the measurement time elapse and measurement stops, a measurement result read request (CTSU_CTSURD) is output.
5. The same channel is measured by outputting a pulse detected on the falling edge during the high-level period of the sensor drive pulse.
6. After the same channel is measured twice, the channel to be measured next is determined and measured in the same way.
7. On completion of all measurement channels, a measurement end interrupt (CTSU_CTSUFN) is output and measurement stops (transition to Status 0).

The CTSU mutual capacitance status flag (CTSUST.CTSUPS bit) changes when Status 5 transitions to Status 1.

Table 45.8 lists the touch pin states in mutual capacitance full-scan mode.

Table 45.8 Touch pin states in mutual capacitance full-scan mode (1 of 2)

Status	Touch pin for receive channels		Touch pin for transmit channels		Remarks
	Measured channel	Non-measured channel	Measured channel	Non-measured channel	
0	Low	Low	Low	Low	-
1	Low	Low	Low/High	Low	-
2	Low	Low	Low	Low	-

Table 45.8 Touch pin states in mutual capacitance full-scan mode (2 of 2)

Status	Touch pin for receive channels		Touch pin for transmit channels		Remarks
	Measured channel	Non-measured channel	Measured channel	Non-measured channel	
3	Pulse	Low	Pulse	Low	The phase pulse is the same as that of the receive channel on the first measurement and opposite on the second measurement
4	Pulse	Low	Pulse	Low	-
5	Low	Low	Low	Low	-

45.3.3 Functions Common to Multiple Modes

45.3.3.1 Sensor Stabilization Wait Time and Measurement Time

Figure 45.18 shows the timing of the sensor stabilization wait and measurement.

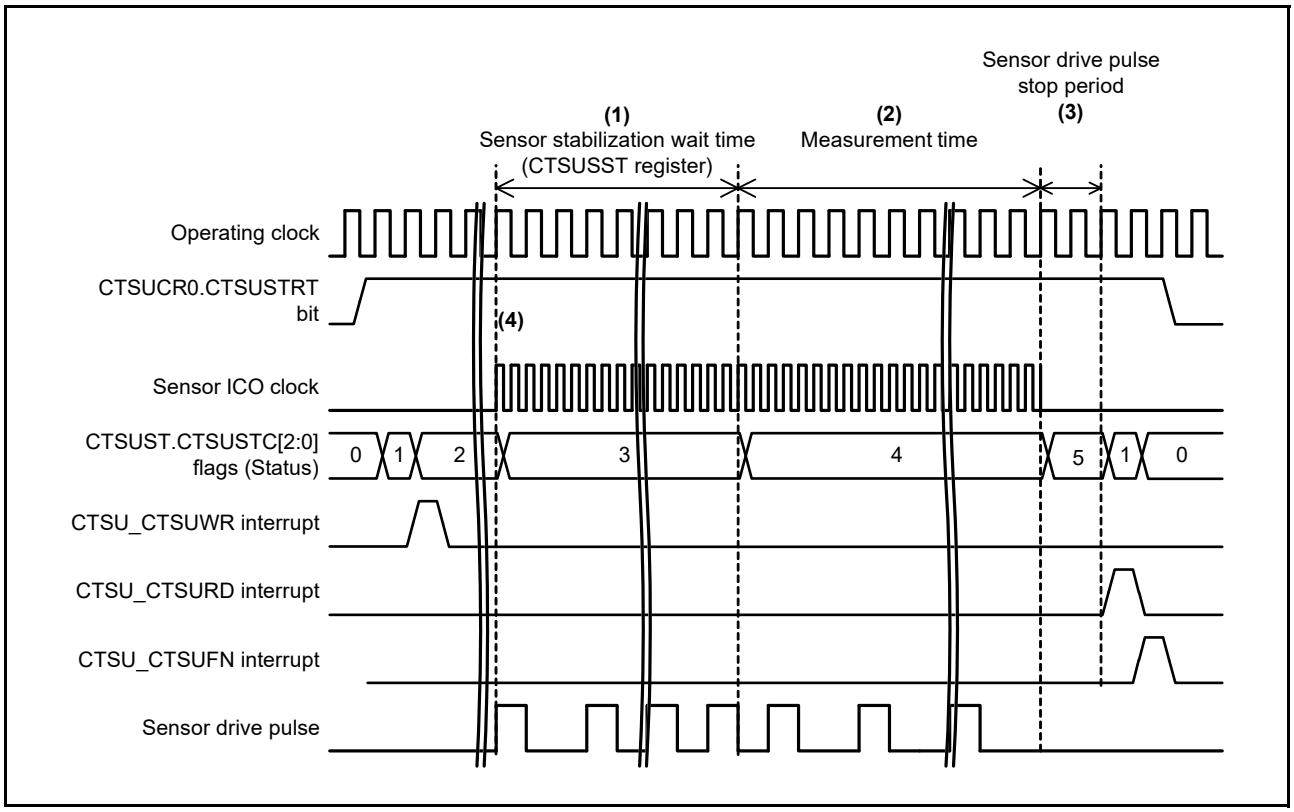


Figure 45.18 Sensor stabilization wait and measurement timing

1. In response to the CTSU_CTSUWR interrupt request, output of the sensor drive pulse is started by a write access to the CTSUSO1 register. The CTSU waits for the stabilization time set in the CTSUSST register.
2. When the sensor stabilization time elapses and the CTSUST.CTSUDTSR flag is set to 0, measurement starts on transition to Status 4. The measurement time is determined by the base clock cycle setting and the CTSUSDPRS.CTSUPRMODE[1:0], CTSUPRRATIO[3:0], and CTSUSO0.CTSUSNUM[5:0] bits. When the measurement time elapses, measurement of the channel stops.
3. After the measurement time elapses, the status transitions to Status 1 after two operating clock cycles, and a CTSU_CTSURD interrupt is generated. Read the data from the CTSUSC and CTSURC counters. At this time, the sensor drive pulse is output low. When measurement of all specified channels completes, the CTSUCR0.CTSUSTRT bit is set to 0.
4. The sensor ICO clock oscillates while the CTSUST.CTSUSTC[2:0] flags = 011b (Status 3) or 100b (Status 4).

45.3.3.2 Interrupts

The CTSU supports the following interrupts:

- Write request interrupt for setting registers for each channel (CTSU_CTSUWR)
- Measurement data transfer request interrupt (CTSU_CTSURD)
- Measurement end interrupt (CTSU_CTSUFN).

(1) Write request interrupt for setting registers for each channel (CTSU_CTSUWR)

Store the settings for each measurement channel in the SRAM, and set up the DTC or ICU transfer associated with the CTSU_CTSUWR interrupt in advance. The CTSU_CTSUWR interrupt is output when Status 1 transitions to Status 2. Write the channel settings from the SRAM to the associated CTSUSSC, CTSUSO0, and CTSUSO1 registers (Figure 45.19). Because write access to the CTSUSO1 register controls the transition to the next status, be sure to set this register last.

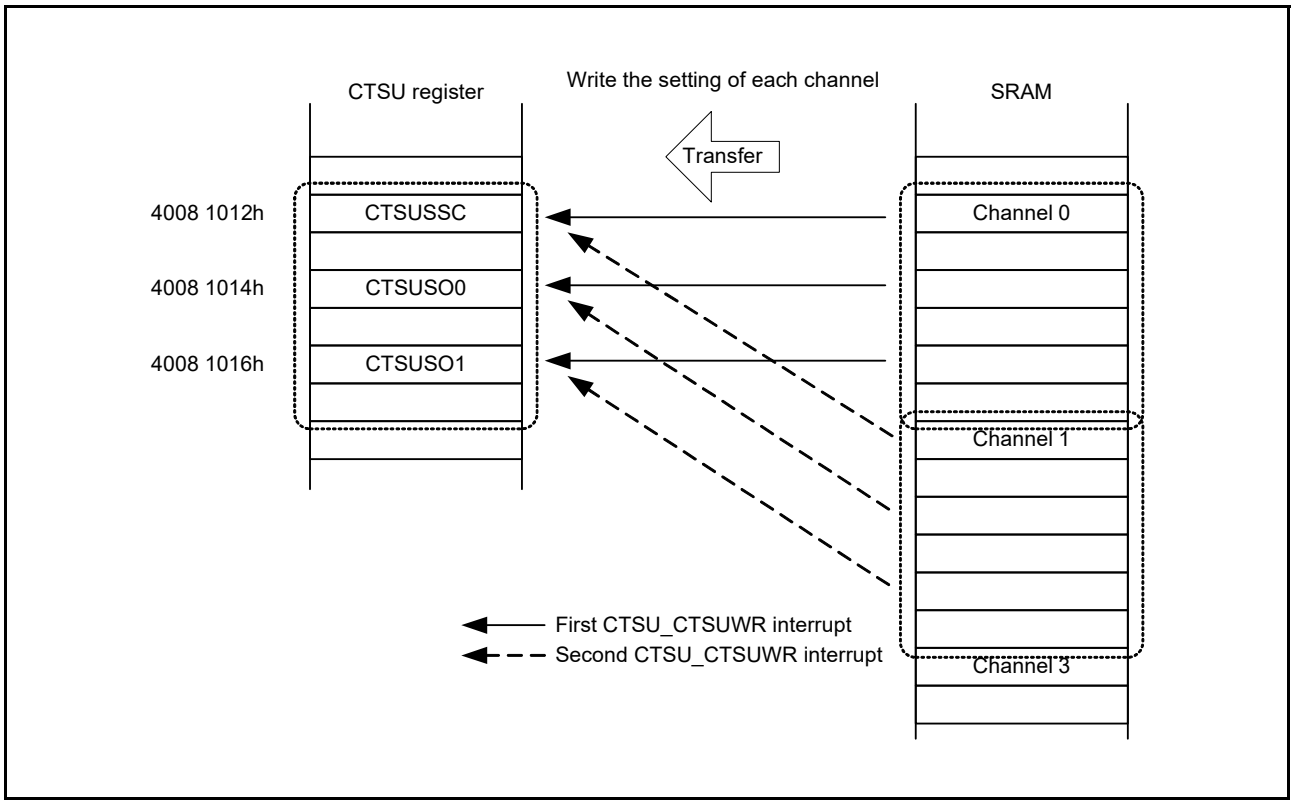


Figure 45.19 Example DTC transfer operation using the CTSU_CTSUWR interrupt

The registers to be set (CTSUSSC, CTSUSO0, and CTSUSO1) are allocated at sequential addresses. On CTSU_CTSUWR interrupt generation, set up the operation as follows:

- Transfer destination address: CTSUSSC register address
- Handling at the transfer destination address: Transfer 2-byte data three times for a single interrupt. The address of the start byte is fixed
- Transfer source address: CTSUSSC register data storage address for the lowest channel in the settings stored in the SRAM
- Handling at the transfer source address: Transfer 2-byte data three times for a single interrupt. The address of the first byte is continued from the previous interrupt handling.
- Number of transfers per interrupt: Specify the number of measurements.

(2) Measurement data transfer request interrupt (CTSU_CTSURD)

Set up the DTC or ICU transfer associated with the CTSU_CTSURD interrupt in advance. The CTSU_CTSURD interrupt is output when Status 5 transitions to Status 1. Read the measurement result from the CTSUSC and CTSURC counters in Figure 45.20.

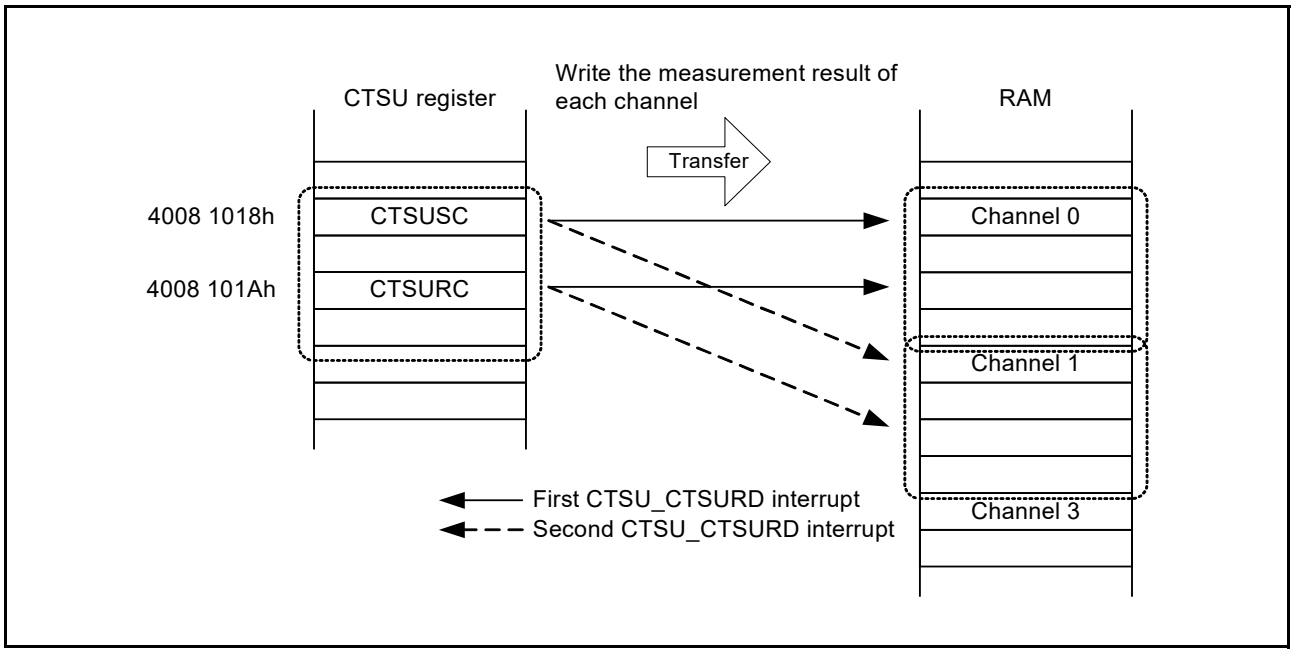


Figure 45.20 Example of DTC transfer operation using the CTSU_CTSURD interrupt

The measurement result registers (CTSUSC and CTSURC counters) used as transfer sources are allocated at sequential addresses. On CTSU_CTSURD interrupt generation, set up the operation as follows:

- Transfer source address: CTSUSC counter address
- Handling at the transfer source address: Transfer 2-byte data twice for a single interrupt. The start address is fixed.
- Transfer destination address: CTSUSC counter data storage address for the lowest number channel in the settings stored in the SRAM
- Handling at the transfer destination address: Transfer 2-byte data twice for a single interrupt. The start address is continued from the previous interrupt handling.
- Number of transfers by an interrupt: Specify the number of measurements.

(3) Measurement end interrupt (CTSU_CTSUFN)

After all channels are measured, an interrupt is generated when Status 1 transitions to Status 0. Use software to check the overflow flags (CTSUST.CTSUSOVF and CTSUROVF flags) and read the measurement results to determine whether the electrode was touched. Interrupt requests are accepted or disabled in the interrupt control block.

45.4 Usage Notes

45.4.1 Measurement Result Data (CTSUSC and CTSURC Counters)

Read access during measurement is prohibited. If the measurement result data is accessed, an incorrect value might be read because of an asynchronous operation.

45.4.2 Software Trigger

When 10b (PCLKB/4) is selected in the CTSUCR1.CTSUCLK[1:0] bits, to restart measurement by writing 1 to the CTSUR0.CTSUSTRT bit after measurement completes, wait for at least 3 cycles to elapse after an interrupt is generated, then write to the CTSUCR0.CTSUSTRT bit.

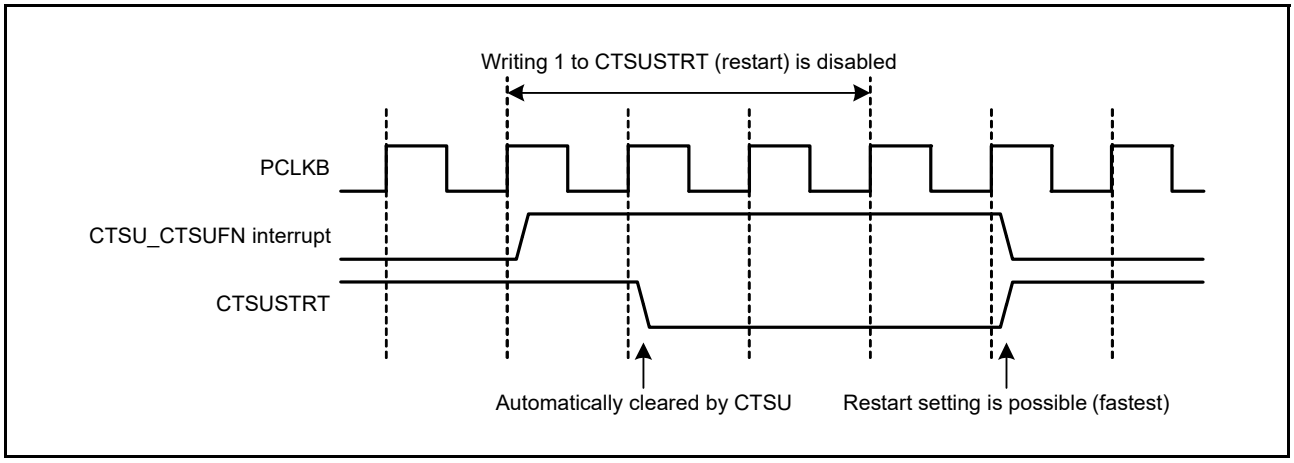


Figure 45.21 Notes on restarting measurement

45.4.3 External Trigger

- If an external trigger is input during the measurement time, measurement does not start. The next external event is enabled after 1 cycle of the operating clock when a CTSU_CTSUFN interrupt is generated.
- To stop external trigger mode, write 0 to the CTSUCR0.CTSUSTRT bit and 0 to the CTSUCR0.CTSUINIT bit at the same time (forced stop).

45.4.4 Notes on Forcing Operation Stop

To force the current operation to stop, write 0 to the CTSUCR0.CTSUSTRT bit and 1 to the CTSUCR0.CTSUINIT bit at the same time. After this setting, the operation is stopped and the internal control registers are initialized.

When the CTSUCR0.CTSUINIT bit is used for initialization, the following registers are initialized in addition to the internal measurement state:

- CTSUMCH0 register
- CTSUMCH1 register
- CTSUST register
- CTSUSC counter
- CTSURC counter.

If operation is forced to stop, an interrupt request might be generated depending on the internal state. After a forced stop, perform the processing for stopping and disabling the DTC or ICU. If a DTC transfer is stopped in an installed system for some reason, also perform the processing to force the stop and to initialize the CTSU.

45.4.5 TSCAP Pin

The TSCAP pin requires an external decoupling capacitor to stabilize the CTSU internal voltage. The traces between the TSCAP pin and the capacitor, and the capacitor and ground should be as short and wide as physically possible. The capacitor connected to the TSCAP pin should be fully discharged using the I/O port control to output low, before turning on the switch (CTSUCR1.CTSUCSW bit = 1) to establish a connection.

45.4.6 Notes on Measurement Operation (CTSUCR0.CTSUSTRT Bit = 1)

During measurement (CTSUCR0.CTSUSTRT bit = 1), do not use settings for stopping the peripheral clock or changing the port settings related to the touch pins (TSn and TSCAP pins) in the higher layers of the system.

If control settings non-compliant with these constraints are made, after operation is forced to stop (CTSUCR0.CTSUSTRT bit = 0 and CTSUCR0.CTSUINIT bit = 1), write 0 to the CTSUCR1.CTSUPON bit and 0 to the CTSUCR1.CTSUCSW bit at the same time, and set the CTSUCR0.CTSUSNZ bit to 0. Then, restart from the initial settings flow shown in Figure 45.9.

46. Data Operation Circuit (DOC)

46.1 Overview

The Data Operation Circuit (DOC) compares, adds, and subtracts 16-bit data. An interrupt can be generated when a selected condition applies.

Table 46.1 lists the DOC specifications and Figure 46.1 shows the block diagram.

Table 46.1 DOC specifications

Parameter	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 is generated on the following conditions: • The compared values either match or mismatch • The result of data addition is greater than FFFFh • The result of data subtraction is less than 0000h.

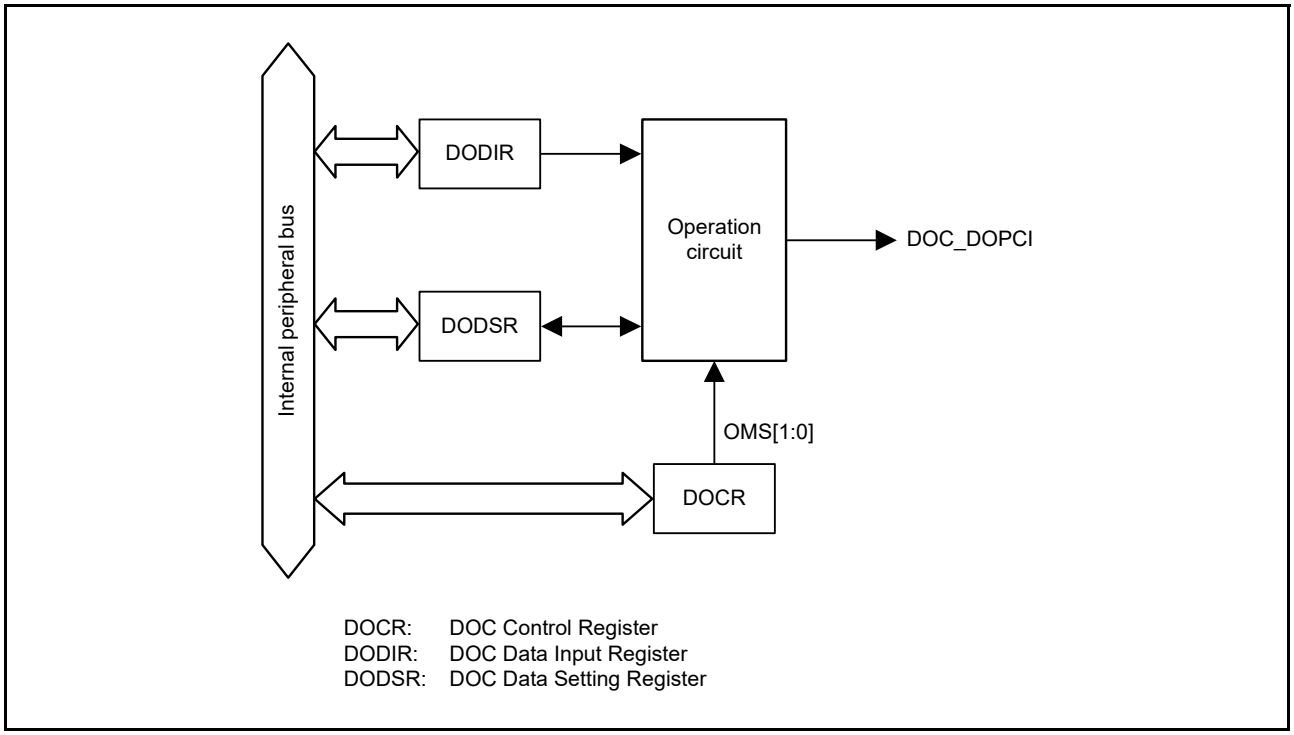


Figure 46.1 DOC block diagram

46.2 Register Descriptions

46.2.1 DOC Control Register (DOCR)

Address(es): DOC.DOCR 4005 4100h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DOPCFCL	DOPCF	—	—	DCSEL	OMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	OMS[1:0]	Operating Mode Select	b1 b0 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited.	R/W
b2	DCSEL*1	Detection Condition Select	0: Set DOPCF when data mismatch is detected 1: Set DOPCF when data match is detected.	R/W
b4, b3	—	Reserved	There bits are read as 0. The write value should be 0.	R/W
b5	DOPCF	Data Operation Circuit Flag	Indicates the result of an operation	R
b6	DOPCFCL	DOPCF Clear	0: Save DOPCF flag state 1: Clear DOPCF flag.	R/W
b7	—	Reserved	This bit is 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 (Data Operation Circuit Flag)

The DOPCF flag indicates the result of an operation.

[Setting conditions]

- The condition selected in the DCSEL bit is met
- A data addition result is greater than FFFFh
- A data subtraction result is less than 0000h.

[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.

46.2.2 DOC Data Input Register (DODIR)

Address(es): DOC.DODIR 4005 4102h

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

The DODIR is a 16-bit read/write register that stores 16-bit data used in all operations.

46.2.3 DOC Data Setting Register (DODSR)

Address(es): DOC.DODSR 4005 4104h

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

The DODSR register is a 16-bit read/write register that 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.

46.3 Operation

46.3.1 Data Comparison Mode

Figure 46.2 shows an example DOC operation in data comparison mode. In this example, the DCSE bit is set to 0, that is, when data mismatch is detected as a result of a data comparison:

1. Write 00b to the DOCR.OMS[1:0] bits to select data comparison mode.
2. Set 16-bit reference data in the DODSR register.
3. Write 16-bit data to be compared to the DODIR register.
4. Continue writing 16-bit data until all data to be compared is written to the DODIR register.
5. When DOCR.DCSEL = 0, if a value written to DODIR does not match that in DODSR, the DOCR.DOPCF flag is set to 1.

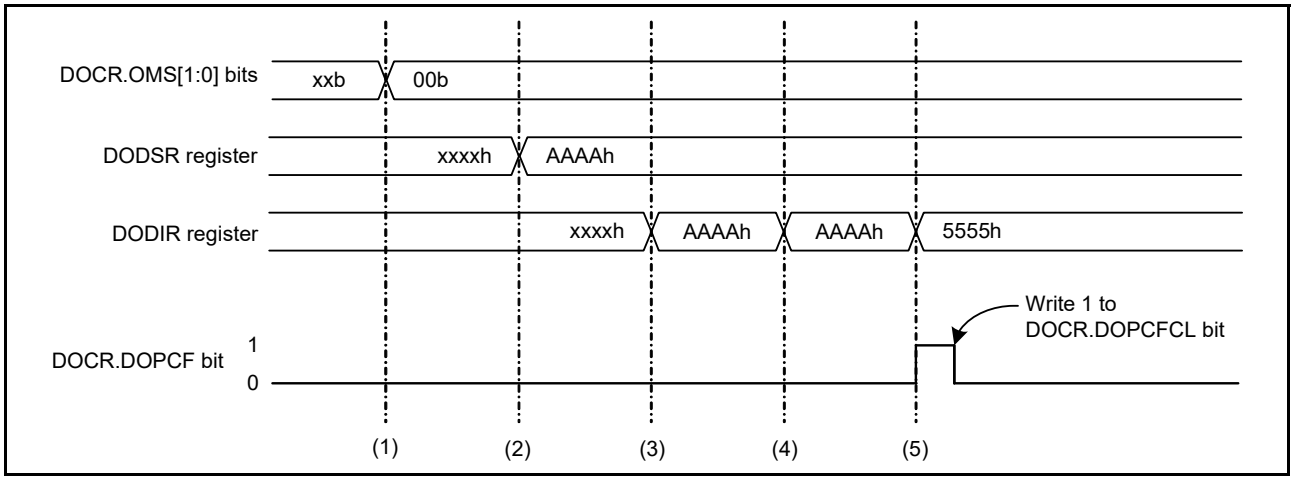


Figure 46.2 Example operation in data comparison mode

46.3.2 Data Addition Mode

Figure 46.3 shows an example DOC 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 16-bit data to the DODIR register until all data to be added is written.
5. If the result of an operation is greater than FFFFh, the DOCR.DOPCF flag is set to 1.

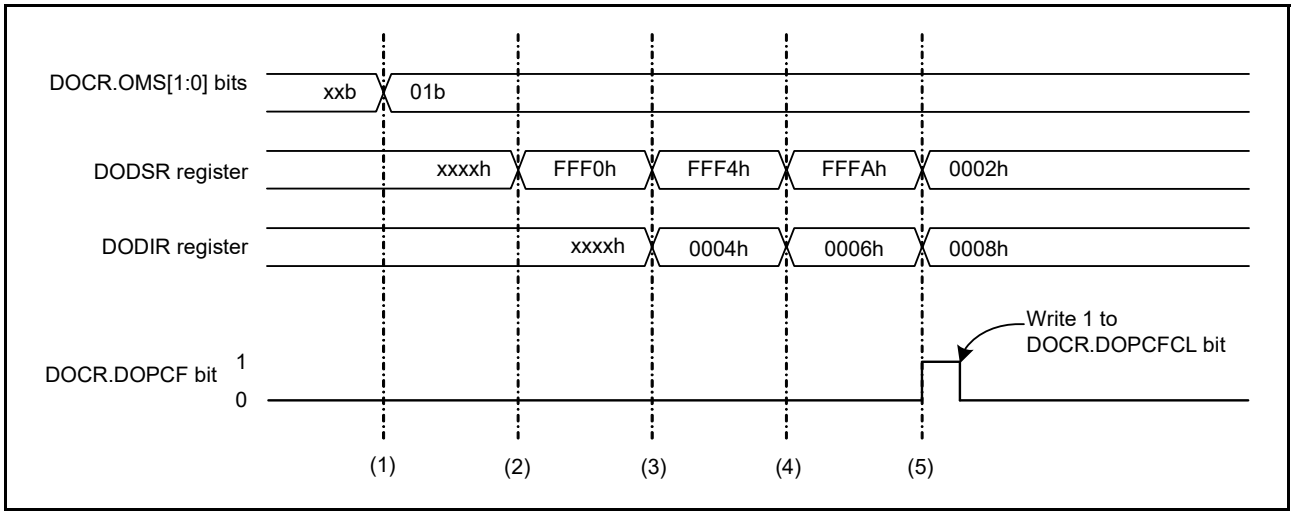


Figure 46.3 Example operation in data addition mode

46.3.3 Data Subtraction Mode

Figure 46.4 shows an example DOC 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 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 0000h, the DOCR.DOPCF flag is set to 1.

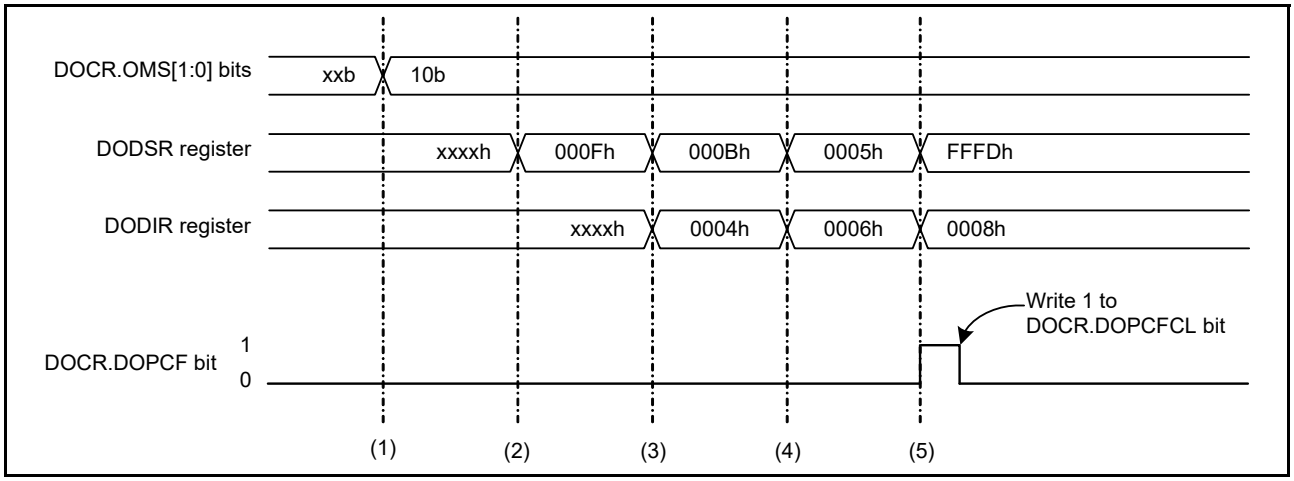


Figure 46.4 Example operation in data subtraction mode

46.4 Interrupt Request and Output to the Event Link Controller (ELC)

The DOC outputs an event signal to the ELC under the following conditions:

- The compared values either match or mismatch
- The data addition result is greater than FFFFh
- The data subtraction result is less than 0000h.

This signal can be used to initiate operations by other modules selected in advance and also can be used as an interrupt request. When an event signal is generated, the Data Operation Circuit flag (DOCR.DOPCF) is set to 1.

46.5 Usage Notes

46.5.1 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 11, Low Power Modes.

47. SRAM

47.1 Overview

The MCU provides an on-chip high-speed SRAM module with either parity-bit checking or Error Correction Code (ECC). The area of the first 16 KB of the SRAM0 is subject to ECC. Parity check is performed on other areas. Table 47.1 lists the SRAM specifications.

Table 47.1 SRAM specifications

Parameter	Without ECC	With ECC
SRAM capacity	SRAM0: 112 KB SRAM1: 64 KB	SRAM0 (ECC area): 16 KB
SRAM address	SRAM0: 2000 4000h to 2001 FFFFh SRAM1: 2002 0000h to 2002 FFFFh	SRAM0 (ECC area): 2000 0000h to 2000 3FFFh
Access*1	0 wait	
Module-stop function	Available	
Parity	Even-parity with 8-bit data and 1-bit parity	No parity
Error checking	Even-parity error check	1-bit error correction and up to 2-bit error detection

Note 1. For details, see section 47.3.7, Access Cycle.

47.2 Register Descriptions

47.2.1 SRAM Parity Error Operation After Detection Register (PARIOAD)

Address(es): SRAM.PARIOAD 4000 2000h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	OAD	Operation After Detection	1: Reset 0: Non-maskable interrupt.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

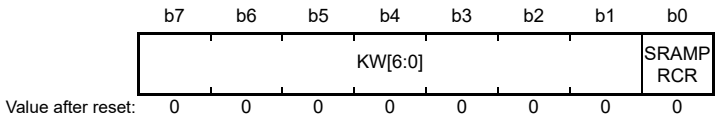
The PARIOAD register controls the operation on detection of a parity error. The SRAM Protection Register (SRAMPRCR) protects this register against writing. Set the SRAMPRCR bit in the SRAMPRCR register to the enabled setting before writing to this register. Do not write to the PARIOAD register while access to the SRAM is in progress.

OAD bit (Operation After Detection)

The OAD bit specifies generation of either a reset or non-maskable interrupt when a parity error is detected. The OAD bit in the PARIOAD register is shared by SRAM0 (without ECC) and SRAM1.

47.2.2 SRAM Protection Register (SRAMPRCR)

Address(es): SRAM.SRAMPRCR 4000 2004h



Bit	Symbol	Bit name	Description	R/W
b0	SRAMPRCR	Register Write Control	0: Disable writes to protected registers 1: Enable writes to protected registers.	R/W
b7 to b1	KW[6:0]	Write Key Code	These bits enable or disable writing to the SRAMPRCR bit	R/W

SRAMPRCR bit (Register Write Control)

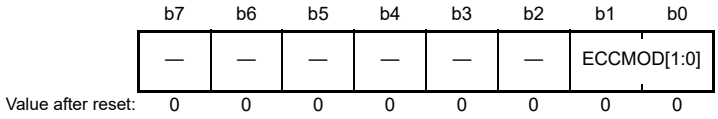
The SRAMPRCR bit controls the write mode of the PARIOAD register. When this bit is set to 1, writing to the PARIOAD register is enabled. When you write to this bit, write 78h to the KW[6:0] bits simultaneously.

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, write 78h to the KW[6:0] bits simultaneously. When a value other than 78h is written to KW[6:0], the SRAMPRCR bit is not updated. The KW[6:0] bits are always read as 00h.

47.2.3 ECC Operating Mode Control Register (ECCMODE)

Address(es): SRAM.ECCMODE 4000 20C0h



Bit	Symbol	Bit name	Description	R/W
b1, b0	ECCMOD[1:0]	ECC Operating Mode Select	b1 b0 0 0: Disable ECC function 0 1: Setting prohibited 1 0: Enable ECC function without error checking 1 1: Enable ECC function with error checking.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

The ECCMODE register specifies the ECC operating mode. The ECC Protection Register (ECCPRCR) protects this register against writing. Set the ECCPRCR bit in the ECCPRCR register to 1 before writing to this register. Do not write to the ECCMODE register while accessing the SRAM.

ECCMOD[1:0] bits (ECC Operating Mode Select)

The ECCMOD[1:0] bits set the access mode to the ECC area in SRAM.

47.2.4 ECC 2-Bit Error Status Register (ECC2STS)

Address(es): SRAM.ECC2STS 4000 20C1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ECC2ERR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ECC2ERR	ECC 2-Bit Error Status	0: No 2-bit ECC error occurred 1: 2-bit ECC error occurred.	R(/W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 0 can be written to clear the bit.

ECC2ERR bit (ECC 2-Bit Error Status)

The ECC2ERR bit indicates whether a 2-bit ECC error occurred in the ECC area of the SRAM. When ECC operations are enabled and error checking is selected, the ECC2ERR bit is set to 1 and the SRAM error signal is asserted if a 2-bit error is detected. Writing 0 to the ECC2ERR bit negates the SRAM error signal triggered by the 2-bit ECC error. The SRAM error can be specified as a non-maskable interrupt or a reset in the ECCOAD register. Do not access the ECC area in the SRAM while writing 0 to this register.

47.2.5 ECC 1-Bit Error Information Update Enable Register (ECC1STSEN)

Address(es): SRAM.ECC1STSEN 4000 20C2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	E1STSEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	E1STSEN	ECC 1-Bit Error Information Update Enable	0: Disable updating of 1-bit ECC error information 1: Enable updating of 1-bit ECC error information.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

The ECC1STSEN register enables or disables updating of the ECC 1-bit Error Status Register (ECC1STS) in response to a 1-bit ECC error in the SRAM (ECC area). The ECC Protection Register (ECCPRCR) protects this register against writing. Set the ECCPRCR bit in the ECCPRCR register to the enabled setting before writing to this bit.

E1STSEN bit (ECC 1-Bit Error Information Update Enable)

The E1STSEN bit enables or disables updating of the SRAM (ECC area) 1-Bit Error Status Register (ECC1STS) in response to a 1-bit error in the ECC area of SRAM. This register also functions as an interrupt or reset mask.

47.2.6 ECC 1-Bit Error Status Register (ECC1STS)

Address(es): SRAM.ECC1STS 4000 20C3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ECC1ERR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ECC1ERR	ECC 1-Bit Error Status	0: No 1-bit ECC error occurred 1: 1-bit ECC error occurred.	R(/W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 0 can be written to clear the bit.

ECC1ERR bit (ECC 1-Bit Error Status)

The ECC1ERR bit indicates whether a 1-bit ECC error occurred in the ECC area of the SRAM. When ECC operations are enabled and error correction is selected, and updating of the 1-bit error information is enabled, this bit is set to 1 and the SRAM error signal is asserted if a 1-bit error is detected. Writing 0 to the ECC1ERR bit negates the SRAM error signal triggered by the 1-bit ECC error.

The SRAM error can be specified as a non-maskable interrupt or a reset in the ECCOAD register. Do not access the ECC area in the SRAM while writing 0 to this register.

47.2.7 ECC Protection Register (ECCPRCR)

Address(es): SRAM.ECCPRCR 4000 20C4h

	b7	b6	b5	b4	b3	b2	b1	b0
	KW[6:0]							ECCPRCR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ECCPRCR	Register Write Control	0: Disable writes to the protected registers 1: Enable writes to the protected registers.	R/W
b7 to b1	KW[6:0]	Write Key Code	These bits enable or disable writes to the ECCPRCR bit.	R/W

ECCPRCR bit (Register Write Control)

The ECCPRCR bit controls the write mode of the ECCMODE, ECC1STSEN, and ECCOAD registers. When this bit is set to 1, writing to the ECCMODE, ECC1STSEN, and ECCOAD registers is enabled. When you write to this bit, write 78h to the KW[6:0] bits simultaneously.

KW[6:0] bits (Write Key Code)

The KW[6:0] bits enable or disable writes to the ECCPRCR bit. When you write to the ECCPRCR bit, write 78h to the KW[6:0] bits simultaneously. When a value other than 78h is written to KW[6:0], the ECCPRCR bit is not updated. The KW[6:0] bits are always read as 00h.

47.2.8 ECC Protection Register 2 (ECCPRCR2)

Address(es): SRAM.ECCPRCR2 4000 20D0h

	b7	b6	b5	b4	b3	b2	b1	b0
	KW2[6:0]							ECCPRCR2
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	ECCPRCR2	Register Write Control	0: Disable writes to the protected registers 1: Enable writes to the protected registers.	R/W
b7 to b1	KW2[6:0]	Write Key Code	These bits enable or disable writes to the ECCPRCR2 bit.	R/W

ECCPRCR2 bit (Register Write Control)

The ECCPRCR2 bit controls the write mode of the ECCETST register. When the ECCPRCR2 bit is set to 1, writes to the ECCETST register is enabled. When you write to this bit, write 78h to the KW2[6:0] bits simultaneously.

KW2[6:0] bits (Write Key Code)

The KW2[6:0] bits enable or disable writing of the ECCPRCR2 bit. When you write to the ECCPRCR2 bit, write 78h to KW2[6:0] simultaneously. When a value other than 78h is written to KW2[6:0], the ECCPRCR2 bit is not updated. The KW2[6:0] bits are always read as 00h.

47.2.9 ECC Test Control Register (ECCETST)

Address(es): SRAM.ECCETST 4000 20D4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TSTBYP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSTBYP	ECC Bypass Select	0: Disable ECC bypass 1: Enable ECC bypass.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

The ECC Protection Register (ECCPRCR2) protects this register against writing. Enable the ECCPRCR2 bit in the ECCPRCR2 register before writing to this bit. Do not write to the ECCETST register while accessing the SRAM.

TSTBYP bit (ECC Bypass Select)

The TSTBYP bit enables direct access to the ECC code by bypassing the ECC function. The ECC bypass function is used when the ECCMOD[1:0] bits in the ECCMODE register are set to 00b. Access the same address with 32-bit access size as the data that is checked by ECC. The lower 7 bits of 32-bit write data can be written as an ECC code when the ECC bypass is enabled. The upper 25 bits in the write data are ignored. The lower 7 bits of the 32-bit read data can be used as ECC code. The upper 25 bits in the read data are unknown.

Note: For details of ECC test, see section 47.3.4, ECC Decoder Testing.

47.2.10 SRAM ECC Error Operation After Detection Register (ECCOAD)

Address(es): SRAM.ECCOAD 4000 20D8h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	OAD	Operation After Detection	1: Reset 0: Non-maskable interrupt.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

The ECC Protection Register (ECCPRCR) protects this register against writing. Enable the ECCPRCR bit in the ECCPRCR register before writing to this bit. Do not write to the ECCOAD register while accessing the SRAM.

OAD bit (Operation After Detection)

The OAD bit selects whether to generate a reset or a non-maskable interrupt when an ECC error is detected. The OAD bit in the ECCOAD register is used for the SRAM (ECC area).

47.3 Operation

47.3.1 Low Power Consumption Function

Power consumption can be reduced by setting the Module Stop Control Register A (MSTPCRA) to stop supply of the clock signal to the SRAM. The control bits are as follows for each module:

- Setting both the MSTPA0 and the MSTPA6 bits in MSTPCRA to 1 stops supply of the clock signal to SRAM0*1
- Setting the MSTPA1 bit in MSTPCRA to 1 stops supply of the clock signal to SRAM1.

Note 1. The settings in the MSTPA0 and MSTPA6 bits in the MSTPCRA register must be the same.

Stopping the clock signal supply places the SRAM in the module-stop state. The SRAM is not accessible in the module-stop state. Do not transition to the module-stop state while access to the SRAM is in progress. Access to the SRAM in the module-stop state is prohibited. If access is attempted, correct operation is not guaranteed.

For details on the MSTPCRA register, see section 11, Low Power Modes.

Power consumption can be further reduced in Software Standby mode as the supply voltage for SRAM0 can be off except for the 48 KB in the head area of SRAM0 (2000 0000h to 2000 BFFFh). For details on Software Standby mode, see section 11, Low Power Modes.

47.3.2 ECC Function

The ECC function can be enabled or disabled through the ECCMODE register setting. In the initial state, the ECC function is disabled. The ECC provides both Single-Error Correction (SEC) and Double-Error Detection (DED) functionality. When the ECC function is enabled, 7 check bits are appended to 32-bit data for writing. For reading, 39-bit data (32-bit data and 7 check bits) is read from the SRAM (ECC area).

When the ECC function and error checking are both enabled, an error correction is performed if a 1-bit error occurs, and the ECC1ERR bit in the ECC1STS register is set to 1 if the E1STSEN bit in the ECC1STSEN register is 1. If a 2-bit error occurs, the error is detected and the ECC2ERR bit in the ECC2STS register is set to 1. However, error correction is not performed.

When the ECC function is enabled and the error checking is disabled, error correction is performed if a 1-bit error occurs but ECC1ERR bit in the ECC1STS register is not updated even if the E1STSEN bit in the ECC1STSEN register is 1. If a 2-bit error occurs, the error is detected but the ECC2ERR bit in the ECC2STS register is not updated, and error correction is not performed.

When the ECC function is disabled, neither error correction nor error detection is performed even when a 1-bit or 2-bit error occurs. Therefore, the ECC1ERR and ECC2ERR bits are not updated.

It is not possible to confirm the location where the error is detected. Therefore, after the occurrence of an error, update all the data by writing 32-bit data to the SRAM.

When a read access is executed consecutively after a write access, read access is executed with priority. Therefore, during initialization, do not perform a read access successively after a write access.

47.3.3 ECC Error Generation

When the ECC function is enabled and error checking is applied to the SRAM (ECC area), an ECC error occurs when the ECC2ERR bit in the ECC2STS register becomes 1 to indicate a 2-bit error, or the ECC1ERR bit in the ECC1STS register becomes 1 to indicate a 1-bit error, through ECC checking.

To mask ECC 1-bit errors, set the ECC1STSEN.E1STSEN bit to 0 to disable ECC1ERR bit update. An ECC error is not generated when the ECC function is either disabled or enabled, and error checking is not selected.

An ECC error can generate either a non-maskable interrupt or a reset, as selected in the ECCOAD register. When the OAD bit in the ECCOAD register is set to 1, an ECC error is output to the reset function. When the OAD bit in the ECCOAD register is set to 0, an ECC error is output to the ICU as a non-maskable interrupt.

47.3.4 ECC Decoder Testing

Figure 47.1 shows the ECC decoder testing.

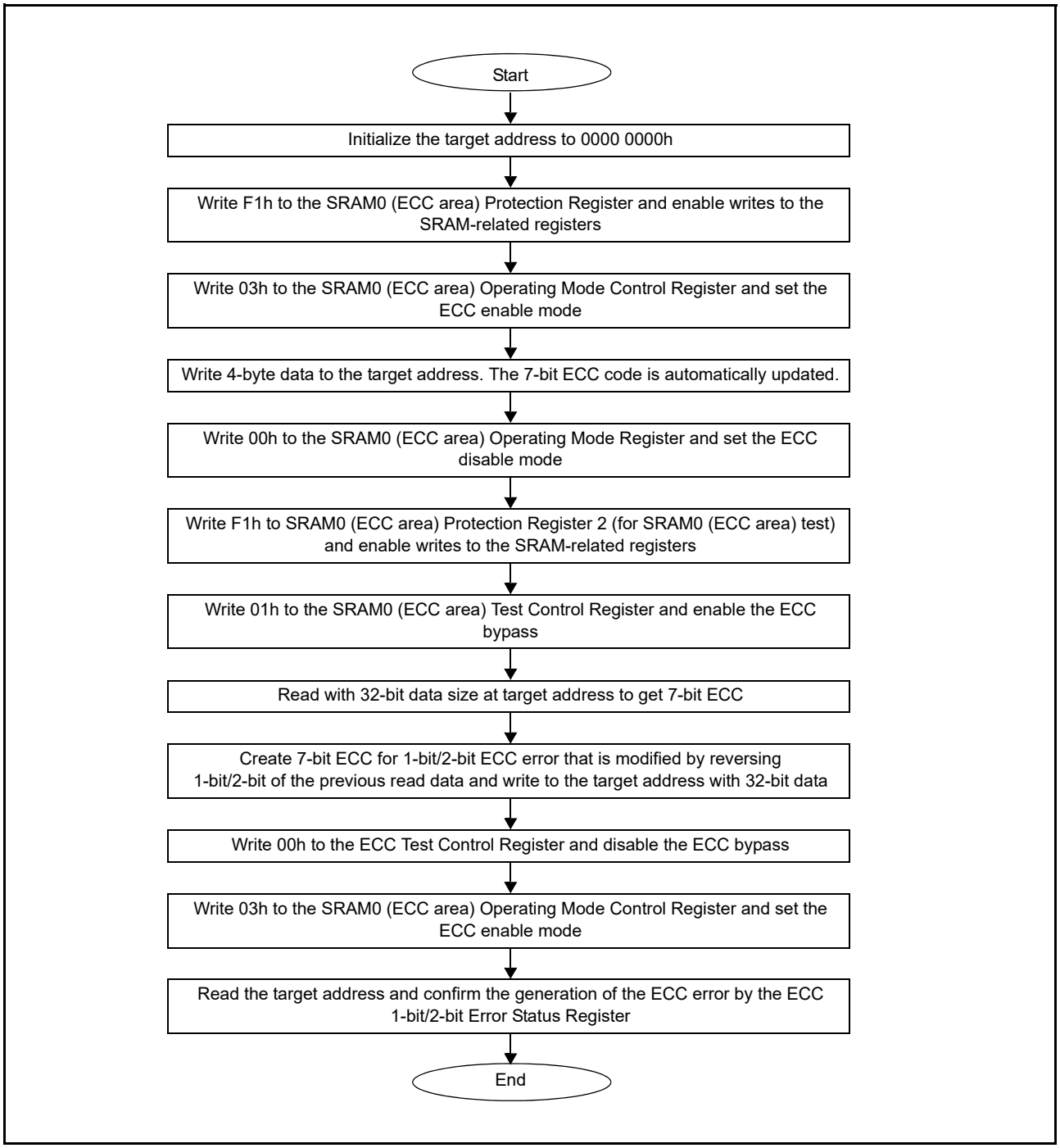


Figure 47.1 ECC decoder testing

47.3.5 Parity Calculation Function

The IEC60730 standard requires checking the SRAM data. When data is written, the function adds a parity bit to all 8-bit data in the SRAM which has 32-bit data width, and when the data is read, the parity is checked. When a parity error occurs, a parity-error notification is generated. This function can also trigger a reset. The specification of SRAM0 without ECC, and SRAM1 is even parity.

The parity-error notification can be specified as a non-maskable interrupt or a reset in the OAD bit in 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 often occur because of noise. To check whether the cause of the parity error is noise or damage, follow the parity check flows shown in Figure 47.2 and Figure 47.3.

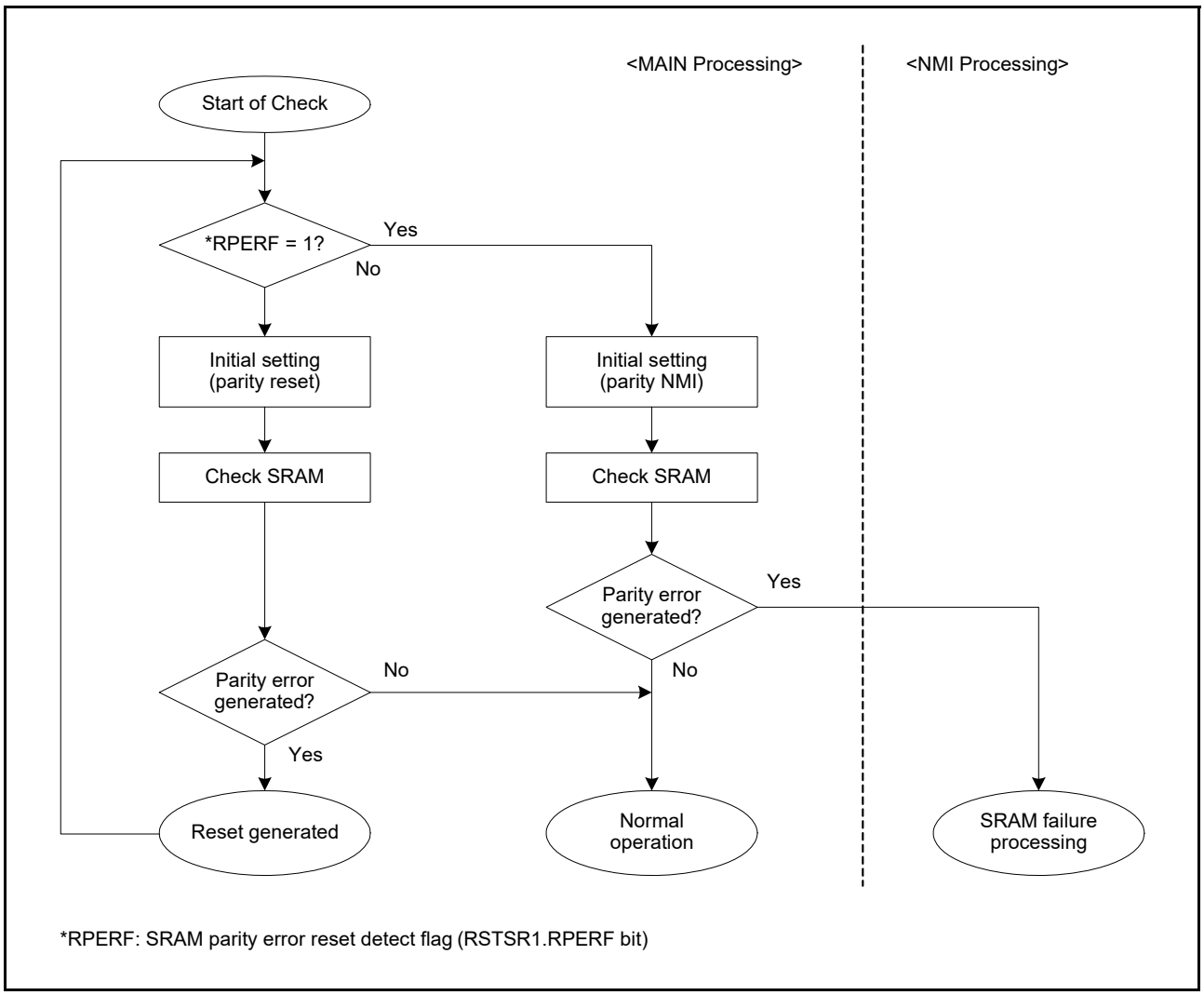


Figure 47.2 Flow of SRAM parity check when SRAM parity reset is enabled

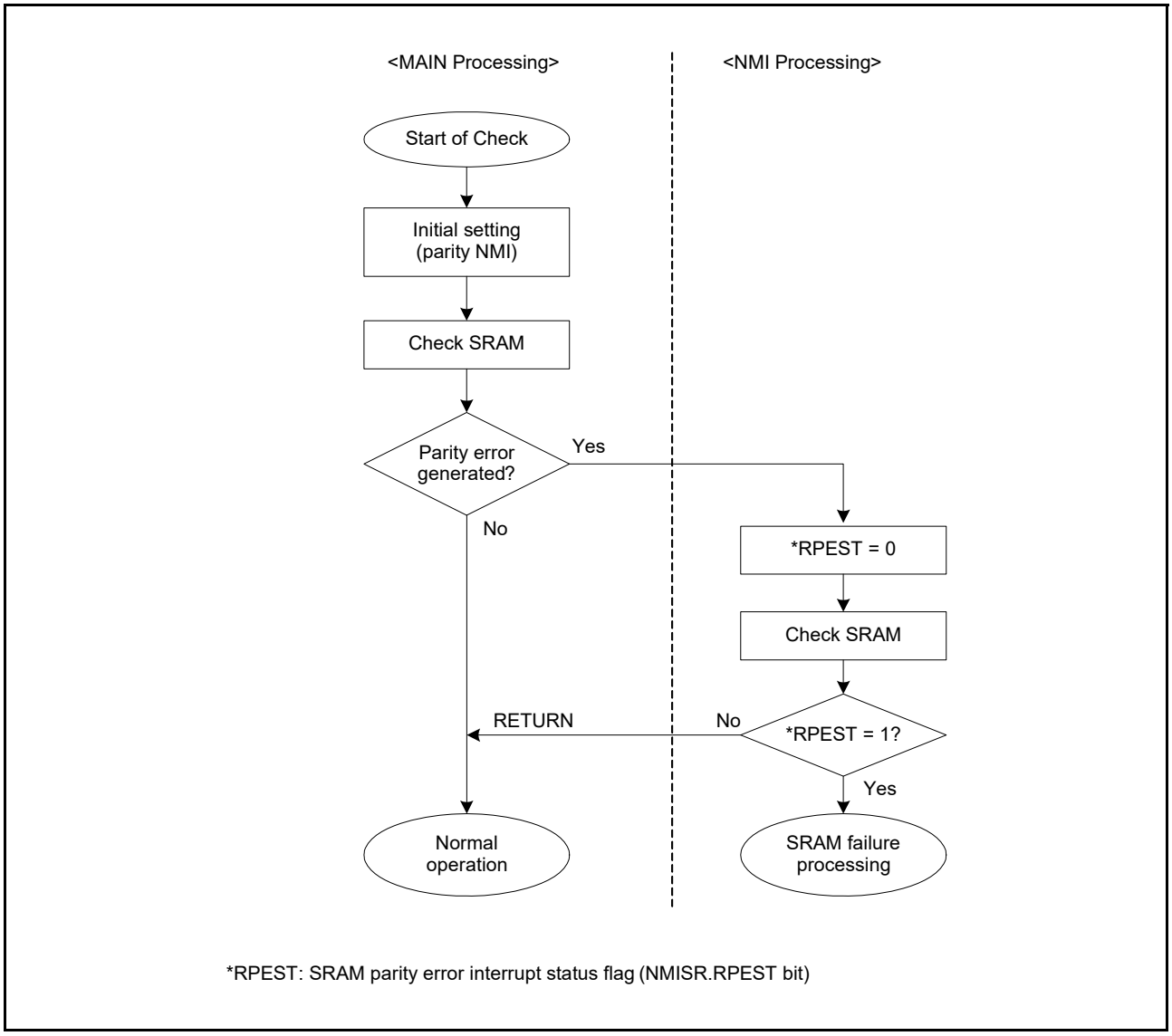


Figure 47.3 Flow of SRAM parity check when SRAM parity interrupt is enabled

47.3.6 SRAM Error Sources

The SRAM error source is a parity error. Parity errors can be specified as non-maskable interrupts or a reset, in the OAD bit in the ECCOAD register for ECC error or PARIOAD register for parity error.

Table 47.2 SRAM error sources

Error source	DTC activation	DMAC activation
ECC error (SRAM0 area with ECC)	Not possible	Not possible
Parity error (SRAM0 area without ECC and SRAM1)	Not possible	Not possible

47.3.7 Access Cycle

Table 47.3 SRAM0 (ECC area 2000 0000h to 2000 3FFFh)

Bit setting	Read (cycle)		Write (cycle)	
	Word access	Halfword/Byte access	Word access	Halfword/Byte access
ECC Off ECCMOD[1] = 0	2		2	
ECC On ECCMOD[1] = 1	2		2	4

Table 47.4 SRAM0 (Parity area 2000 4000h to 2001 FFFFh)

Read (cycle)		Write (cycle)	
Word Access	Halfword/Byte access	Word access	Halfword/Byte access
2		2	

Table 47.5 SRAM1 (Parity area 2002 0000h to 2002 FFFFh)

Read (cycle)		Write (cycle)	
Word access	Halfword/Byte access	Word access	Halfword/Byte access
2		2	

47.4 Usage Notes

47.4.1 Instruction Fetch from the SRAM area

When using SRAM0 and SRAM1 to operate a program, initialize the SRAM area so that the CPU can correctly prefetch the data. If the CPU prefetches data from an uninitialized SRAM area, an ECC error or a parity error might occur. Initialize the additional 12-byte area from the end address of programs with a 4-byte boundary. Renesas recommends using a NOP instruction to initialize these areas.

47.4.2 Store Buffer of SRAM

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 out 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).

48. Flash Memory

48.1 Overview

The MCU provides up to 1-MB code flash memory and 16-KB data flash memory. The Flash Control Block (FCB) controls the programming commands.

Table 48.1 lists the specifications of the code flash memory and data flash memory, and Figure 48.1 shows the block diagram of the related modules. Figure 48.2 shows the configuration of the code flash memory, and Figure 48.3 shows the configuration of the data flash memory.

Table 48.1 Specifications of code flash memory and data flash memory

Parameter	Code flash memory	Data flash memory
Memory capacity	• Up to 1 MB of user area	16 KB of data area
Read cycle	• 32 MHz < ICLK frequency ≤ 48 MHz: Cache hit: 1 cycle Cache miss: 2, 3 cycles • ICLK frequency ≤ 32 MHz: Cache hit: 1 cycle Cache miss: 1 cycle	A read operation takes 6 cycles of FCLK in bytes (FCLK frequency ≤ 32 MHz)
Value after erasure	FFh	FFh
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).	
Security function	Protection against illicit tampering with or reading of data in flash memory	
Protection	Protection against erroneous overwriting of flash memory	
Background operations (BGOs)	Code flash memory can be read during data flash memory programming	
Units of programming and erasure	• 64-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	
	An expansion area of flash memory (option bytes) 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 serial programming mode (USB boot mode): • USBFS used • Dedicated hardware not required, so direct connection to a PC is possible. Programming in on-chip debug mode: • JTAG or 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.	

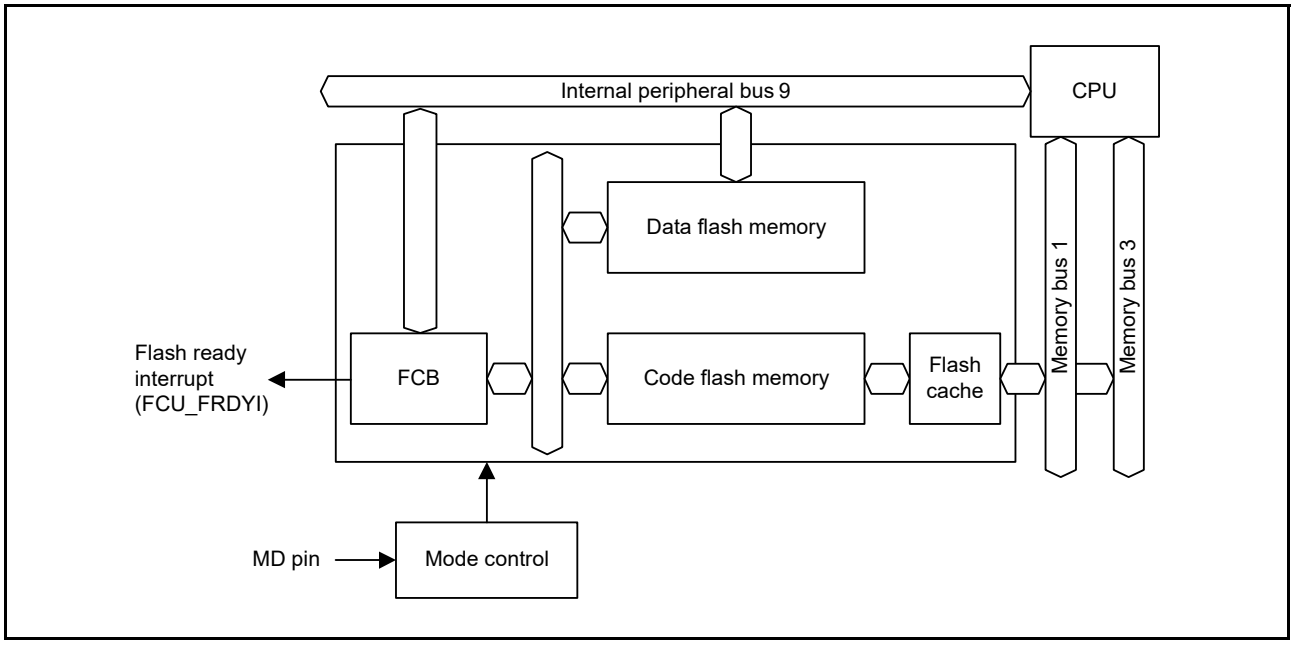


Figure 48.1 Flash memory-related modules block diagram

48.2 Memory Structure

Figure 48.2 shows the mapping of the code flash memory, and Table 48.2 shows the read and programming and erasure addresses of the code flash memory. The user space of the code flash memory is divided into 2-KB blocks, which serve as the units of erasure. The user area is available for storing the user program.

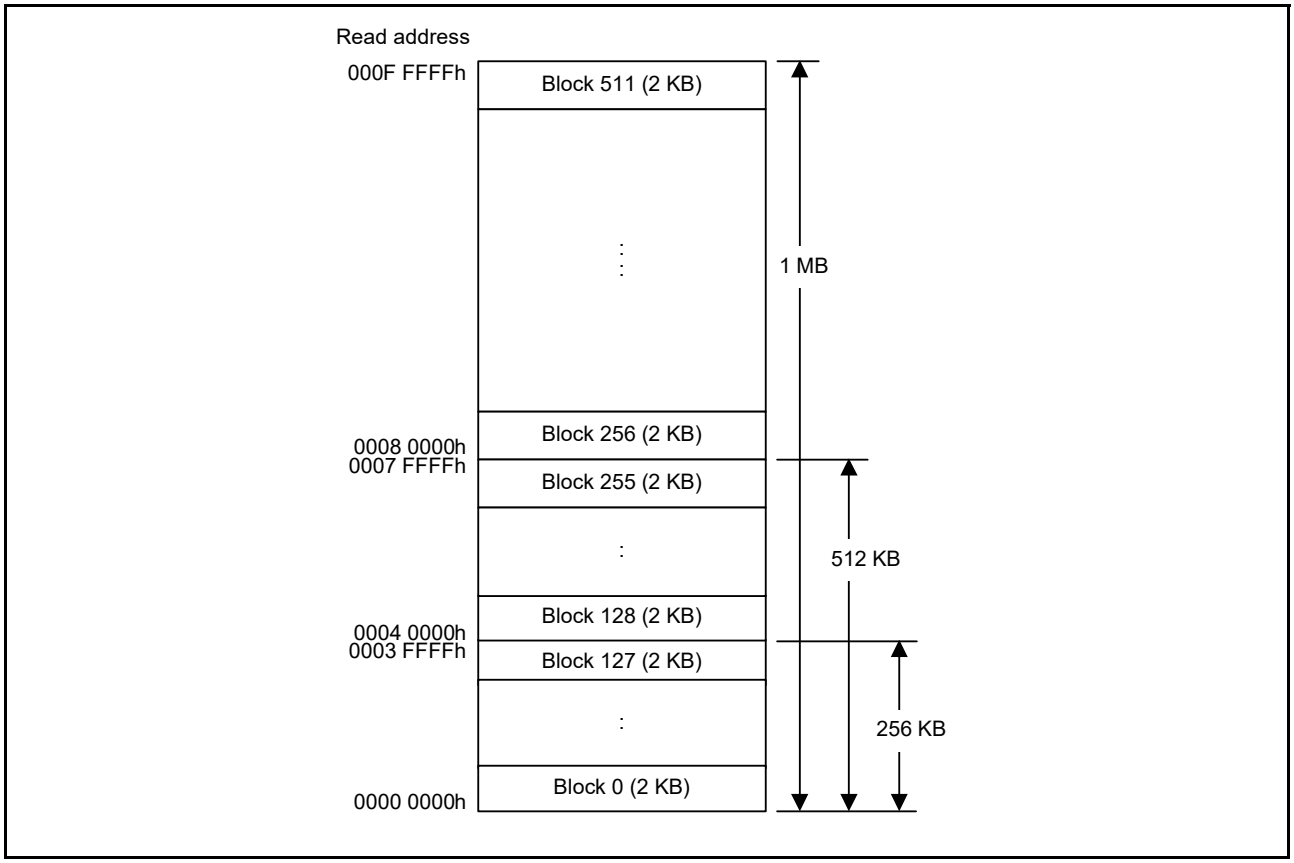


Figure 48.2 Mapping of the code flash memory

Table 48.2 Read and P/E addresses of the code flash memory

Size of code flash memory	Read address	P/E address	Number of blocks
1 MB	0000 0000h to 000F FFFFh	0000 0000h to 000F FFFFh	0 to 511

The data area of the data flash memory is divided into 1-KB blocks, with each being a unit for erasure. Figure 48.3 shows the mapping of the data flash memory, and Table 48.3 shows the read and programming and erasure addresses of the data flash memory.

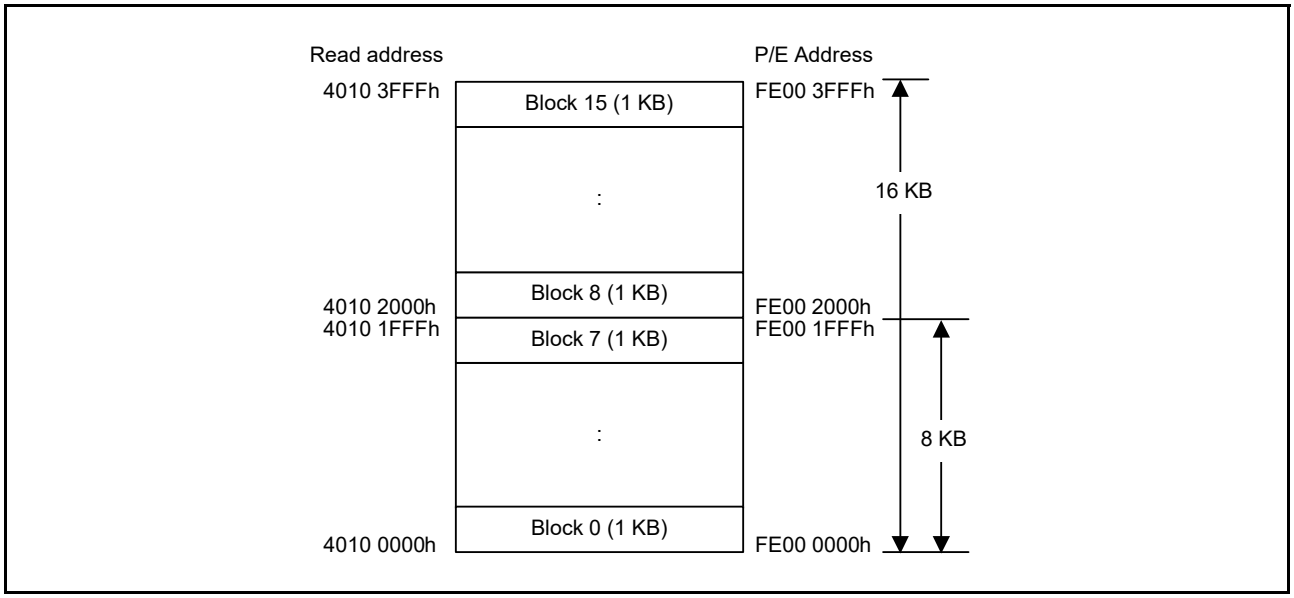


Figure 48.3 Mapping of the data flash memory

Table 48.3 Read and P/E addresses of the data flash memory

Size of data flash memory	Read address	P/E address	Number of blocks
16 KB	4010 0000h to 4010 3FFFh	FE00 0000h to FE00 3FFFh	0 to 15

48.3 Flash Cache

48.3.1 Overview

The flash cache (FCACHE) speeds up read access from the bus master to the flash memory. The FCACHE includes:

- FCACHE1, for CPU instruction fetch
- FCACHE2, for CPU operand access and DMA
- FLPF, for prefetch access of CPU instruction fetch.

Table 48.4 Flash cache overview

Parameter	Flash cache 1 (FCACHE1)	Flash cache 2 (FCACHE2)	Prefetch buffer (FLPF)
Cache target region	0000 0000h - 007F FFFFh	0000 0000h - 007F FFFFh	0000 0000h - 007F FFFFh
Target bus master	CPU instruction fetch	CPU operand access and access from other than CPU	FLPF
Capacity	256 bytes	8 bytes	16 bytes
Associativity	• 8-way set associative • 64 bits/entry (64-bit aligned data) • 4 entries/ways	• Fully associative • 64 bits/entry (64-bit aligned data) • 1 entry	• 64 bits/entry (64-bit aligned data) • 2 entries • Next address of previous CPU instruction
Access cycle	Cache hit: 0 wait Cache miss: According to SYSTEM.MEMWAIT Register setting: MEMWAIT = 0: 0 wait MEMWAIT = 1: 1 or 2 waits	Cache hit: 0 wait Cache miss: According to SYSTEM.MEMWAIT Register setting: MEMWAIT = 0: 0 wait MEMWAIT = 1: 1 or 2 waits	Cache hit: 0 wait Cache miss: According to SYSTEM.MEMWAIT Register setting: MEMWAIT = 0: 0 wait MEMWAIT = 1: 1 or 2 waits

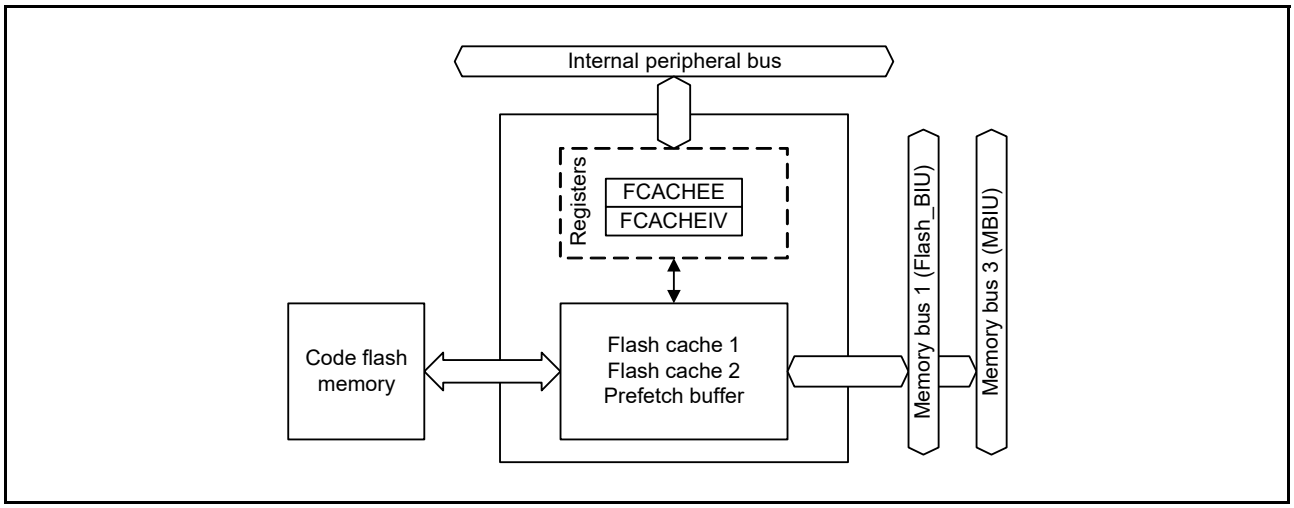


Figure 48.4 FCACHE block diagram

48.3.2 Register Descriptions

48.3.2.1 Flash Cache Enable Register (FCACHEE)

Address(es): FCACHE.FCACHEE 4001 C100h

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

Bit	Symbol	Bit name	Description	R/W
b0	FCACHEEN	FCACHE Enable	0: Disable FCACHE 1: Enable FCACHE.	R/W
b15 to b1	—	Reserved	These bits are read as 0.	R

The FCACHEE.FCACHEEN bit enables or disables the flash cache function for FCACHE1, FCACHE2, and FLPF. This bit does not affect FCACHEIV.FCACHEIV. When FCACHE is enabled, the HPROT[3] bit setting determines whether it is cacheable or non-cacheable. See section 15.7, Notes on using Flash Cache for details on HPROT[3].

48.3.2.2 Flash Cache Invalidate Register (FCACHEIV)

Address(es): FCACHE.FCACHEIV 4001 C104h

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

Bit	Symbol	Bit name	Description	R/W
b0	FCACHEIV	Flash Cache Invalidate	• Reads: 0: Do not invalidate 1: Invalidate. • Writes: When write value is 1, FCACHE is invalidated. When write value is 0, this setting is ignored.	R/W
b15 to b1	—	Reserved	These bits are read as 0.	R

When 1 is written to the FCACHEIV.FCACHEIV bit, flash cache data in FCACHE1, FCACHE2, and FLPF is invalidated.

48.4 Operation

Use the FCACHEE register to set up and enable flash operation. To set up the flash cache and prepare to rewrite the flash memory:

1. Disable the flash cache by resetting FCACHEE.FCACHEEN.*1
2. Set the MEMWAIT.MEMWAIT bit as required for the ICLK frequency and power control mode set in the OPCCR and SOPCCR registers.
3. Invalidate the flash cache by setting FCACHEIV.FCACHEIV.
4. Check that FCACHEIV.FCACHEIV is 0.
5. Enable the flash cache by setting FCACHEE.FCACHEEN.

Note 1. It is not necessary to disable the flash cache on the first setup after reset.

Note: Do not change operation mode (read mode, wait mode) when the flash cache is enabled.

48.4.1 Notice to use Flash Cache

When using flash cache by access from the CPU, Arm® MPU should also be set to cacheable.

See the *ARMv7-M Architecture Reference Manual* and the *ARM® Cortex®-M4 Devices Generic User Guide*.

48.5 Operating Modes Associated with the Flash Memory

Figure 48.5 shows a diagram of the mode transitions associated with the flash memory. For information on setting up the modes, see section 3, Operating Modes.

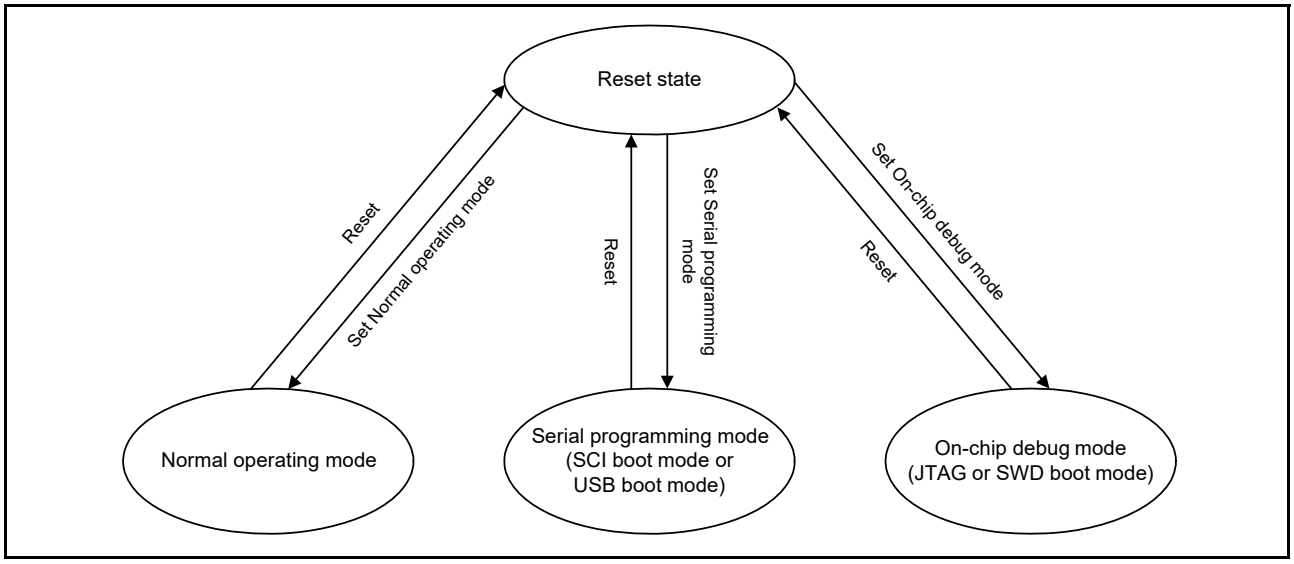


Figure 48.5 Mode transitions associated with flash memory

The flash memory areas where programming and erasure are permitted and where the boot program executes at a reset differ with the mode. Table 48.5 shows the differences between the modes.

Table 48.5 Difference between modes

Parameter	Normal operating mode	Serial programming mode (SCI or USB boot mode)	On-chip debug mode (JTAG or 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

48.5.1 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 48.6 shows how the ID code determines the authentication method.

Table 48.6 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/USB boot mode) On-chip debug mode (JTAG/SWD Boot mode)	FFh, …, FFh (all bytes FFh)	Protection disabled	ID code validation is not performed, the ID code always matches, and connection to the programmer or the on-chip debugger is permitted.
	Bit [127] = 1, bit [126] = 1, and at least one of all 16 bytes is not FFh	Protection enabled	Matching ID code: Authentication ends and connection with the programmer or the on-chip debugger is permitted. Non-matching ID code: Additional transition to the ID code protection waiting state. When the ID code sent from the programmer or the on-chip debugger is ALeRASE in ASCII code (414C_6552_4153_45FF_FFFF_FFFF_FFFF_FFFFh), the content of the user flash (code and data) area and configuration area are erased. However, forced erasure is not executed when the FSPR bit is 0.
	Bit [127] = 1 and bit [126] = 0	Protection enabled	Matching ID code: Authentication ends and connection with the programmer or the on-chip debugger is permitted. Non-matching ID code: Additional transition to the ID code protection waiting state.
	Bit [127] = 0	Protection enabled	ID code validation is not performed, the ID code is always non-matching, and connection to the programmer or the on-chip debugger is prohibited.

48.6 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 JTAG/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 continue while processing external communications and other functions.

Table 48.7 lists the programming methods and the associated operating modes.

Table 48.7 Programming methods

Programming method	Functional overview	Operating mode
Serial programming	A dedicated flash-memory programmer connected through the SCI or USBFS 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 or USBFS 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 is also capable of programming 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 programmed. As a result, a program resident in code flash memory is able to program data flash memory.	Normal operating mode
JTAG or SWD programming	A dedicated flash-memory programmer or an on-chip debugger connected through JTAG/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 through JTAG/SWD and a dedicated programming adapter board allow off-board programming of the flash memory, before it is mounted on the target system.	

The MCU supports programming commands for self-programming. Table 48.8 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 48.8 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. Results of reading from data flash memory to which nothing is written after erasure are not guaranteed, so use blank checking to confirm that writing to memory has not proceeded after erasure.	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 ROM. 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 security function for serial programming	Supported with conditions (only allows switching from enabled to disabled)	Supported with conditions (only allows switching from enabled to disabled)
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 security function. Authentication of ID codes is a security function for use with serial programming and with JTAG or SWD programming. Table 48.9 lists the security functions supported by the on-chip flash memory, and Table 48.10 lists the available operations and security settings.

Table 48.9 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 48.10 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	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.	ID authentication is not performed. • Blank check: supported • Block erasure: supported • Programming: supported • Security configuration: supported • Protection configuration: supported.	ID authentication is not performed

48.6.1 Configuration Area Bit Map

The bits used for ID authentication, startup area select, access window protection, and security configuration functions are mapped in Figure 48.6. The boot program must use these bits as hexadecimal data.

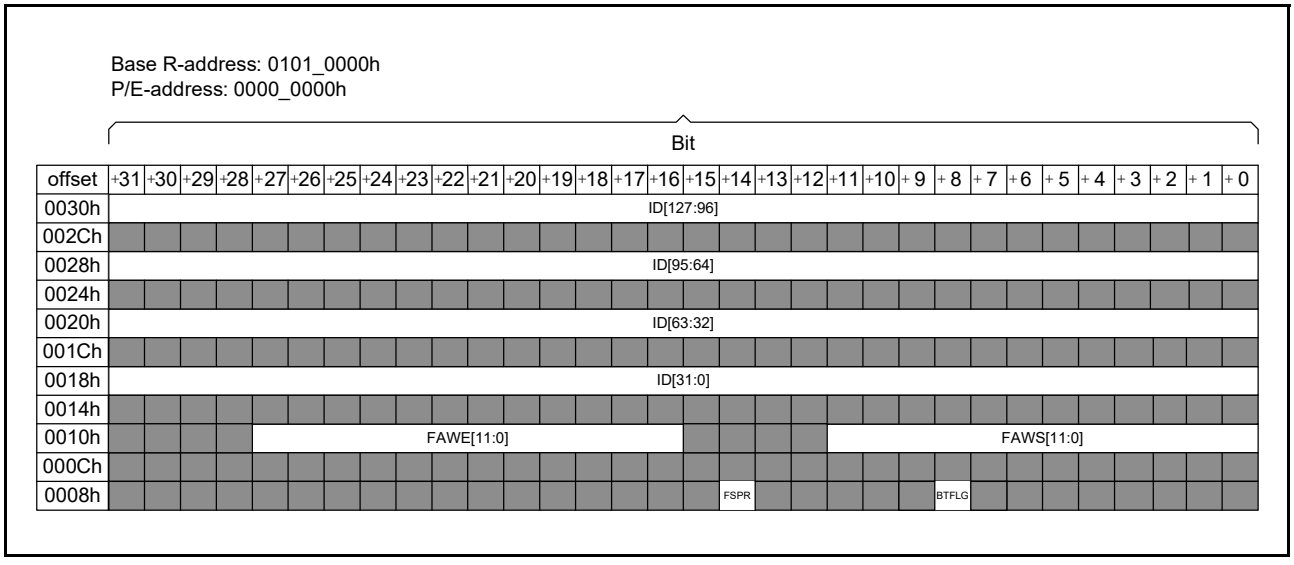
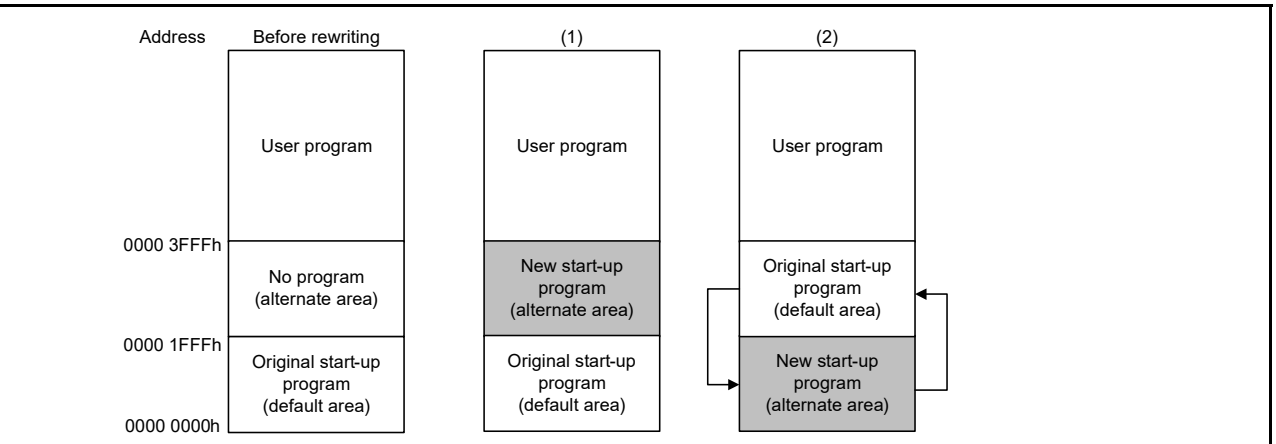


Figure 48.6 Configuration area bit map

48.6.2 Startup Area Select

The startup area select function allows the boot program to be safely updated. The size of the startup area is 8 KB, and the startup area is 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 or the AWSC register. The startup area can be locked by the FSPR bit. Figure 48.7 shows an overview of the startup program protection.



(1) Program a new start-up program in the alternate area. If the alternate area fails to be rewritten, the new start-up program can be rewritten again after starting up using the default area because the original start-up 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 48.7 Overview of startup program protection

48.6.3 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 48.8 shows an overview of flash area protection.

The access window is specified in both the FAWS [11:0] and FAWE [11:0] bits. Setting of the FAWE and FAWS bits in various conditions is described as follows:

- FAWE [11:0] = FAWS [11:0]: The P/E command can execute anywhere in the user area of the code flash memory.
- FAWE [11:0] > FAWS [11:0]: 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 bits.
- FAWE [11:0] < FAWS [11:0]: The P/E command cannot execute anywhere in the user area of the code flash memory.

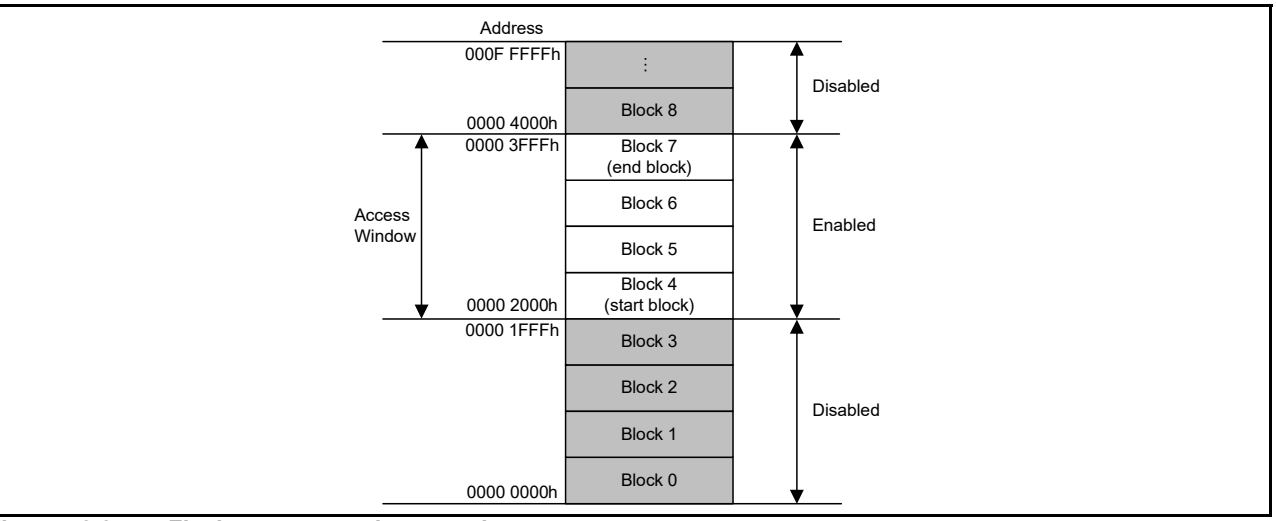


Figure 48.8 Flash area protection overview

48.7 Programming Commands

The FCB controls the programming commands.

48.8 Suspend Operation

The forced stop command forces the blank check or block 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.

48.9 Protection

The types of protection provided include:

- Software protection
- Error protection
- Boot program protection.

48.10 Serial Programming Mode

The serial programming modes include:

- Boot mode with SCI9
- USB boot mode with the USBFS.

Table 48.11 lists the I/O pins of the flash memory-related modules.

Table 48.11 I/O pins of flash memory-related modules

Pin name	I/O	Applicable modes	Function
MD	Input	SCI boot mode USB 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
USB_DP, USB_DM	I/O	USB boot mode	USB data I/O
USB_VBUS	Input		Detection of connection and disconnection of USB cables

Note: Serial programming mode is not executed when security MPU is enabled.

48.10.1 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. The USB cable must not be connected on reset release.

Figure 48.9 shows the system configuration for operations in boot mode.

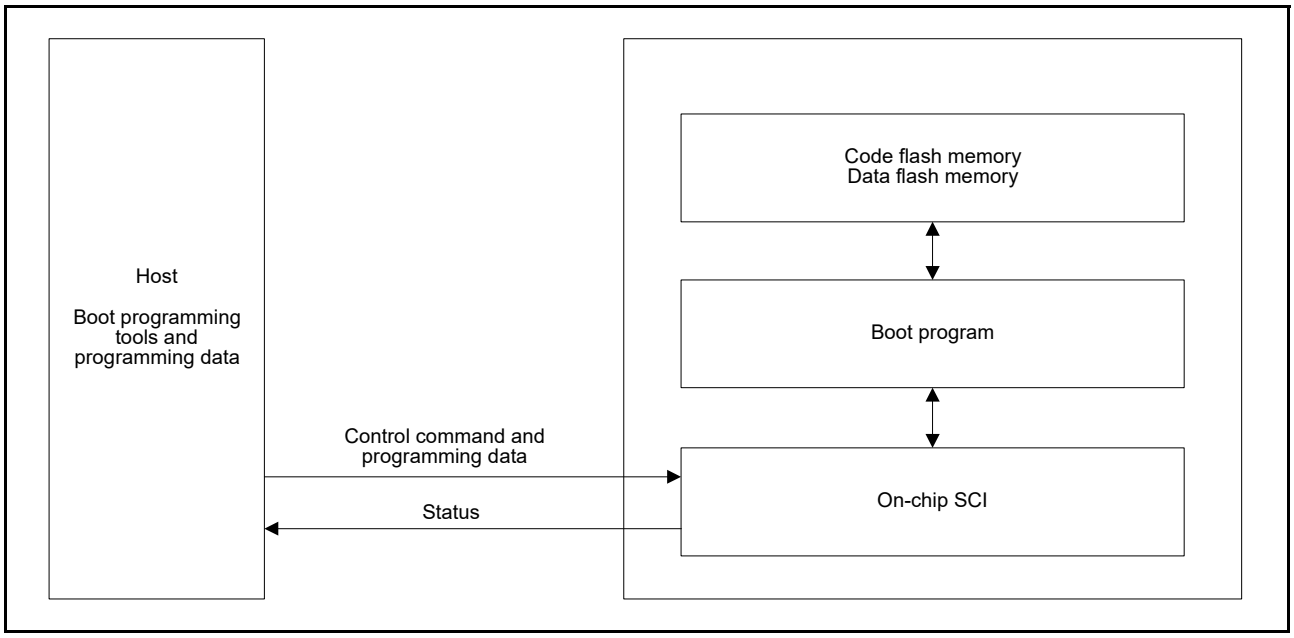


Figure 48.9 System configuration in SCI boot mode

48.10.2 USB Boot Mode

In USB boot mode, the code and data flash memory are programmed or erased by control commands and data for programming transmitted from an externally connected host through the USB interface.

Using USB boot mode requires preparation on the host side of the tools for transmitting control commands and data for programming. Figure 48.10 shows the configuration of a system in USB boot mode. The USB cable must be connected on reset release.

For a USB self-powered system, the total current consumption from VBUS should not exceed 100 mA.

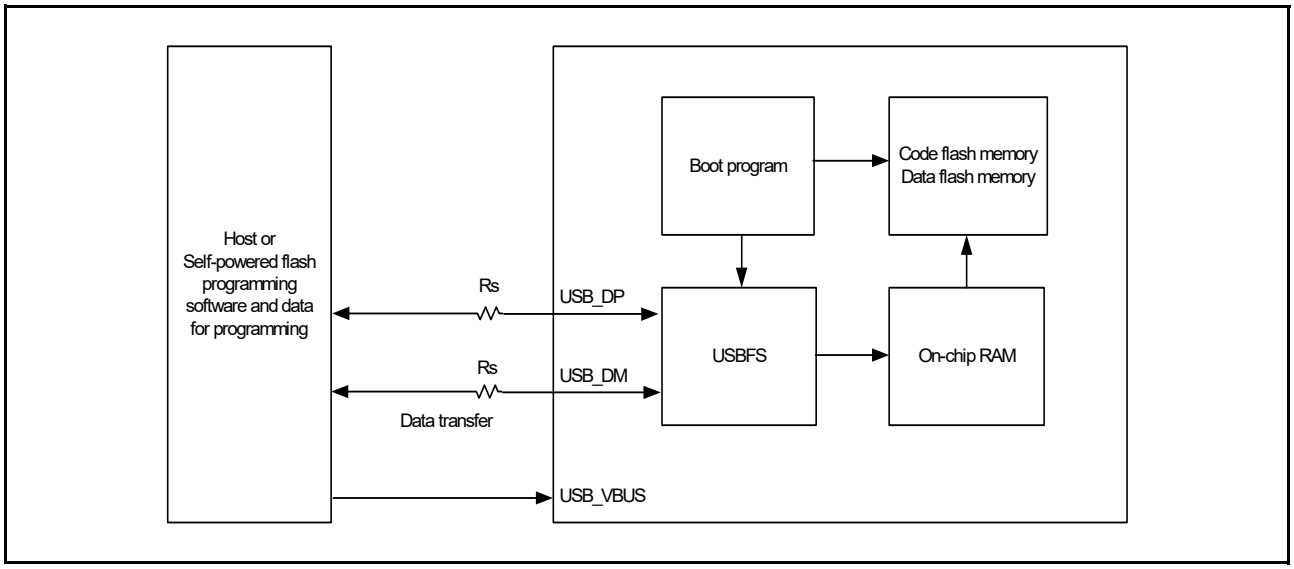


Figure 48.10 System configuration in USB boot mode

48.11 Using a Serial Programmer

A dedicated flash memory programmer can be used to program the flash memory in serial programming mode.

48.11.1 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 48.11 shows the environment recommended by Renesas for programming the flash memory of the MCU with data.

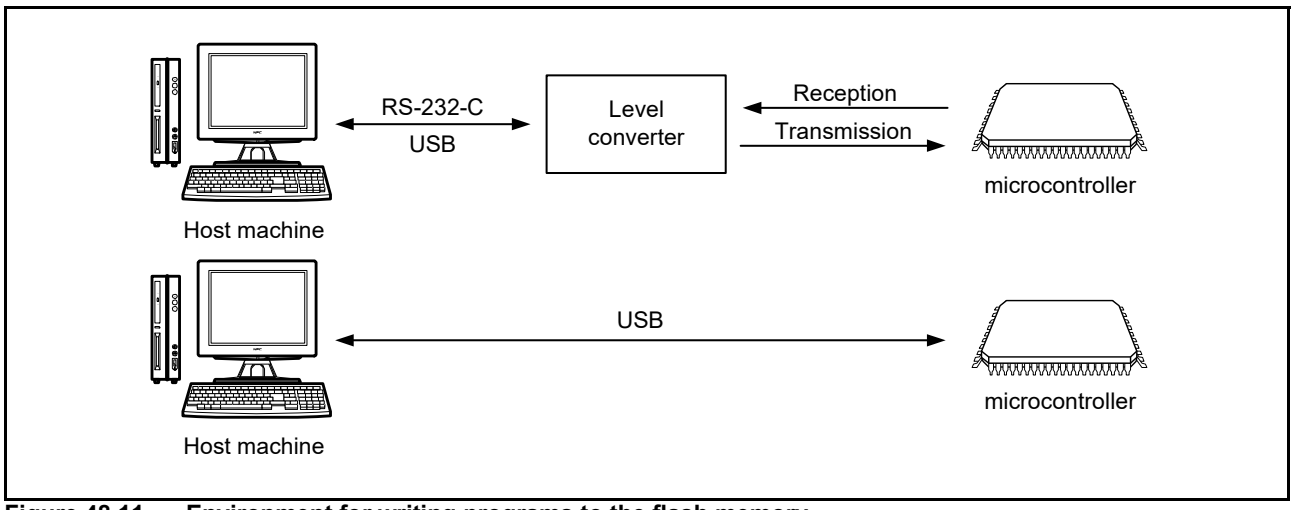


Figure 48.11 Environment for writing programs to the flash memory

48.12 Self-Programming

48.12.1 Overview

The MCU supports programming of the flash memory by the user program itself. 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.

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 Table 48.12. This program can also be copied in advance to and executed from the internal SRAM or external memory. When executing from the internal SRAM or external memory, this program can also program the code flash memory area.

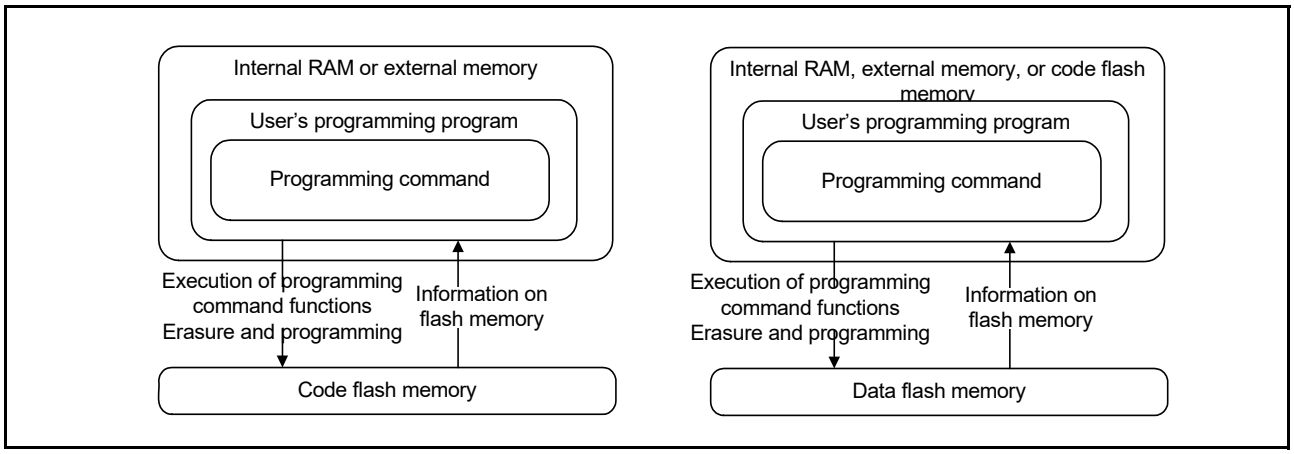


Figure 48.12 Schematic view of self-programming

48.12.2 Background Operation

Background operation can be used when a combination of the flash memory for writing and reading is as listed in Table 48.12.

Table 48.12 Conditions under which background operation is available

Product	Writable range	Readable range
All products	Data flash memory	Code flash memory

48.13 Reading the Flash Memory

48.13.1 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.

48.13.2 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.

48.14 Usage Notes

48.14.1 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.

48.14.2 Suspension with Erase Suspend Commands

When suspending an erase operation with the erase suspend command, complete the operation with a resume command.

48.14.3 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.

48.14.4 Reset during Programming and Erasure

If inputting a reset from the RES pin, release the reset after a reset input time of at least tRESW (see section 52, Electrical Characteristics) within the range of the operating voltage defined in the electrical characteristics.

The IWDT reset and software reset do not require a tRESW input time.

48.14.5 Non-maskable Interrupt Disabled during Programming and Erasure

Do not enable non-maskable interrupts*1 during programming and erasure operations in the code flash memory. When a non-maskable interrupt occurs during a programming or erasure operation, the vectors are fetched from the code flash memory, and undefined data is read. This constraint only applies to the code flash memory.

Note 1. A non-maskable interrupt is an 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, VBATT monitor interrupt, SRAM parity error interrupt, SRAM ECC error interrupt, MPU bus master error interrupt, MPU bus slave error interrupt, or CPU stack pointer monitor interrupt.

48.14.6 Location of Interrupt Vectors during Programming and Erasure

When an interrupt occurs during programming and erasure operations, the vector can be fetched from the code flash memory. 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.

48.14.7 Programming and Erasure in Low-Speed Operating Mode

Do not program or erase the flash memory when low-speed operating mode is selected in the SOPCCR register for low power consumption functions.

48.14.8 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 48.14.9, Actions Prohibited during Programming and Erasure, erase the area again.

48.14.9 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] bit value
- Do not update the SOPCCR.SOPCM bit value
- Do not change the division ratio of the flash interface clock (FCLK)
- 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 DFLCTL.DFLEN bit value during a program or erase operation to the data flash memory.

49. Segment LCD Controller/Driver (SLCDC)

49.1 Overview

The MCU provides a controller for LCD display and display pins. Table 49.1 lists the SLCDC specifications.

Table 49.1 SLCDC specifications

Item	Description
Features	• Liquid crystal waveform (waveform A or B) selectable • LCD driver voltage generator can switch between internal voltage boosting method, capacitor split method, and external resistance division method • Automatic output of segment and shared signals based on automatic display data register read • Voltage boost circuit reference voltage selectable from16 steps (contrast adjustment) • LCD blinking and display selectable.
Number of pins	For details on the number of pins, see Table 49.2, SLCDC display function pins for 145/144-pin products.
Source clocks	• Main clock oscillator • Sub-clock oscillator • Low-speed on-chip oscillator • High-speed on-chip oscillator.
Module-stop state function	Module-stop state can be set to reduce power consumption

The number of LCD display function pins for the MCU differs depending on the product. Table 49.2 to Table 49.4 show the display function pins for products with different pin counts. Table 49.5 to Table 49.7 show the maximum number of pixels for products with different pin counts. Figure 49.1 shows the SLCDC block diagram.

Table 49.2 SLCDC display function pins for 145/144-pin products

Item	145, 144 Pins															
LCD controller/ driver	Number of segment pins (SEG): 52 (48)*1 Number of common pins (COM): 8															
Multiplexed I/O port	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PORT1	SEG 25	SEG 24	SEG0/ COM4	CAP L*2	CAPH *2	-	-	-	COM 3	COM 2	COM 1	COM 0	VL4*3	VL3*3	VL2*3	VL1*3
PORT2	-	-	-	-	-	-	-	-	-	-	-	SEG 23	SEG 22	SEG 21	-	-
PORT3	SEG 5	SEG 4	SEG 20	SEG 9	SEG 10	SEG 11	SEG 12	SEG 13	SEG 14	SEG 15	SEG 16	SEG 17	SEG3 / COM 7	SEG2 / COM 6	SEG1 / COM 5	-
PORT5	-	-	-	-	-	-	-	-	-	-	-	-	SEG 51	SEG 50	SEG 49	SEG 48
PORT6	-	SEG 34	SEG 33	SEG 32	SEG 31	SEG 30	SEG 29	SEG 28	-	SEG 35	SEG 36	SEG 37	SEG 38	SEG 39	SEG 40	SEG 41
PORT8	-	-	-	-	-	-	SEG 19	SEG 18	SEG 27	SEG 26	SEG 42	SEG 43	SEG 47	SEG 46	SEG 45	SEG 44
PORT9	-	-	-	-	-	-	-	-	-	-	-	-	-	SEG8	SEG7	SEG6

Note 1. ( ) indicates the number of signal output pins when 8-time slice is selected.
Note 2. CAPH and CAPL are capacitor connection pins for the LCD controller/driver.
Note 3. VL1, VL2, VL3, and VL4 are power supply pins for driving the LCD.

Table 49.3 SLCDC display function pins for 121-pin products

Item	121 Pins															
LCD controller/ driver	Number of segment pins (SEG): 38 (34)*1 Number of common pins (COM): 8															
Multiplexed I/O port	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PORT1	SEG 25	SEG 24	SEG0/ COM4	CAPL *2	CAP H*2	-	-	-	COM 3	COM 2	COM 1	COM 0	VL4*3	VL3*3	VL2*3	VL1*3
PORT2	-	-	-	-	-	-	-	-	-	-	-	SEG 23	SEG 22	SEG 21	-	-
PORT3	SEG 5	SEG 4	SEG 20	-	-	-	SEG 12	SEG 13	SEG 14	SEG 15	SEG 16	SEG 17	SEG3 / COM 7	SEG2 / COM 6	SEG1 / COM 5	-
PORT5	-	-	-	-	-	-	-	-	-	-	-	-	SEG 51	SEG 50	SEG 49	SEG 48
PORT6	-	-	SEG 33	SEG 32	SEG 31	SEG 30	SEG 29	SEG 28	-	-	SEG 36	SEG 37	SEG 38	SEG 39	SEG 40	SEG 41
PORT8	-	-	-	-	-	-	SEG 19	SEG 18	-	-	-	-	-	-	SEG 45	SEG 44

Note 1. ( ) indicates the number of signal output pins when 8-time slice is selected.
Note 2. CAPH and CAPL are capacitor connection pins for the LCD controller/driver.
Note 3. VL1, VL2, VL3, and VL4 are power supply pins for driving the LCD.

Table 49.4 SLCDC display function pins for 100-pin products

Item	100 Pins															
LCD controller/ driver	Number of segment pins (SEG): 26 (22)*1 Number of common pins (COM): 8															
Multiplexed I/O port	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PORT1	SEG 25	SEG 24	SEG0 / COM 4	CAPL *2	CAP H*2	-	-	-	COM 3	COM 2	COM 1	COM 0	VL4*3	VL3*3	VL2*3	VL1*3
PORT2	-	-	-	-	-	-	-	-	-	-	-	SEG 23	SEG 22	SEG 21	-	-
PORT3	-	-	-	-	-	-	-	-	SEG 14	SEG 15	SEG 16	SEG 17	SEG3 / COM 7	SEG2 / COM 6	SEG1 / COM 5	-
PORT5	-	-	-	-	-	-	-	-	-	-	-	-	SEG 51	SEG 50	SEG 49	SEG 48
PORT6	-	-	-	-	-	SEG 30	SEG 29	SEG 28	-	-	-	-	SEG 38	SEG 39	SEG 40	SEG 41
PORT8	-	-	-	-	-	-	SEG 19	SEG 18	-	-	-	-	-	-	-	-

Note 1. ( ) indicates the number of signal output pins when 8-time slice is selected.
Note 2. CAPH and CAPL are capacitor connection pins for the LCD controller/driver.
Note 3. VL1, VL2, VL3, and VL4 are power supply pins for driving the LCD.

Table 49.5 Maximum number of pixels for 100-pin products

Drive waveform for LCD driver	LCD driver voltage generator	Bias mode	Number of time slices	Maximum number of pixels
Waveform A	External resistance division	-	Static	26 (26 segment signals, 1 common signal)
		1/2	2	52 (26 segment signals, 2 common signals)
			3	78 (26 segment signals, 3 common signals)
		1/3	3	
			4	104 (26 segment signals, 4 common signals)
		1/4	8	176 (22 segment signals, 8 common signals)
	Internal voltage boosting	1/3	3	78 (26 segment signals, 3 common signals)
			4	104 (26 segment signals, 4 common signals)
		1/4	8	176 (22 segment signals, 8 common signals)
	Capacitor split	1/3	3	78 (26 segment signals, 3 common signals)
			4	104 (26 segment signals, 4 common signals)
Waveform B	External resistance division, internal voltage boosting	1/3	4	
		1/4	8	176 (22 segment signals, 8 common signals)
	Capacitor split	1/3	4	104 (26 segment signals, 4 common signals)

Table 49.6 Maximum number of pixels for 121-pin products

Drive waveform for LCD driver	LCD driver voltage generator	Bias mode	Number of time slices	Maximum number of pixels
Waveform A	External resistance division	-	Static	38 (38 segment signals, 1 common signal)
		1/2	2	76 (38 segment signals, 2 common signals)
			3	114 (38 segment signals, 3 common signals)
		1/3	3	
			4	152 (38 segment signals, 4 common signals)
		1/4	8	272 (34 segment signals, 8 common signals)
	Internal voltage boosting	1/3	3	114 (38 segment signals, 3 common signals)
			4	152 (38 segment signals, 4 common signals)
		1/4	8	272 (34 segment signals, 8 common signals)
	Capacitor split	1/3	3	114 (38 segment signals, 3 common signals)
			4	152 (38 segment signals, 4 common signals)
Waveform B	External resistance division, internal voltage boosting	1/3	4	
		1/4	8	272 (34 segment signals, 8 common signals)
	Capacitor split	1/3	4	152 (38 segment signals, 4 common signals)

Table 49.7 Maximum number of pixels for 145/144-pin products

Drive waveform for LCD driver	LCD driver voltage generator	Bias mode	Number of time slices	Maximum number of pixels
Waveform A	External resistance division	-	Static	52 (52 segment signals, 1 common signal)
		1/2	2	104 (52 segment signals, 2 common signals)
			3	156 (52 segment signals, 3 common signals)
		1/3	3	
			4	208 (52 segment signals, 4 common signals)
		1/4	8	384 (48 segment signals, 8 common signals)
	Internal voltage boosting	1/3	3	156 (52 segment signals, 3 common signals)
			4	208 (52 segment signals, 4 common signals)
		1/4	8	384 (48 segment signals, 8 common signals)
	Capacitor split	1/3	3	156 (52 segment signals, 3 common signals)
			4	208 (52 segment signals, 4 common signals)
Waveform B	External resistance division, internal voltage boosting	1/3	4	
		1/4	8	384 (48 segment signals, 8 common signals)
	Capacitor split	1/3	4	208 (52 segment signals, 4 common signals)

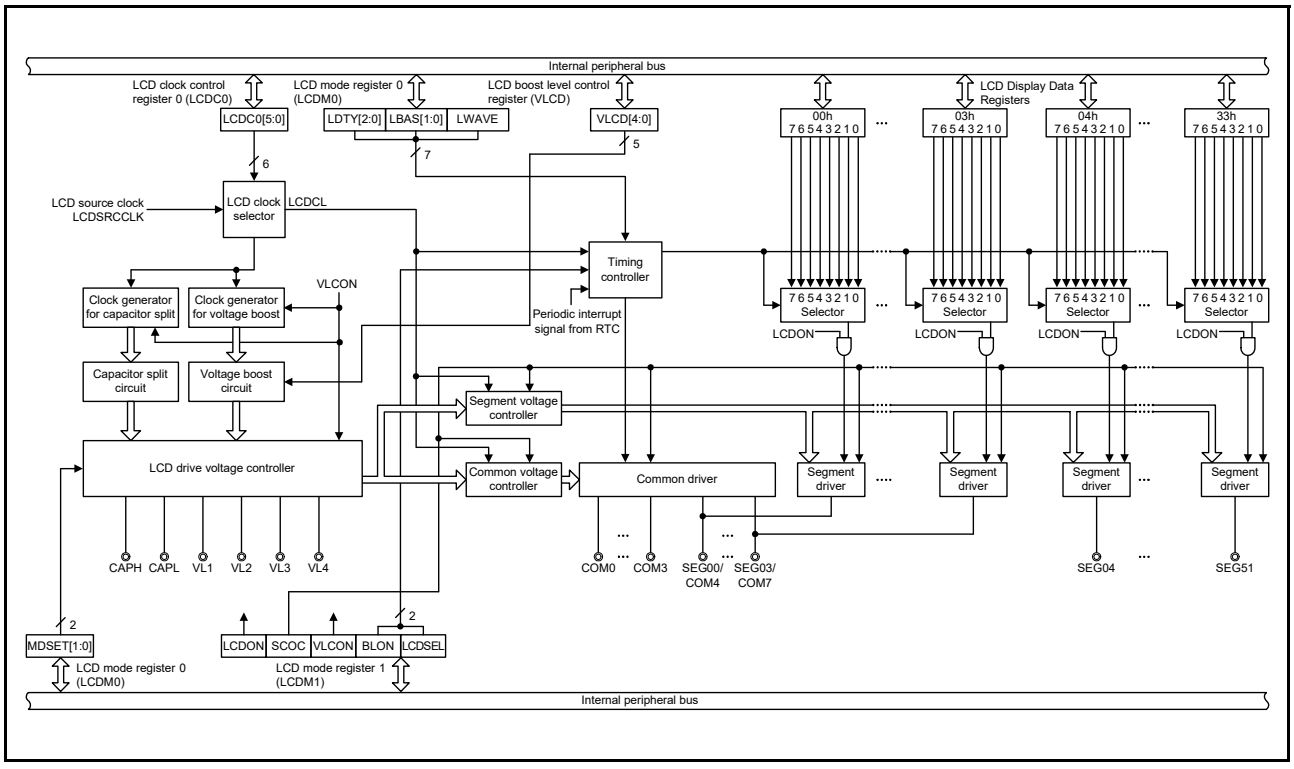


Figure 49.1 SLCDC block diagram

49.2 Register Descriptions

49.2.1 LCD Mode Register 0 (LCDM0)

Address(es): SLCDC.LCDM0 4008 2000h

	b7	b6	b5	b4	b3	b2	b1	b0
	MDSET[1:0]		LWAVE	LDTY[2:0]			LBAS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b1, b0	LBAS[1:0]	LCD Display Bias Method Select	b1 b0 0 0: 1/2 bias method 0 1: 1/3 bias method 1 0: 1/4 bias method 1 1: Setting prohibited.	R/W
b4 to b2	LDTY[2:0]	Time Slice of LCD Display Select	b4 b2 0 0 0: Static 0 0 1: 2-time slice 0 1 0: 3-time slice 0 1 1: 4-time slice 1 0 1: 8-time slice. Other settings are prohibited.	R/W
b5	LWAVE	LCD Display Waveform Select	0: Waveform A 1: Waveform B.	R/W
b7, b6	MDSET[1:0]	LCD Drive Voltage Generator Select	b7 b6 0 0: External resistance division method 0 1: Internal voltage boosting method 1 0: Capacitor split method 1 1: Setting prohibited.	R/W

Note: Do not rewrite the LCDM0 value when the SCOC bit of the LCDM1 register is 1.
Note: When static is selected (LDTY[2:0] = 000b), you must set the LBAS[1:0] bits to the default value (00b). Otherwise, the operation is not guaranteed.
Note: Only the combinations of display waveform, number of time slices, and bias method shown in Table 49.8 are supported. Combinations of settings not shown in Table 49.8 are prohibited.

Table 49.8 Combinations of display waveform, time slices, bias method, and frame frequency

Display mode			Set value						Driving voltage generation method		
Display waveform	Number of time slices	Bias mode	LWAVE	LDTY[2:0]			LBAS[1:0]		External resistance division	Internal voltage boosting	Capacitor split
Waveform A	8	1/4	0	1	0	1	1	0	A	A	N/A
Waveform A	4	1/3	0	0	1	1	0	1	A	A	A
Waveform A	3	1/3	0	0	1	0	0	1	A	A	A
Waveform A	3	1/2	0	0	1	0	0	0	A	N/A	N/A
Waveform A	2	1/2	0	0	0	1	0	0	A	N/A	N/A
Waveform A	Static		0	0	0	0	0	0	A	N/A	N/A
Waveform B	8	1/4	1	1	0	1	1	0	A	A	N/A
Waveform B	4	1/3	1	0	1	1	0	1	A	A	A

A: Available
N/A: Not available

49.2.2 LCD Mode Register 1 (LCDM1)

Address(es): SLCDC.LCDM1 4008 2001h

	b7	b6	b5	b4	b3	b2	b1	b0
	LCDON	SCOC	VLCON	BLON	LCDSEL	—	—	LCDVLM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b0	LCDVLM	Voltage Boosting Pin Initial Value Switching Control	0: Set when VCC ≥ 2.7 V 1: Set when VCC ≤ 4.2 V. In condition 2.7V ≤ VCC ≤ 4.2V, any value can be set.	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	LCDSEL	Display Data Area Control	b4 b3	R/W
b4	BLON	Display Data Area Control	0 0: Display an A-pattern area data (lower 4 bits of LCD display data register) 0 1: Display a B-pattern area data (upper 4 bits of LCD display data register) 1 0: Alternately display A-pattern and B-pattern area data (blinking display associated with the periodic interrupt (RTC_PRD) timing of the realtime clock (RTC)) 1 1: Alternately display A-pattern and B-pattern area data (blinking display associated with the periodic interrupt (RTC_PRD) timing of the realtime clock (RTC)).	R/W
b5	VLCON	Voltage Boost Circuit or Capacitor Split Circuit Operation Enable/Disable	0: Stop voltage boost circuit or capacitor split circuit operation 1: Enable voltage boost circuit or capacitor split circuit operation.*2	R/W
b6	SCOC	LCD Display Enable/Disable	b7 b6	R/W
b7	LCDON	LCD Display Enable/Disable	0 0: Output ground level to segment/common pin 0 1: Display off (all segment outputs are deselected) 1 0: Output ground level to segment/common pin 1 1: Display on.	R/W

Note 1. This bit is used to improve voltage boost efficiency when using the voltage boost circuit by setting the initial VLX pin status. If VCC is 2.7 V or higher when voltage boosting starts, set the LCDVLM bit to 0; if VCC is 4.2 V or lower, set the LCDVLM bit to 1. If VCC is within the range between 2.7 V and 4.2 V, the LCDVLM bit may be set to 0 or 1.
Note 2. This setting is prohibited when using the external resistance division method.
Note 3. To reduce power consumption when nothing is to be displayed on the LCD while the voltage boost circuit is in use, set the SCOC and VLCON bits to 0 and set the LCDM0.MDSET[1:0] bits to 00b. When LCDM0.MDSET[1:0] = 01b, the internal reference voltage generator operates and so consumes power.
Note 4. When the external resistance division method is set (LCDM0.MDSET[1:0] = 00b) or capacitor split method is set (LCDM0.MDSET[1:0] = 10b), set the LCDVLM bit to 0.
Note 5. Do not rewrite the VLCON and LCDVLM bits while SCOC = 1.
Note 6. Set the BLON and LCDSEL bits to 0 when 8 is selected as the number of time slices for the display mode.
Note 7. To use the internal voltage boosting method, specify the reference voltage by using the VLCD register (select the internal boosting method (by setting the LCDM0.MDSET[1:0] bits to 01b) if the default reference voltage is used), wait for the reference voltage setup time (minimum 5 ms), and then set the VLCON bit to 1.

49.2.3 LCD Clock Control Register 0 (LCDC0)

Address(es): SLCDC.LCDC0 4008 2002h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LCDC0[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit name	Description	R/W
b5 to b0	LCDC0[5:0]	LCD Clock (LCDCL) Setting	b5 b0 0 0 0 0 0 1: (Sub clock)/2^2 or (LOCO clock)/2^2 0 0 0 0 1 0: (Sub clock)/2^3 or (LOCO clock)/2^3 0 0 0 0 1 1: (Sub clock)/2^4 or (LOCO clock)/2^4 0 0 0 1 0 0: (Sub clock)/2^5 or (LOCO clock)/2^5 0 0 0 1 0 1: (Sub clock)/2^6 or (LOCO clock)/2^6 0 0 0 1 1 0: (Sub clock)/2^7 or (LOCO clock)/2^7 0 0 0 1 1 1: (Sub clock)/2^8 or (LOCO clock)/2^8 0 0 1 0 0 0: (Sub clock)/2^9 or (LOCO clock)/2^9 0 0 1 0 0 1: (Sub clock)/2^{10} or (LOCO clock)/2^{10} 0 1 0 0 0 1: (Main clock)/2^8 or (HOCO clock)/2^8 0 1 0 0 1 0: (Main clock)/2^9 or (HOCO clock)/2^9 0 1 0 0 1 1: (Main clock)/2^{10}or (HOCO clock)/2^{10} 0 1 0 1 0 0: (Main clock)/2^{11}or (HOCO clock)/2^{11} 0 1 0 1 0 1: (Main clock)/2^{12} or (HOCO clock)/2^{12} 0 1 0 1 1 0: (Main clock)/2^{13} or (HOCO clock)/2^{13} 0 1 0 1 1 1: (Main clock)/2^{14} or (HOCO clock)/2^{14} 0 1 1 0 0 0: (Main clock)/2^{15} or (HOCO clock)/2^{15} 0 1 1 0 0 1: (Main clock)/2^{16} or (HOCO clock)/2^{16} 0 1 1 0 1 0: (Main clock)/2^{17} or (HOCO clock)/2^{17} 0 1 1 0 1 1: (Main clock)/2^{18} or (HOCO clock)/2^{18} 1 0 1 0 1 1: (Main clock)/2^{19} or (HOCO clock)/2^{19}. Other settings are prohibited.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Be sure to set bits [6] and [7] to 0.

Note 2. Set the frame frequency in a range from 32 Hz to 128 Hz. Set the LCD clock (LCDCL) to no more than 512 Hz when using the internal voltage boosting method and the capacitor split method.

Note 3. Do not set LCDC0 when the LCDM1.SCOC bit is 1.

49.2.4 LCD Boost Level Control Register (VLCD)

Address(es): SLCDC.VLCD 4008 2003h

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

Bit	Symbol	Bit name	Description	R/W
b4 to b0	VLCD[4:0]	Reference Voltage (Contrast Adjustment) Select		R/W

b4	b3	b2	b1	b0	VL1 voltage: Reference voltage	VL4 voltage: 1/3 bias method	VL4 voltage: 1/4 bias method
0	0	1	0	0	1.00 V	3.00 V	4.00 V
0	0	1	0	1	1.05 V	3.15 V	4.20 V
0	0	1	1	0	1.10 V	3.30 V	4.40 V
0	0	1	1	1	1.15 V	3.45 V	4.60 V
0	1	0	0	0	1.20 V	3.60 V	4.80 V
0	1	0	0	1	1.25 V	3.75 V	5.00 V
0	1	0	1	0	1.30 V	3.90 V	5.20 V
0	1	0	1	1	1.35 V	4.05 V	Setting prohibited
0	1	1	0	0	1.40 V	4.20 V	Setting prohibited
0	1	1	0	1	1.45 V	4.35 V	Setting prohibited
0	1	1	1	0	1.50 V	4.50 V	Setting prohibited
0	1	1	1	1	1.55 V	4.65 V	Setting prohibited
1	0	0	0	0	1.60 V	4.80 V	Setting prohibited
1	0	0	0	1	1.65 V	4.95 V	Setting prohibited
1	0	0	1	0	1.70 V	5.10 V	Setting prohibited
1	0	0	1	1	1.75 V	5.25 V	Setting prohibited

Other settings are prohibited.

b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The VLCD setting is valid only when the voltage boost circuit is operating.
Note 2. Be sure to set bits [5] to [7] to 0.
Note 3. Be sure to change the VLCD value after stopping the operation of the voltage boost circuit (VLCON = 0).
Note 4. To use the internal voltage boosting method, specify the reference voltage by using the VLCD register (select the internal boosting method (by setting the LCDM0.MDSET[1:0] bits to 01b) if the default reference voltage is used), wait for the reference voltage setup time (5 ms (min.)), and then set VLCON to 1.
Note 5. When using the external resistance division method and the capacitor split method, use the default value (04h) for the VLCD resistor.

49.3 LCD Display Data Registers

The LCD display data registers are mapped as shown in Table 49.9 and Table 49.10. The contents displayed on the LCD can be changed by changing the contents of the LCD display data registers.

Table 49.9 Relationship between LCD Display Data Register contents and segment/common outputs (1 of 2)
Other than 8-time-slice (static, 2-time slice, 3-time slice, and 4-time slice)

Register name	Address	b7	b6	b5	b4	b3	b2	b1	b0	145/144-pin	121-pin	100-pin	64-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0				
SEG00	4008 2100h	SEG00 (B-pattern area)				SEG00 (A-pattern area)				A	A	A	A
SEG01	4008 2101h	SEG01 (B-pattern area)				SEG01 (A-pattern area)				A	A	A	N/A
SEG02	4008 2102h	SEG02 (B-pattern area)				SEG02 (A-pattern area)				A	A	A	N/A
SEG03	4008 2103h	SEG03 (B-pattern area)				SEG03 (A-pattern area)				A	A	N/A	N/A
SEG04	4008 2104h	SEG04 (B-pattern area)				SEG04 (A-pattern area)				A	A	N/A	N/A
SEG05	4008 2105h	SEG05 (B-pattern area)				SEG05 (A-pattern area)				A	A	N/A	N/A
SEG06	4008 2106h	SEG06 (B-pattern area)				SEG06 (A-pattern area)				A	N/A	N/A	N/A
SEG07	4008 2107h	SEG07 (B-pattern area)				SEG07 (A-pattern area)				A	N/A	N/A	N/A
SEG08	4008 2108h	SEG08 (B-pattern area)				SEG08 (A-pattern area)				A	N/A	N/A	N/A
SEG09	4008 2109h	SEG09 (B-pattern area)				SEG09 (A-pattern area)				A	N/A	N/A	N/A
SEG10	4008 210Ah	SEG10 (B-pattern area)				SEG10 (A-pattern area)				A	N/A	N/A	N/A
SEG11	4008 210Bh	SEG11 (B-pattern area)				SEG11 (A-pattern area)				A	N/A	N/A	N/A
SEG12	4008 210Ch	SEG12 (B-pattern area)				SEG12 (A-pattern area)				A	A	N/A	N/A
SEG13	4008 210Dh	SEG13 (B-pattern area)				SEG13 (A-pattern area)				A	A	N/A	N/A
SEG14	4008 210Eh	SEG14 (B-pattern area)				SEG14 (A-pattern area)				A	A	A	N/A
SEG15	4008 210Fh	SEG15 (B-pattern area)				SEG15 (A-pattern area)				A	A	A	N/A
SEG16	4008 2110h	SEG16 (B-pattern area)				SEG16 (A-pattern area)				A	A	A	N/A
SEG17	4008 2111h	SEG17 (B-pattern area)				SEG17 (A-pattern area)				A	A	A	A
SEG18	4008 2112h	SEG18 (B-pattern area)				SEG18 (A-pattern area)				A	A	A	N/A
SEG19	4008 2113h	SEG19 (B-pattern area)				SEG19 (A-pattern area)				A	A	A	N/A
SEG20	4008 2114h	SEG20 (B-pattern area)				SEG20 (A-pattern area)				A	A	A	A
SEG21	4008 2115h	SEG21 (B-pattern area)				SEG21 (A-pattern area)				A	A	A	A
SEG22	4008 2116h	SEG22 (B-pattern area)				SEG22 (A-pattern area)				A	A	A	A
SEG23	4008 2117h	SEG23 (B-pattern area)				SEG23 (A-pattern area)				A	A	A	A
SEG24	4008 2118h	SEG24 (B-pattern area)				SEG24 (A-pattern area)				A	A	A	N/A
SEG25	4008 2119h	SEG25 (B-pattern area)				SEG25 (A-pattern area)				A	A	A	N/A
SEG26	4008 211Ah	SEG26 (B-pattern area)				SEG26 (A-pattern area)				A	N/A	N/A	N/A
SEG27	4008 211Bh	SEG27 (B-pattern area)				SEG27 (A-pattern area)				A	N/A	N/A	N/A
SEG28	4008 211Ch	SEG28 (B-pattern area)				SEG28 (A-pattern area)				A	A	A	N/A
SEG29	4008 211Dh	SEG29 (B-pattern area)				SEG29 (A-pattern area)				A	A	A	N/A
SEG30	4008 211Eh	SEG30 (B-pattern area)				SEG30 (A-pattern area)				A	A	A	N/A
SEG31	4008 211Fh	SEG31 (B-pattern area)				SEG31 (A-pattern area)				A	A	N/A	N/A
SEG32	4008 2120h	SEG32 (B-pattern area)				SEG32 (A-pattern area)				A	A	N/A	N/A
SEG33	4008 2121h	SEG33 (B-pattern area)				SEG33 (A-pattern area)				A	A	N/A	N/A
SEG34	4008 2122h	SEG34 (B-pattern area)				SEG34 (A-pattern area)				A	N/A	N/A	N/A
SEG35	4008 2123h	SEG35 (B-pattern area)				SEG35 (A-pattern area)				A	N/A	N/A	N/A
SEG36	4008 2124h	SEG36 (B-pattern area)				SEG36 (A-pattern area)				A	A	N/A	N/A
SEG37	4008 2125h	SEG37 (B-pattern area)				SEG37 (A-pattern area)				A	A	N/A	N/A
SEG38	4008 2126h	SEG38 (B-pattern area)				SEG38 (A-pattern area)				A	A	A	N/A
SEG39	4008 2127h	SEG39 (B-pattern area)				SEG39 (A-pattern area)				A	A	A	N/A
SEG40	4008 2128h	SEG40 (B-pattern area)				SEG40 (A-pattern area)				A	A	A	N/A
SEG41	4008 2129h	SEG41 (B-pattern area)				SEG41 (A-pattern area)				A	A	A	N/A

Table 49.9 Relationship between LCD Display Data Register contents and segment/common outputs (2 of 2)
Other than 8-time-slice (static, 2-time slice, 3-time slice, and 4-time slice)

Register name	Address	b7	b6	b5	b4	b3	b2	b1	b0	145/144-pin	121-pin	100-pin	64-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0				
SEG42	4008 212Ah	SEG42 (B-pattern area)				SEG42 (A-pattern area)				A	N/A	N/A	N/A
SEG43	4008 212Bh	SEG43 (B-pattern area)				SEG43 (A-pattern area)				A	N/A	N/A	N/A
SEG44	4008 212Ch	SEG44 (B-pattern area)				SEG44 (A-pattern area)				A	A	N/A	N/A
SEG45	4008 212Dh	SEG45 (B-pattern area)				SEG45 (A-pattern area)				A	A	N/A	N/A
SEG46	4008 212Eh	SEG46 (B-pattern area)				SEG46 (A-pattern area)				A	N/A	N/A	N/A
SEG47	4008 212Fh	SEG47 (B-pattern area)				SEG47 (A-pattern area)				A	N/A	N/A	N/A
SEG48	4008 2130h	SEG48 (B-pattern area)				SEG48 (A-pattern area)				A	A	A	A
SEG49	4008 2131h	SEG49 (B-pattern area)				SEG49 (A-pattern area)				A	A	A	A
SEG50	4008 2132h	SEG50 (B-pattern area)				SEG50 (A-pattern area)				A	A	A	A
SEG51	4008 2133h	SEG51 (B-pattern area)				SEG51 (A-pattern area)				A	A	A	N/A

A: Available
N/A: Not available

Table 49.10 Relationship between LCD Display Data Register contents and segment/common outputs (1 of 2)
8-time-slice

Register Name	Address	b7	b6	b5	b4	b3	b2	b1	b0	145/144-pin	121-pin	100-pin	64-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0				
SEG00	4008 2100h	SEG00								A	A	A	A
SEG01	4008 2101h	SEG01								A	A	A	N/A
SEG02	4008 2102h	SEG02								A	A	A	N/A
SEG03	4008 2103h	SEG03								A	A	N/A	N/A
SEG04	4008 2104h	SEG04								A	A	N/A	N/A
SEG05	4008 2105h	SEG05								A	A	N/A	N/A
SEG06	4008 2106h	SEG06								A	N/A	N/A	N/A
SEG07	4008 2107h	SEG07								A	N/A	N/A	N/A
SEG08	4008 2108h	SEG08								A	N/A	N/A	N/A
SEG09	4008 2109h	SEG09								A	N/A	N/A	N/A
SEG10	4008 210Ah	SEG10								A	N/A	N/A	N/A
SEG11	4008 210Bh	SEG11*1								A	N/A	N/A	N/A
SEG12	4008 210Ch	SEG12*1								A	A	N/A	N/A
SEG13	4008 210Dh	SEG13*1								A	A	N/A	N/A
SEG14	4008 210Eh	SEG14								A	A	A	N/A
SEG15	4008 210Fh	SEG15*1								A	A	A	N/A
SEG16	4008 2110h	SEG16								A	A	A	N/A
SEG17	4008 2111h	SEG17								A	A	A	A
SEG18	4008 2112h	SEG18								A	A	A	N/A
SEG19	4008 2113h	SEG19								A	A	A	N/A
SEG20	4008 2114h	SEG20								A	A	A	A
SEG21	4008 2115h	SEG21								A	A	A	A
SEG22	4008 2116h	SEG22								A	A	A	A
SEG23	4008 2117h	SEG23								A	A	A	A
SEG24	4008 2118h	SEG24								A	A	A	N/A
SEG25	4008 2119h	SEG25								A	A	A	N/A
SEG26	4008 211Ah	SEG26								A	N/A	N/A	N/A
SEG27	4008 211Bh	SEG27								A	N/A	N/A	N/A
SEG28	4008 211Ch	SEG28								A	A	A	N/A
SEG29	4008 211Dh	SEG29								A	A	A	N/A
SEG30	4008 211Eh	SEG30								A	A	A	N/A

Table 49.10 Relationship between LCD Display Data Register contents and segment/common outputs (2 of 2)
8-time-slice

Register Name	Address	b7	b6	b5	b4	b3	b2	b1	b0	145/144-pin	121-pin	100-pin	64-pin
		COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0				
SEG31	4008 211Fh	SEG31								A	A	N/A	N/A
SEG32	4008 2120h	SEG32								A	A	N/A	N/A
SEG33	4008 2121h	SEG33								A	A	N/A	N/A
SEG34	4008 2122h	SEG34								A	N/A	N/A	N/A
SEG35	4008 2123h	SEG35								A	N/A	N/A	N/A
SEG36	4008 2124h	SEG36								A	A	N/A	N/A
SEG37	4008 2125h	SEG37								A	A	N/A	N/A
SEG38	4008 2126h	SEG38								A	A	A	N/A
SEG39	4008 2127h	SEG39								A	A	A	N/A
SEG40	4008 2128h	SEG40								A	A	A	N/A
SEG41	4008 2129h	SEG41								A	A	A	N/A
SEG42	4008 212Ah	SEG42								A	N/A	N/A	N/A
SEG43	4008 212Bh	SEG43								A	N/A	N/A	N/A
SEG44	4008 212Ch	SEG44								A	A	N/A	N/A
SEG45	4008 212Dh	SEG45								A	A	N/A	N/A
SEG46	4008 212Eh	SEG46								A	N/A	N/A	N/A
SEG47	4008 212Fh	SEG47								A	N/A	N/A	N/A
SEG48	4008 2130h	SEG48								A	A	A	A
SEG49	4008 2131h	SEG49								A	A	A	A
SEG50	4008 2132h	SEG50								A	A	A	A
SEG51	4008 2133h	SEG51								A	A	A	N/A

A: Available

N/A: Not available

Note: All LCD display data registers (SEG00 to SEG51) have an initial value of 0h, and all bits that are readable and writable.

Note 1. The COM4 to COM7 pins and SEG00 to SEG03 pins are used alternatively. For more information, see section 20, I/O Ports.

When the number of time slices is static, two, three, or four, the lower four bits and upper four bits of each address of the LCD display data register become an A-pattern area and a B-pattern area, respectively.

The correspondences between A-pattern area data and COM signals are as follows:
bit 0 ↔ COM0, bit 1 ↔ COM1, bit 2 ↔ COM2, and bit 3 ↔ COM3.

The correspondences between B-pattern area data and COM signals are as follows:
bit 4 ↔ COM0, bit 5 ↔ COM1, bit 6 ↔ COM2, and bit 7 ↔ COM3.

A-pattern area data is displayed on the LCD panel when BLON = LCDSEL = 0 is selected, and B-pattern area data is displayed on the LCD panel when BLON = 0 and LCDSEL = 1 is selected.

49.4 Selection of LCD Display Data Register

When the number of time slices is static, two, three, or four, the LCD display data register can be selected from the following types, according to the BLON and LCDSEL bit settings:

- Displaying an A-pattern area data (lower 4 bits of LCD display data register)
- Displaying a B-pattern area data (upper 4 bits of LCD display data register)
- Alternately displaying A-pattern and B-pattern area data (blinking display associated with the periodic interrupt timing of the realtime clock (RTC)).

Note: If the normal liquid crystal waveform is displayed when the number of time slices is eight, LCD display data registers (A-pattern, B-pattern, or blinking display) cannot be selected.

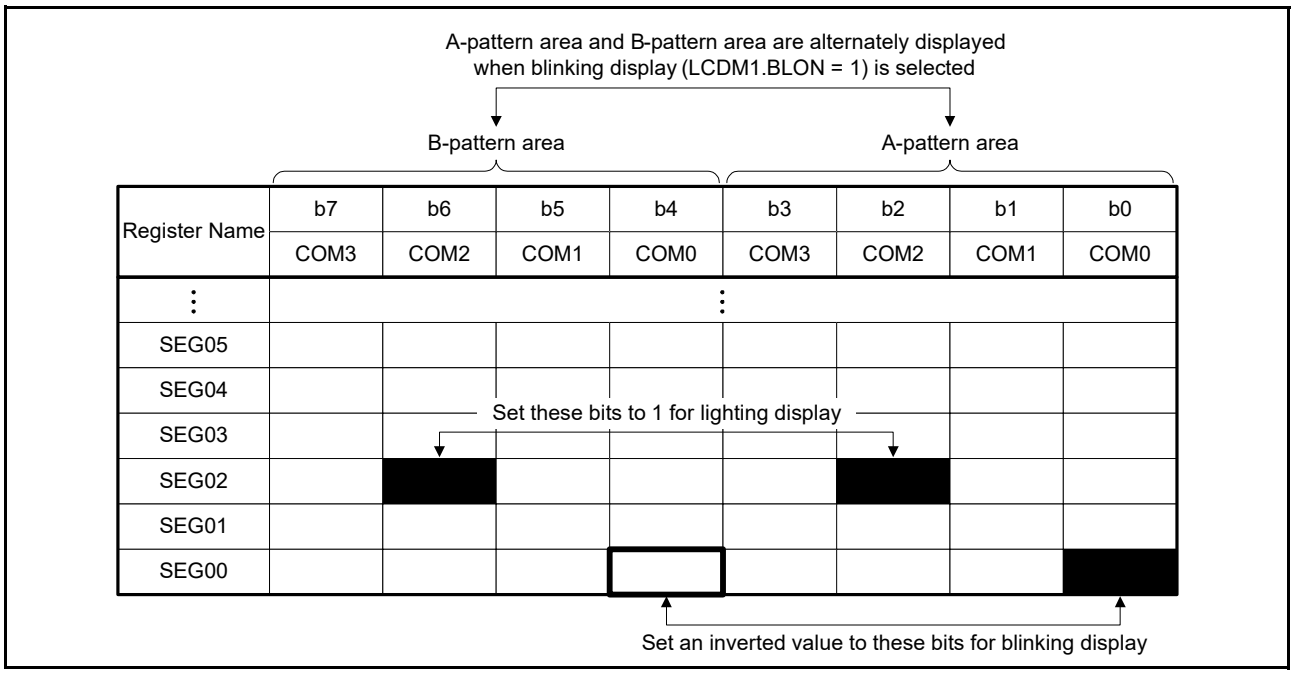


Figure 49.2 Example of setting LCD display data registers when the pattern is changed

49.4.1 A-Pattern Area and B-pattern Area Data Display

When both BLON and LCDSEL are 0, A-pattern area (lower four bits of the LCD display data register) data is output as the LCD display register.

When BLON is 0 and LCDSEL is 1, B-pattern area (upper four bits of the LCD display data register) data is output as the LCD display register.

For details on the display area, see section 49.3, LCD Display Data Registers.

49.4.2 Blinking Display (Alternately Displaying A-Pattern and B-Pattern Area Data)

When BLON is set to 1, A-pattern and B-pattern area data are alternately displayed, according to the constant-period interrupt timing of the realtime clock (RTC). See section 25, Realtime Clock (RTC) for information about the setting of the RTC constant-period interrupt (0.5 s setting only) timing.

To use the LCD blinking display feature, set inverted values to the B-pattern area bits associated with the A-pattern area bits. For example, set bit [0] of SEG00 register to 1, and set bit [4] of SEG00 register to 0 to use the blinking display. When not using the blinking display feature, set the same values to both the A-pattern and B-pattern area bits. For example, set bit [2] of SEG02 register to 1, and set bit [6] of SEG02 register to 1 for lighting display. For details on the display area, see section 49.3, LCD Display Data Registers.

Figure 49.3 and Figure 49.4 show the timing operation of display switching.

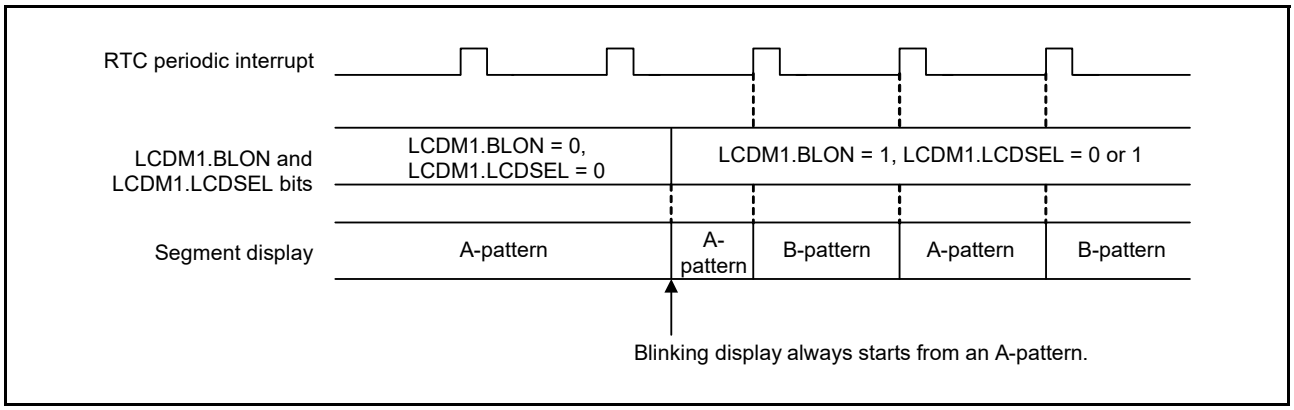


Figure 49.3 Switching operation from A-pattern display to blinking display

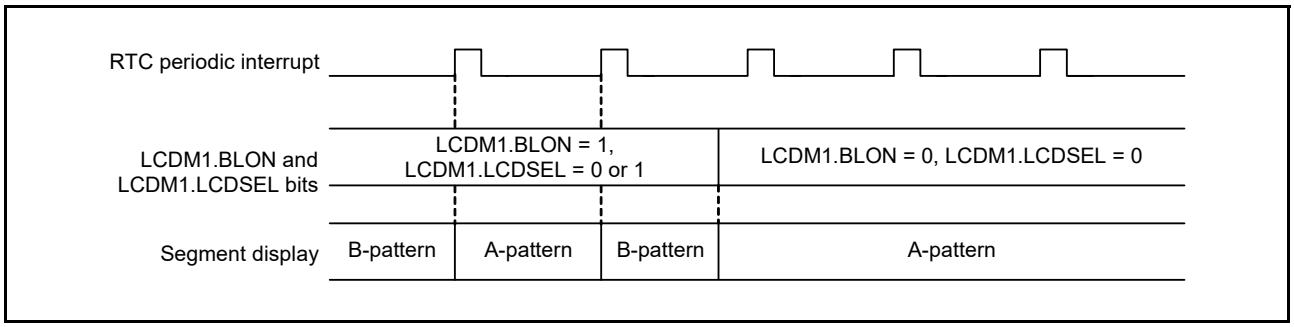


Figure 49.4 Switching operation from blinking display to A-pattern display

49.5 Setting LCD Controller/Driver

To operate the LCD controller/driver, follow procedures (1) to (3) in this section.Otherwise, the LCD operation is not guaranteed.

(1) External resistance division method during normal liquid crystal waveform display

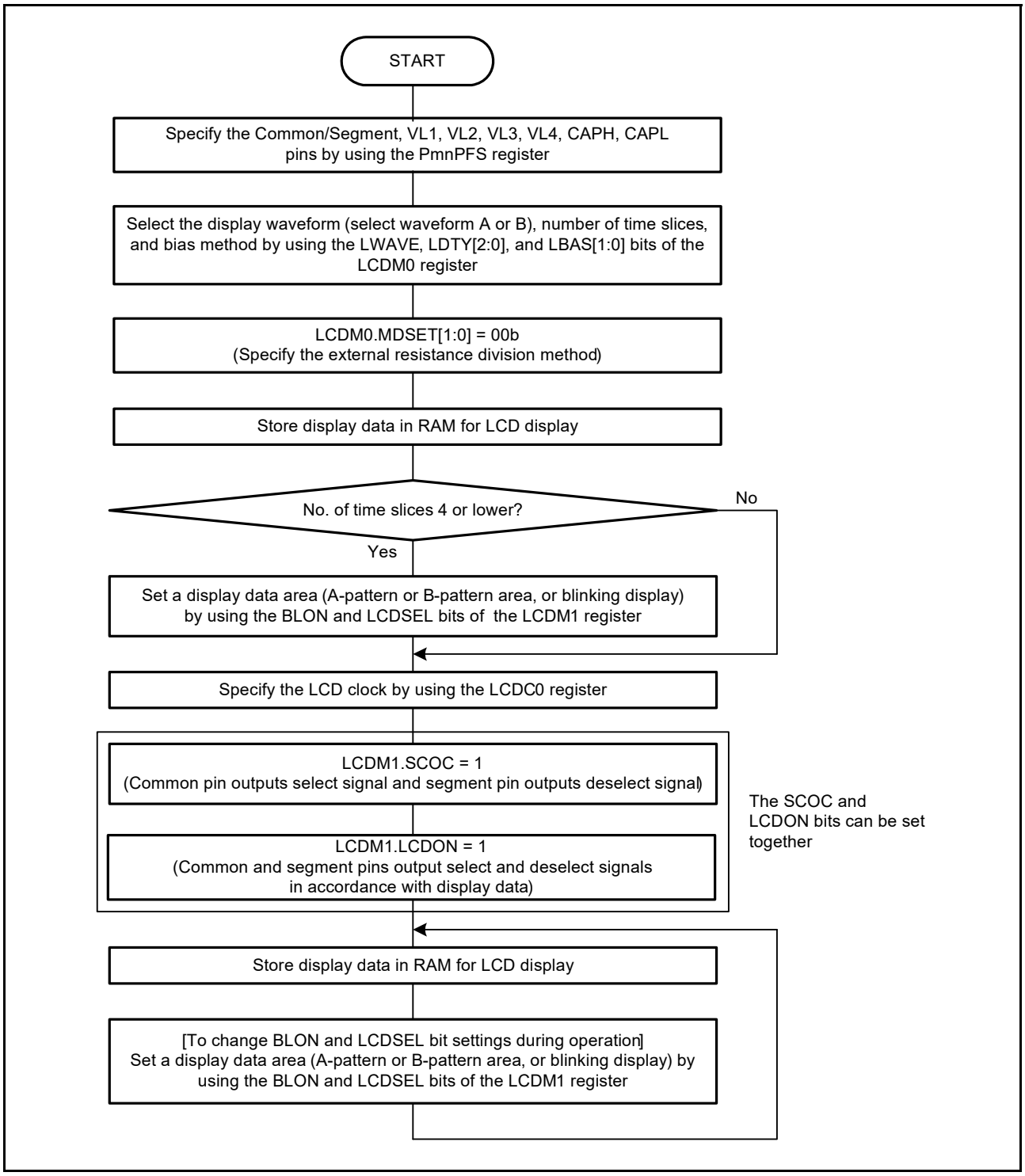


Figure 49.5 Setting procedure for external resistance division method during normal liquid crystal waveform display

(2) Internal voltage boosting method

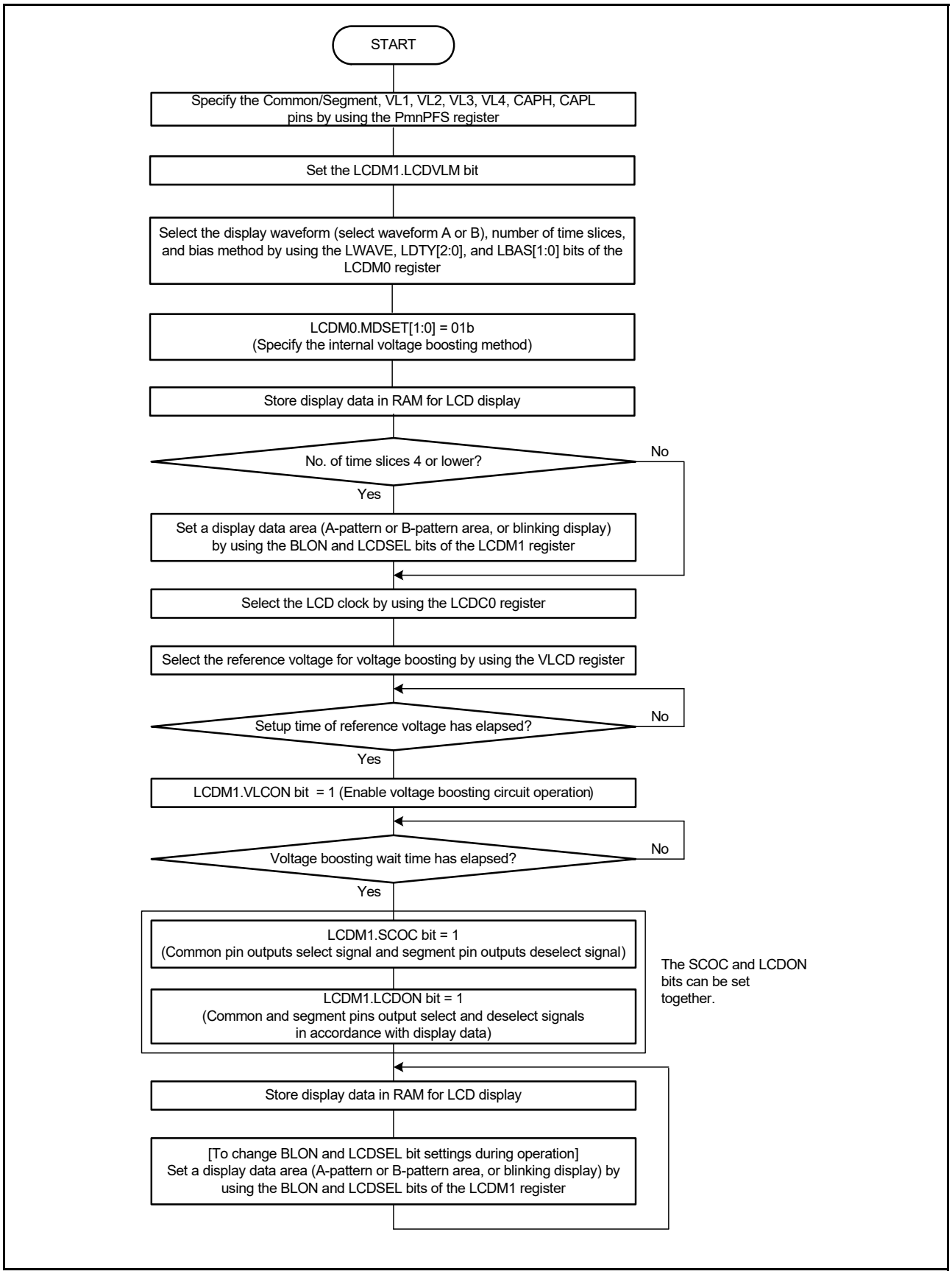


Figure 49.6 Setting procedure for internal voltage boosting method during normal liquid crystal waveform

display

(3) Capacitor split method

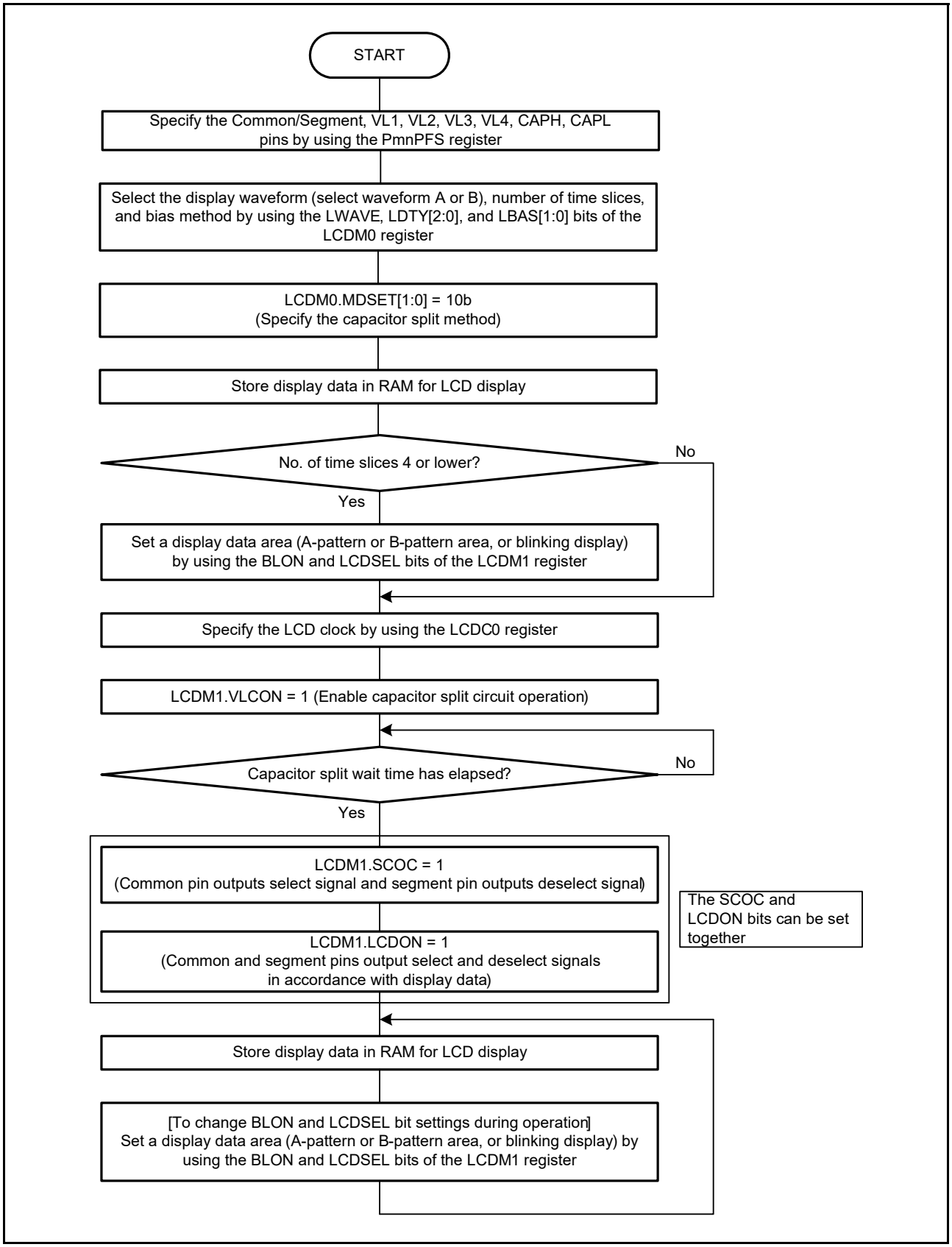
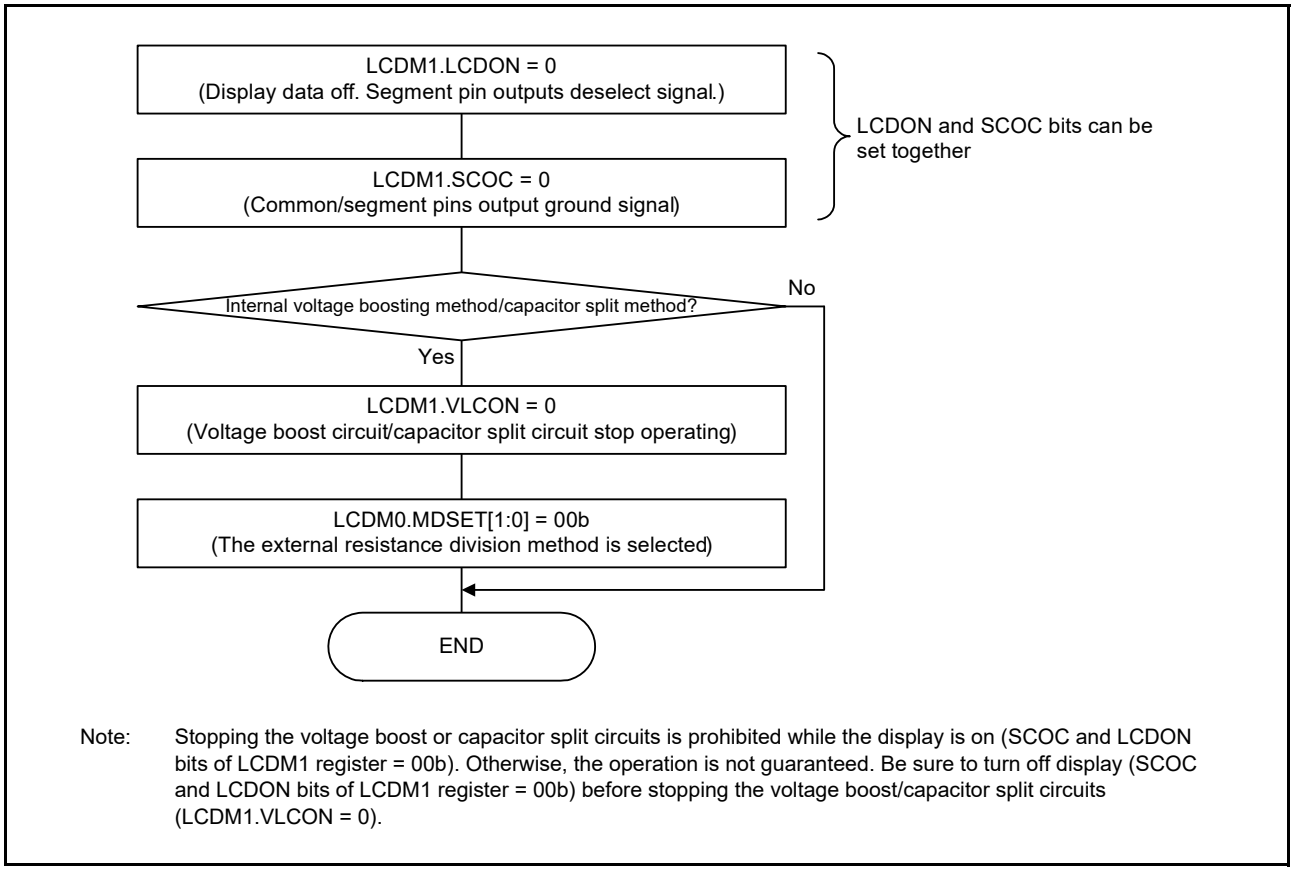


Figure 49.7 Setting procedure for capacitor split method during normal liquid crystal waveform display

49.6 Operation Stop Procedure

To stop the operation of the LCD, follow the steps shown in Figure 49.8.

The LCD stops operating when the LCDM1.LDCON and LCDM1.SCOC bits are set to 0.



Note: Stopping the voltage boost or capacitor split circuits is prohibited while the display is on (SCOC and LCDON bits of LCDM1 register = 00b). Otherwise, the operation is not guaranteed. Be sure to turn off display (SCOC and LCDON bits of LCDM1 register = 00b) before stopping the voltage boost/capacitor split circuits (LCDM1.VLCON = 0).

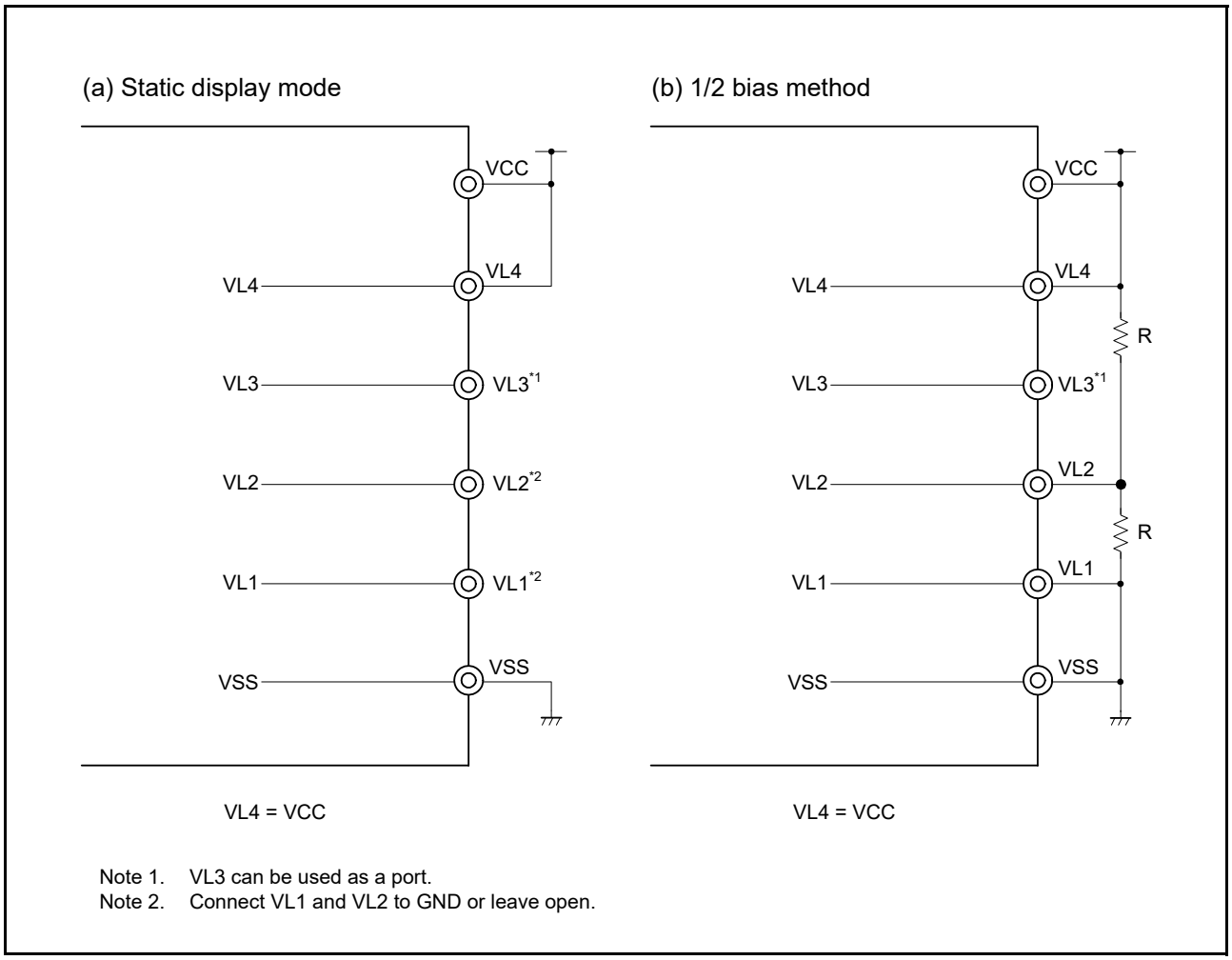
Figure 49.8 Operation stop procedure during normal liquid crystal waveform (A or B) display

49.7 Supplying LCD Drive Voltages VL1, VL2, VL3, and VL4

The power supply voltages for the LCD driver can be produced through external resistance division, internal voltage boosting, or capacitor split.

49.7.1 External Resistance Division Method

Figure 49.9 and Figure 49.10 show examples of LCD drive power supply connection, associated with each bias method.



Note 1. VL3 can be used as a port.
Note 2. Connect VL1 and VL2 to GND or leave open.

Figure 49.9 Examples of LCD drive power connections using external resistance division method (1/2)

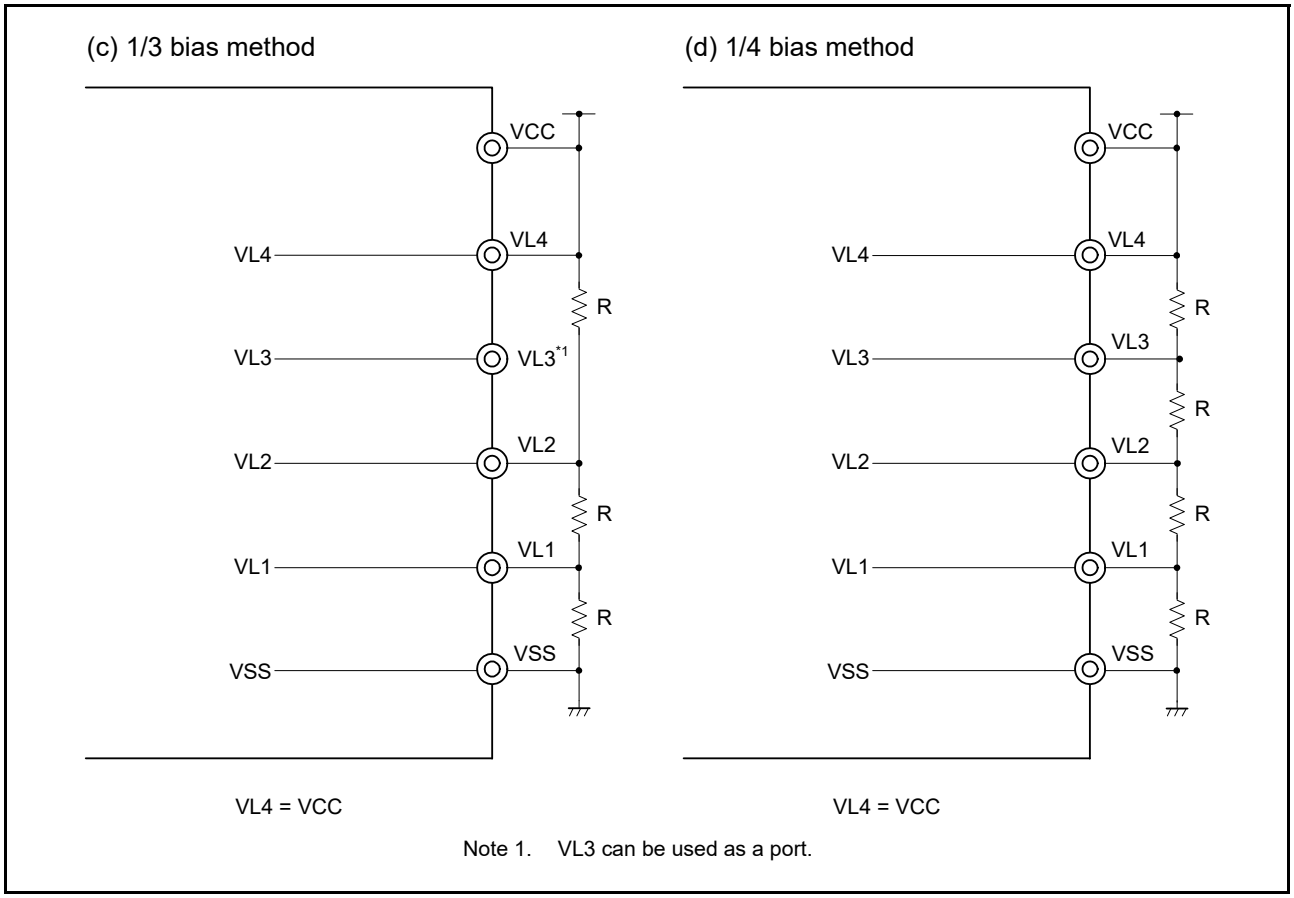


Figure 49.10 Examples of LCD drive power connections using external resistance division method (2/2)

Note: The reference resistance R value for external resistance division is 10 kΩ to 1 MΩ. In addition, to stabilize the voltage at the VL1 to VL4 pins, connect a capacitor between each pin VL1 to VL4 and the GND pin as needed. The reference capacitance is about 0.47 μF, but it depends on the LCD panel used, the number of segment pins, the number of common pins, the frame frequency, and the operating environment. Thoroughly evaluate these values in accordance with your system and adjust the capacitance.

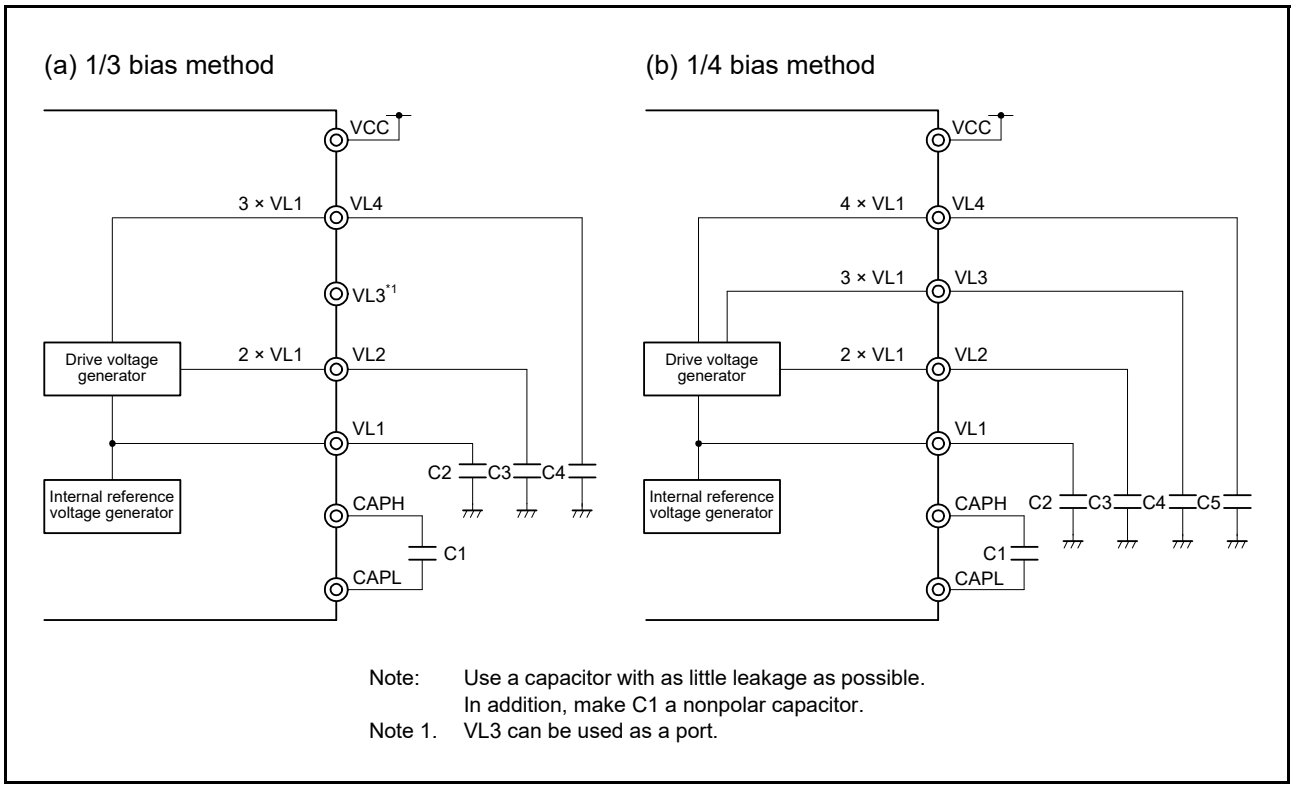
49.7.2 Internal Voltage Boosting Method

The MCU contains an internal voltage boost circuit for generating LCD drive power supplies. The internal voltage boost circuit and external capacitors (0.47 µF ±30%) are used to generate an LCD drive voltage. Only 1/3 bias mode or 1/4 bias mode can be set for the internal voltage boosting method.

The internal voltage boost circuit can supply a constant voltage, regardless of changes in VCC, because it is a power supply separate from the main unit. In addition, the contrast can be adjusted using the LCD boost level control register (VLCD).

Table 49.11 LCD drive voltages using internal voltage boosting method

LCD drive voltage pin	1/3 bias method	1/4 bias method
VL4	3 × VL1	4 × VL1
VL3	-	3 × VL1
VL2	2 × VL1	2 × VL1
VL1	LCD reference voltage	LCD reference voltage



Note: Use a capacitor with as little leakage as possible.
In addition, make C1 a nonpolar capacitor.
Note 1. VL3 can be used as a port.

Figure 49.11 Examples of LCD drive power connections using internal voltage boosting method

49.7.3 Capacitor Split Method

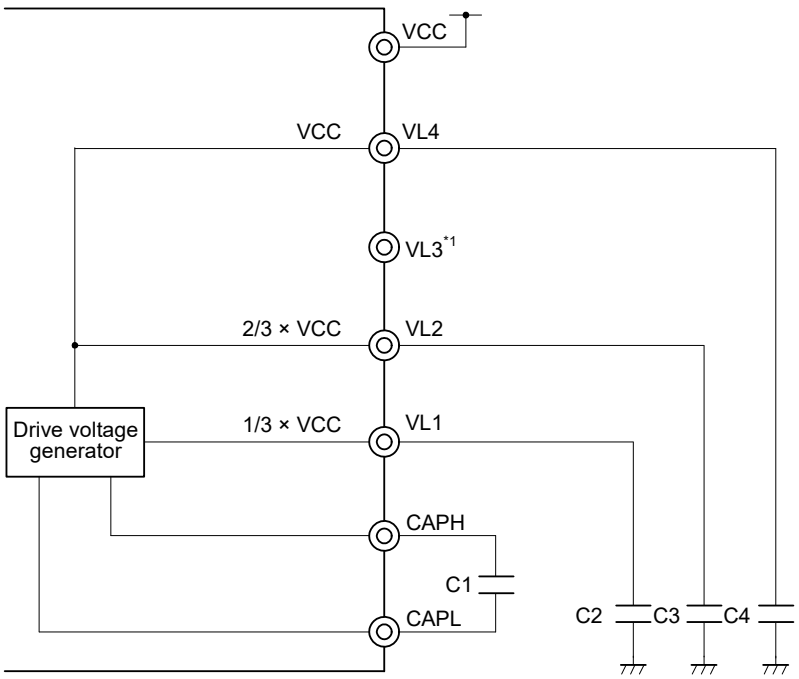
The MCU contains an internal voltage reduction circuit for generating LCD drive power supplies. The internal voltage reduction circuit and external capacitors (0.47 µF ±30%) are used to generate an LCD drive voltage. Only 1/3 bias mode can be set for the capacitor split method.

Unlike the external resistance division method, the capacitor split method does not require continuous current flow, and therefore current consumption can be reduced.

Table 49.12 LCD drive voltages using capacitor split method

LCD drive voltage pin	1/3 bias method
VL4	VCC
VL3	-
VL2	2/3 × VL4
VL1	1/3 × VL4

• 1/3 bias method



Note: Use a capacitor with less leakage current.
In addition, make C1 a non-polar capacitor.
Note 1. VL3 can be used as a port.

Figure 49.12 Examples of LCD drive power connections using capacitor split method

49.8 Common and Segment Signals

Each pixel of an LCD panel turns on when the potential difference between the corresponding common and segment signals becomes higher than a specific voltage (LCD drive voltage, VLCD). The pixels turn off when the potential difference becomes lower than VLCD.

Applying DC voltage to the common and segment signals of an LCD panel causes deterioration. To avoid this problem, the SLCDC is driven by AC voltage.

(1) Common signals

Each common signal is selected sequentially according to a specified number of time slices listed in Table 49.13. In the static display mode, the same signal is output to COM0 to COM3.

In the two-time-slice mode, leave the COM2 and COM3 pins open. In the three-time-slice mode, leave the COM3 pin open.

Use the COM4 to COM7 pins as open or segment pins except when operating in eight-time-slice mode.

Table 49.13 COM signal

Number of time slices	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
Static display mode					*1	*1	*1	*1
Two-time-slice mode			Open	Open	*1	*1	*1	*1
Three-time-slice mode				Open	*1	*1	*1	*1
Four-time-slice mode					*1	*1	*1	*1
Eight-time-slice mode								

Note 1. Use the pins as open or segment pins.

(2) Segment signals

The segment signals correspond to the LCD display data register (see section 49.3, LCD Display Data Registers).

When the number of time slices is eight, bits 0 to 7 of each display data register are read in synchronization with COM0 to COM7, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins.

When the number of time slices is number other than eight, bits 0 to 3 of each byte in A-pattern area are read in synchronization with COM0 to COM3, and bits 4 to 7 of each byte in B-pattern area are read in synchronization with COM0 to COM3, respectively. If a bit is 1, it is converted to the select voltage, and if it is 0, it is converted to the deselect voltage. The conversion results are output to the segment pins.

Check what combination of front-surface electrodes (associated with the segment signals) and rear-surface electrodes (associated with the common signals) forms display patterns in the LCD display data register, and write the bit data that associated with the desired display pattern on a one-to-one basis.

Note: The mounted segment output pins vary depending on the product.

(3) Output waveforms of common and segment signals

The voltages listed in Table 49.14 are output as common and segment signals.

When both common and segment signals are at the select voltage, display on-voltage is ±VLCD. Other combinations of the signals correspond to display off-voltage.

Table 49.14 LCD drive voltage

Static display mode

	Segment signal	
	Select signal level	**Deselect signal level**
Common signal	**VSS/VL4**	**VL4/VSS**
VL4/VSS	–VLCD/+VLCD	0 V/0 V

1/2 bias method

		Segment signal	
		Select signal level	**Deselect signal level**
Common signal		**VSS/VL4**	**VL4/VSS**
Select Signal Level	VL4/VSS	–VLCD/+VLCD	0 V/0 V
Deselect Signal Level	VL2	$-\frac{1}{2}$VLCD/+$\frac{1}{2}$VLCD	+$\frac{1}{2}$VLCD/–$\frac{1}{2}$VLCD

1/3 bias method (waveform A or B)

		Segment signal	
		Select signal level	**Deselect signal level**
Common signal		**VSS/VL4**	**VL2/VL1**
Select Signal Level	VL4/VSS	–VLCD/+VLCD	$-\frac{1}{3}$VLCD/+$\frac{1}{3}$VLCD
Deselect Signal Level	VL1/VL2	$-\frac{1}{3}$VLCD/+$\frac{1}{3}$VLCD	+$\frac{1}{3}$VLCD/–$\frac{1}{3}$VLCD

1/4 bias method (waveform A or B)

		Segment signal	
		Select signal level	**Deselect signal level**
Common signal		**VSS/VL4**	**VL2**
Select Signal Level	VL4/VSS	–VLCD/+VLCD	$-\frac{1}{2}$VLCD/+$\frac{1}{2}$VLCD
Deselect Signal Level	VL1/VL3	$-\frac{1}{4}$VLCD/+$\frac{1}{4}$VLCD	+$\frac{1}{4}$VLCD/–$\frac{1}{4}$VLCD

Figure 49.13 and Figure 49.14 show the common signal waveforms. Figure 49.15 to Figure 49.17 show the voltages and phases of the common and segment signals.

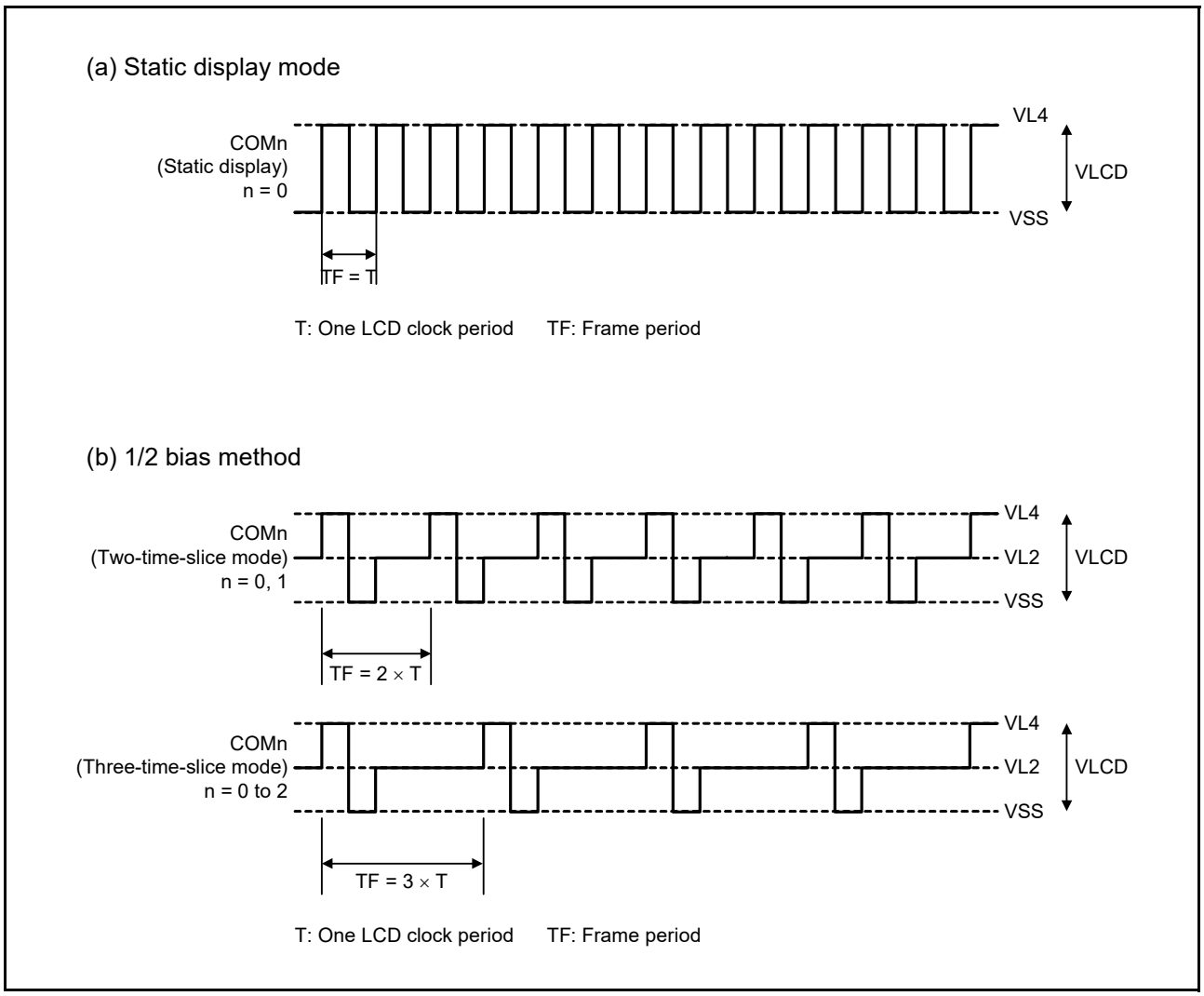
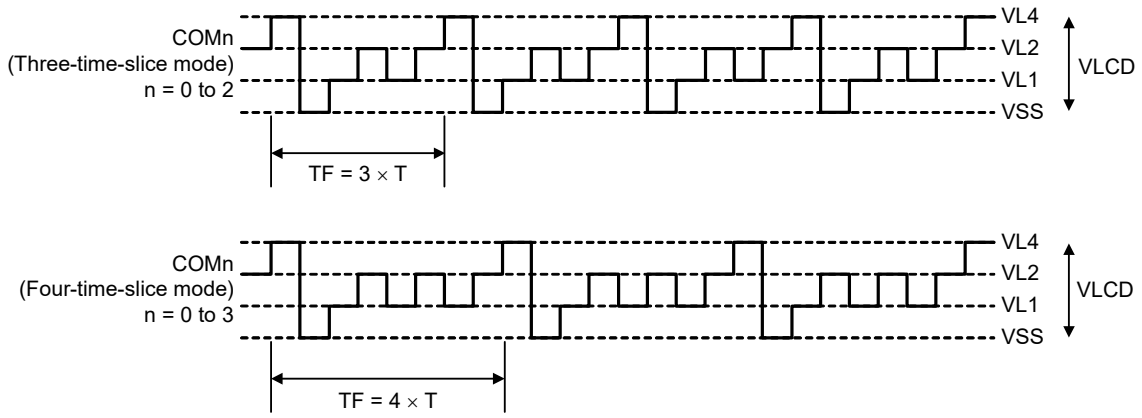


Figure 49.13 Common signal waveforms (1/2)

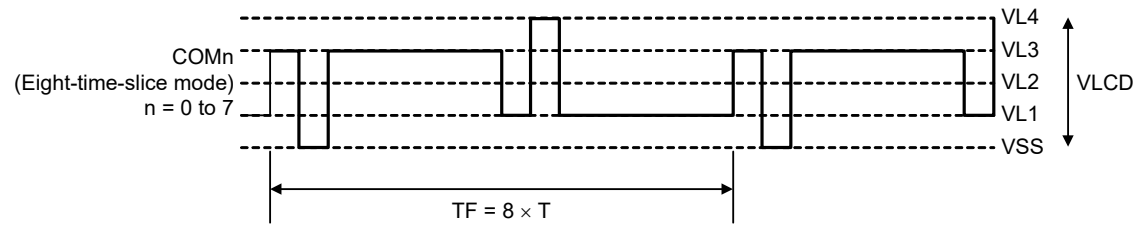
(c) 1/3 bias method



T: One LCD clock period TF: Frame period

< Example of calculation of LCD frame frequency (When four-time slice mode is used) >
LCD clock: $32768/2^7$ = 256 Hz (When setting to LCDC0 register = 06h)
LCD frame frequency: 64 Hz

(d) 1/4 bias method

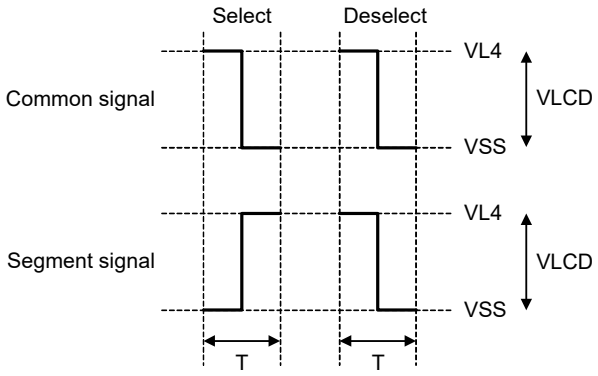


T: One LCD clock period TF: Frame period

< Example of calculation of LCD frame frequency (When eight-time slice mode is used) >
LCD clock: $32768/2^7$ = 256 Hz (When setting to LCDC0 register = 06h)
LCD frame frequency: 32 Hz

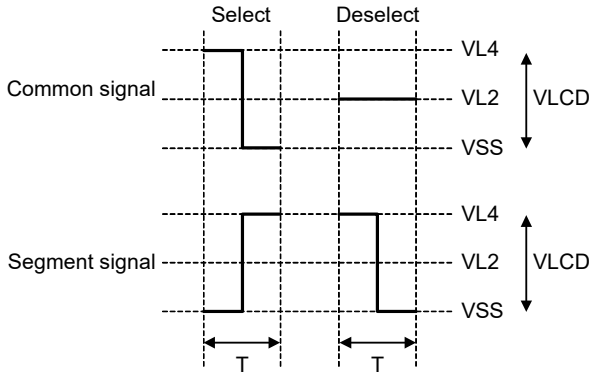
Figure 49.14 Common signal waveforms (2/2)

(a) Static display mode (waveform A)



T: One LCD clock period

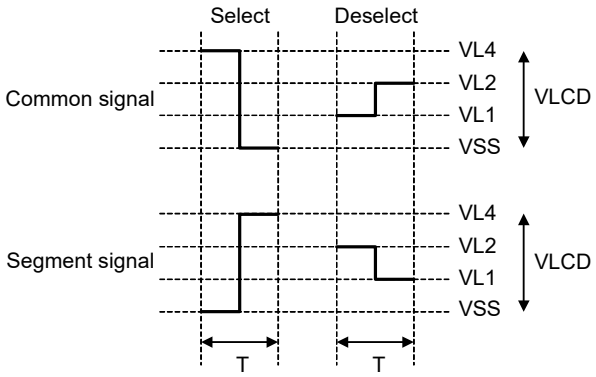
(b) 1/2 bias method (waveform A)



T: One LCD clock period

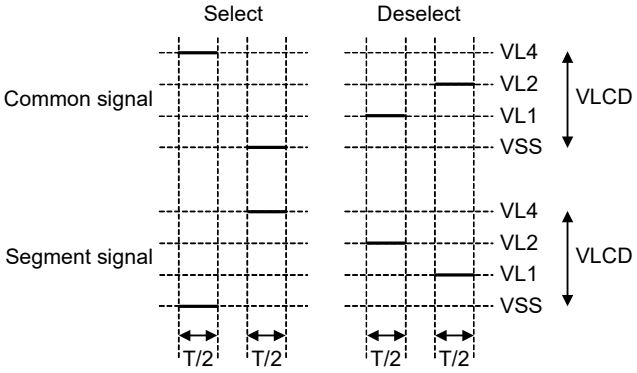
Figure 49.15 Voltages and phases of common and segment signals (1/3)

(c) 1/3 bias method (waveform A)



T: One LCD clock period

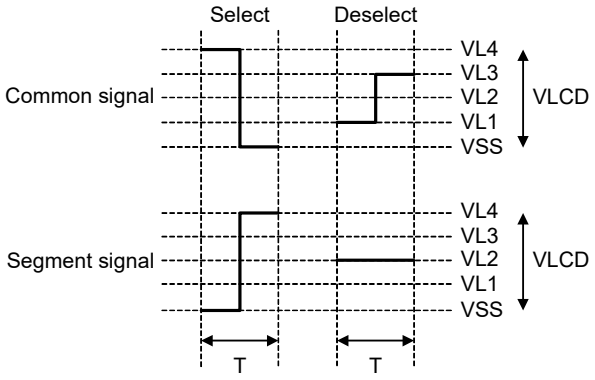
(d) 1/3 bias method (waveform B)



T: One LCD clock period

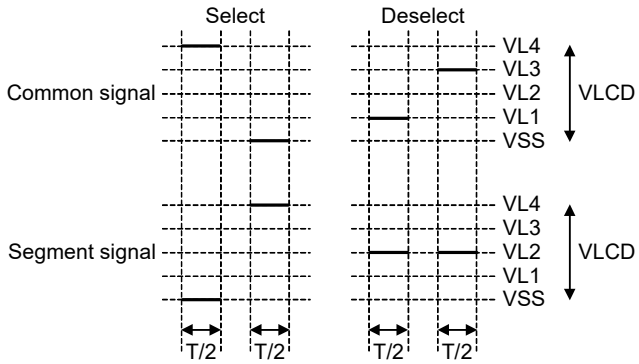
Figure 49.16 Voltages and phases of common and segment signals (2/3)

(e) 1/4 bias method (waveform A)



T: One LCD clock period

(f) 1/4 bias method (waveform B)



T: One LCD clock period

Figure 49.17 Voltages and phases of common and segment signals (3/3)

49.9 Display Modes

49.9.1 Static Display Example

Figure 49.19 shows how a three-digit LCD panel with the display pattern shown in Figure 49.18 is connected to the segment signals (SEG00 to SEG23) and the common signal (COM0). This example displays "12.3" in the LCD panel. The contents of the display data register correspond to this display.

The following description focuses on numeral "2." (2.) displayed in the second digit. To display "2." in the LCD panel, the select or deselect voltage must be applied to the SEG08 to SEG15 pins at the select timing of the common signal COM0. See Figure 49.18 for the relationship between the segment signals and LCD segments.

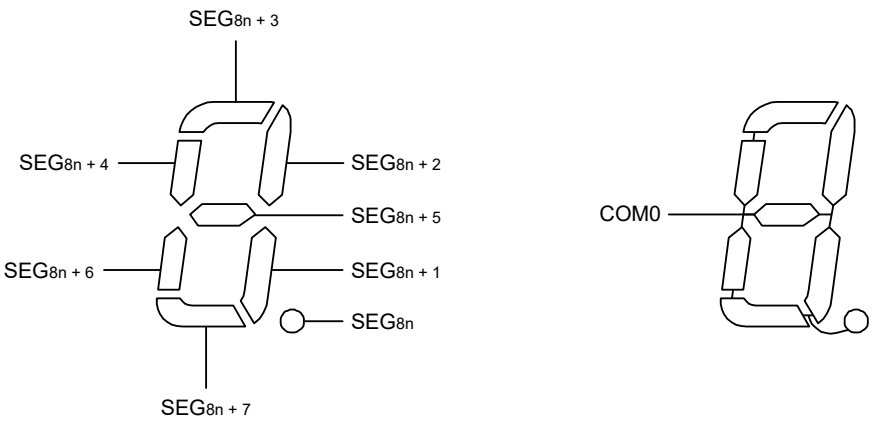
Table 49.15 Example of select (1) and deselect (0) data (COM0)

Common	Segment							
	SEG08	SEG09	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15
COM0	Select	Deselect	Select	Select	Deselect	Select	Select	Select

According to Table 49.15, the bit-0 pattern of the display data register must be 10110111b.

Figure 49.20 shows the LCD drive waveforms of SEG11 and SEG12, and COM0. When the select voltage is applied to SEG11 at the timing of COM0, an alternating rectangle waveform, +VLCD/–VLCD, is generated to turn on the corresponding LCD segment.

COM1 to COM3 are supplied with the same waveform as COM0. Therefore, COM0 to COM3 can be connected together to increase the driving capacity.



Note: The mounted segment output pins vary depending on the product.

Figure 49.18 Static LCD display pattern and electrode connections

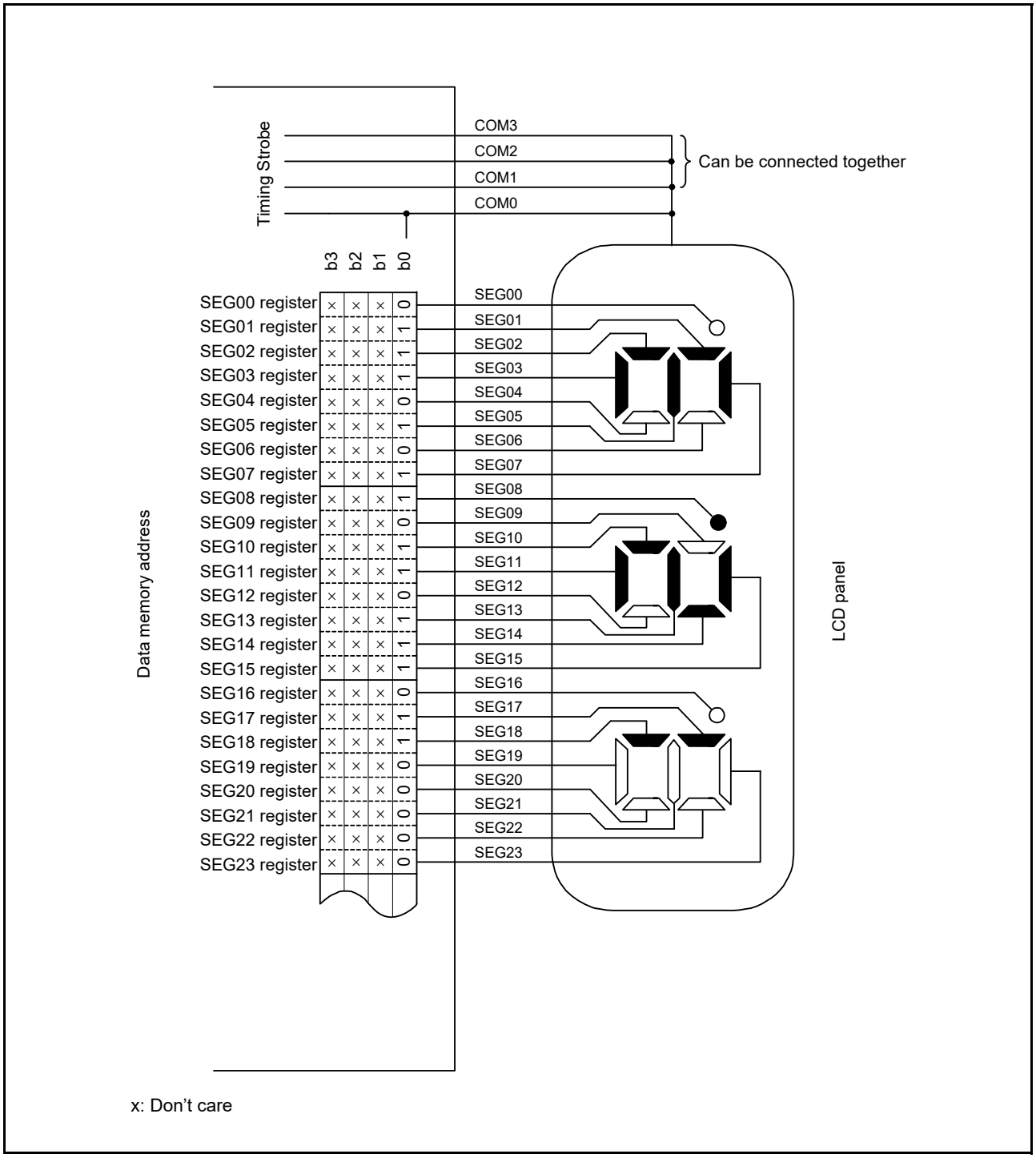


Figure 49.19 Example of connecting static LCD panel

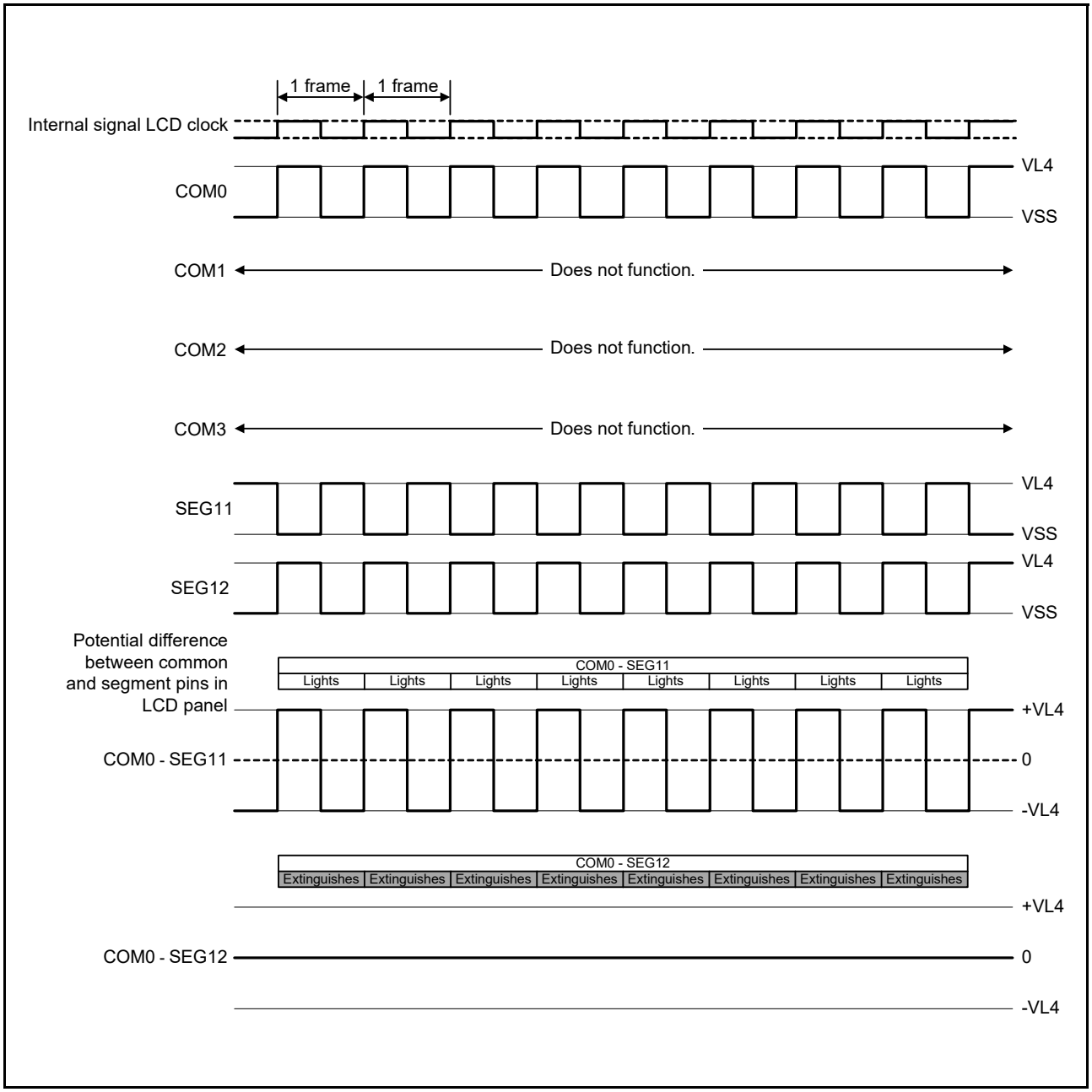


Figure 49.20 Static LCD drive waveform examples for SEG11, SEG12, and COM0

49.9.2 Two-Time-Slice Display Example

Figure 49.22 shows how a 6-digit LCD panel with the display pattern shown in Figure 49.21 is connected to the segment signals (SEG00 to SEG23) and the common signals (COM0 and COM1). This example displays "12345.6" in the LCD panel. The contents of the display data register correspond to this display.

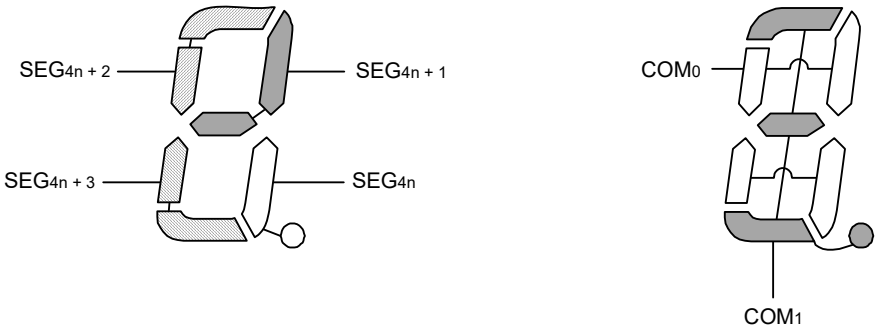
The following description focuses on numeral "3" (3) displayed in the fourth digit. To display "3" in the LCD panel, the select or deselect voltage must be applied to the SEG12 to SEG15 pins at the select timing of the common signals COM0 and COM1. See Figure 49.21 for the relationship between the segment signals and LCD segments.

Table 49.16 Example of select (1) and deselect (0) data (COM0 and COM1)

	Segment			
Common	**SEG12**	**SEG13**	**SEG14**	**SEG15**
COM0	Select	Select	Deselect	Deselect
COM1	Deselect	Select	Select	Select

According to Table 49.16, the display data register location that corresponds to SEG15 must contain "xx10b".

Figure 49.23 shows examples of LCD drive waveforms between the SEG15 signal and each common signal. When the select voltage is applied to SEG15 at the timing of COM1, an alternating rectangle waveform, +VLCD/–VLCD, is generated to turn on the associated LCD segment.



Note: The mounted segment output pins vary depending on the product.

Figure 49.21 Two-time-slice LCD display pattern and electrode connections

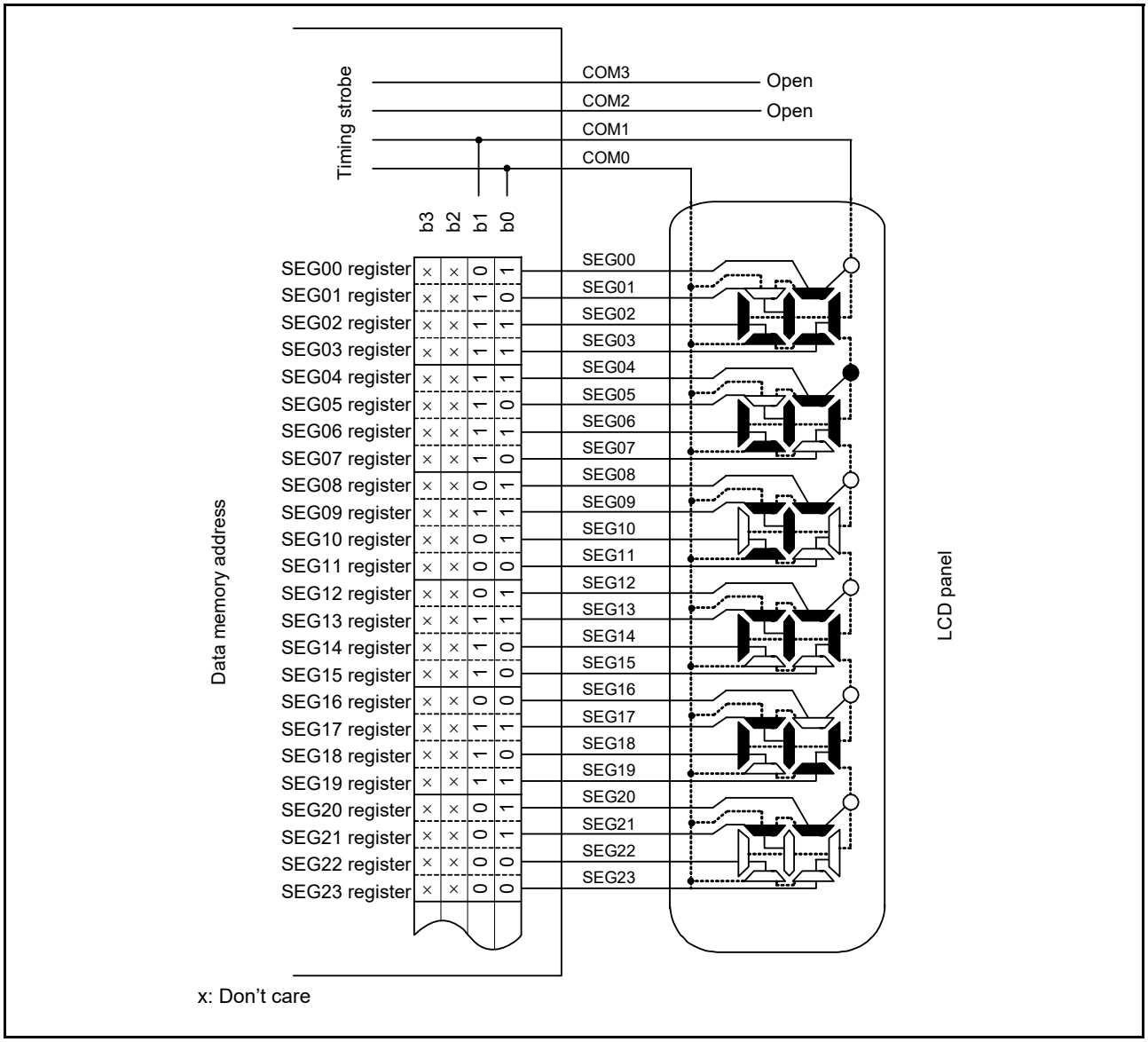


Figure 49.22 Example of connecting two-time-slice LCD panel

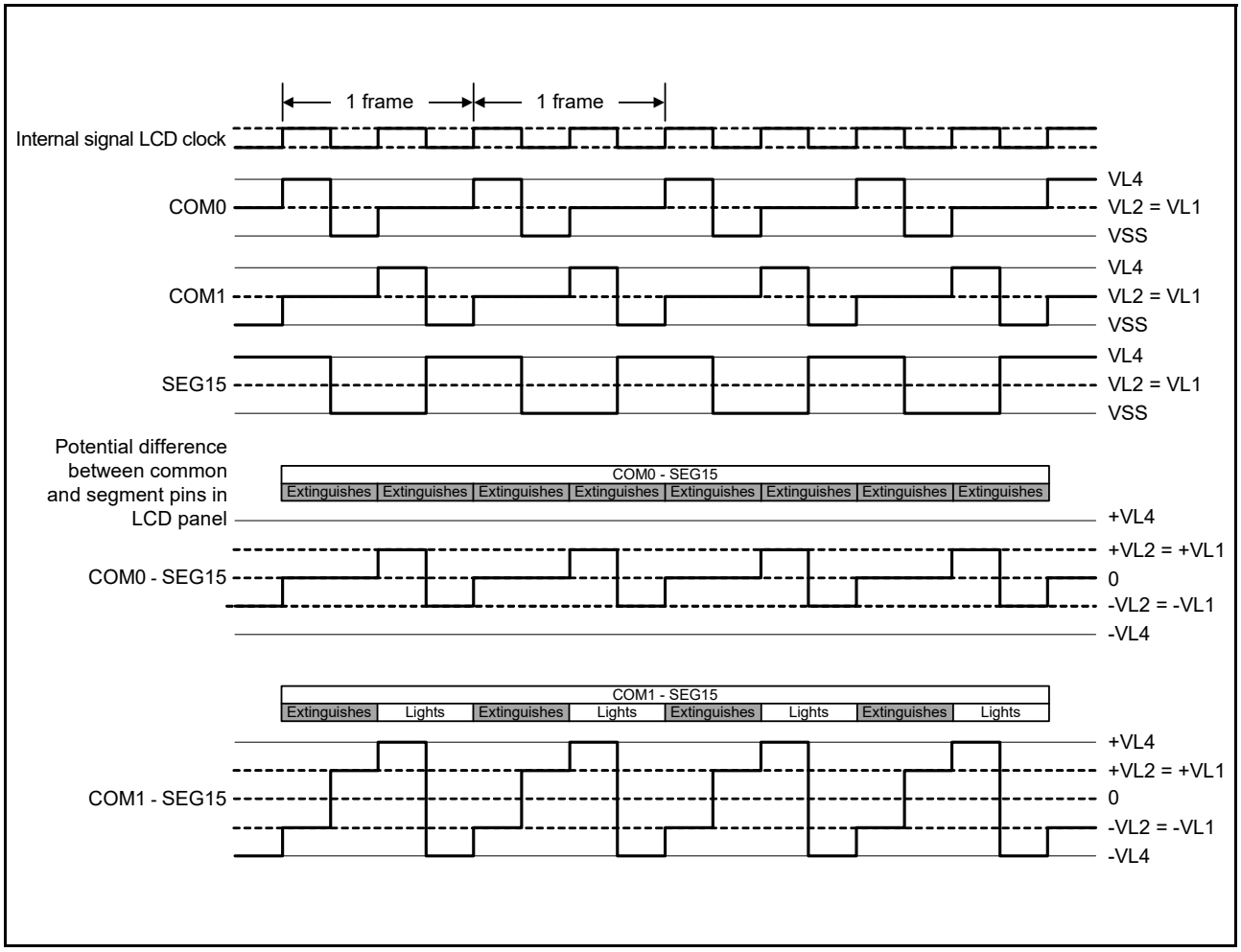


Figure 49.23 Two-time-slice LCD drive waveform examples between SEG15 and each common signals using 1/2 bias method

49.9.3 Three-Time-Slice Display Example

Figure 49.25 shows how an 8-digit LCD panel with the display pattern shown in Figure 49.24 is connected to the segment signals (SEG00 to SEG23) and the common signals (COM0 to COM2). This example displays "123456.78" in the LCD panel. The contents of the display data register correspond to this display.

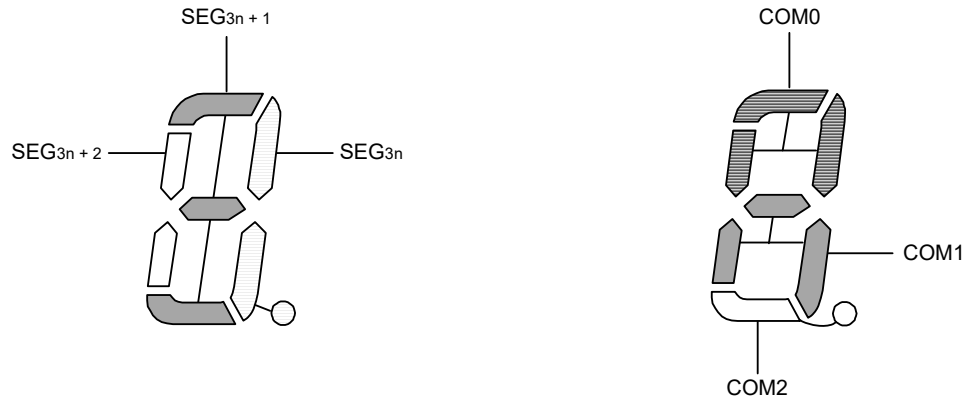
The following description focuses on numeral "6." (6.) displayed in the third digit. To display "6." in the LCD panel, the select or deselect voltage must be applied to the SEG06 to SEG08 pins at the select timing of the common signals COM0 to COM2. See Figure 49.24 for the relationship between the segment signals and LCD segments.

Table 49.17 Example of select (1) and deselect (0) data (COM0 to COM2)

Common	Segment		
	SEG06	**SEG07**	**SEG08**
COM0	Deselect	Select	Select
COM1	Select	Select	Select
COM2	Select	Select	—

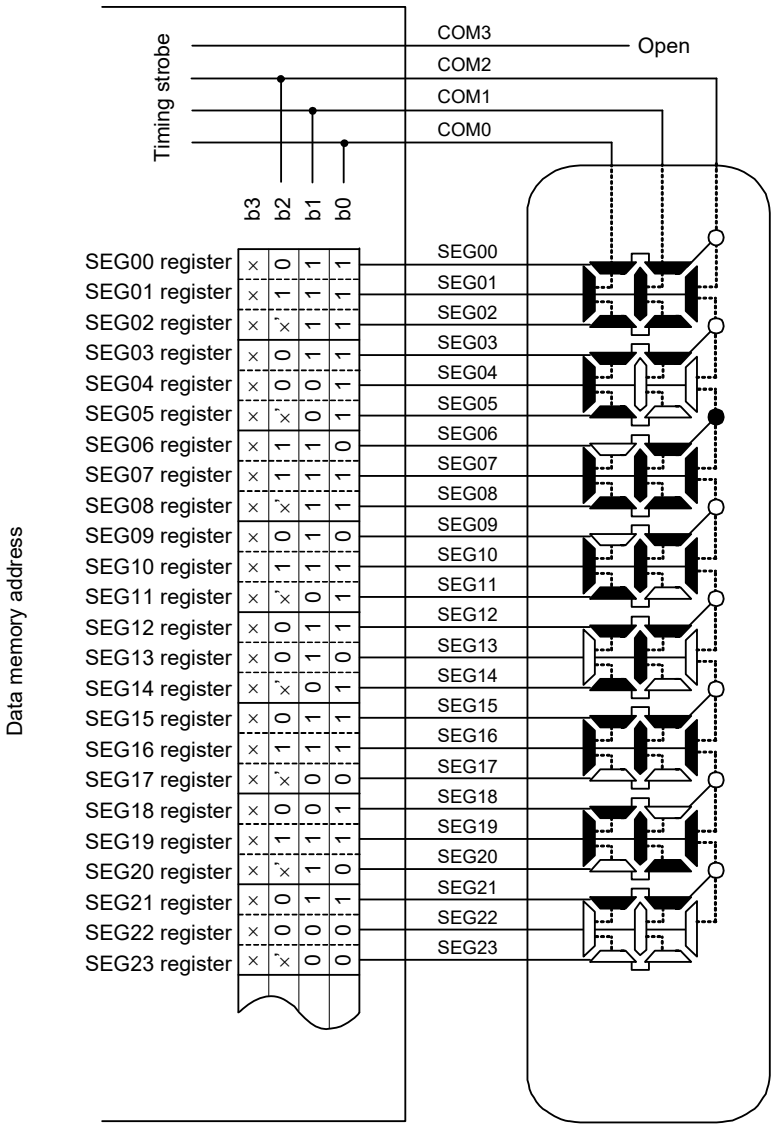
According to Table 49.17, the display data register location that corresponds to SEG06 must contain "x110b".

Figure 49.26 and Figure 49.27 show examples of LCD drive waveforms between the SEG06 signal and each common signal in the 1/2 and 1/3 bias methods, respectively. When the select voltage is applied to SEG06 at the timing of COM1 or COM2, an alternating rectangle waveform, +VLCD/–VLCD, is generated to turn on the associated LCD segment.



Note: The mounted segment output pins vary depending on the product.

Figure 49.24 Three-time-slice LCD display pattern and electrode connections



x: Don't care.
x': Can be used to store any data because there is no corresponding segment in the LCD

Figure 49.25 Example of connecting three-time-slice LCD panel

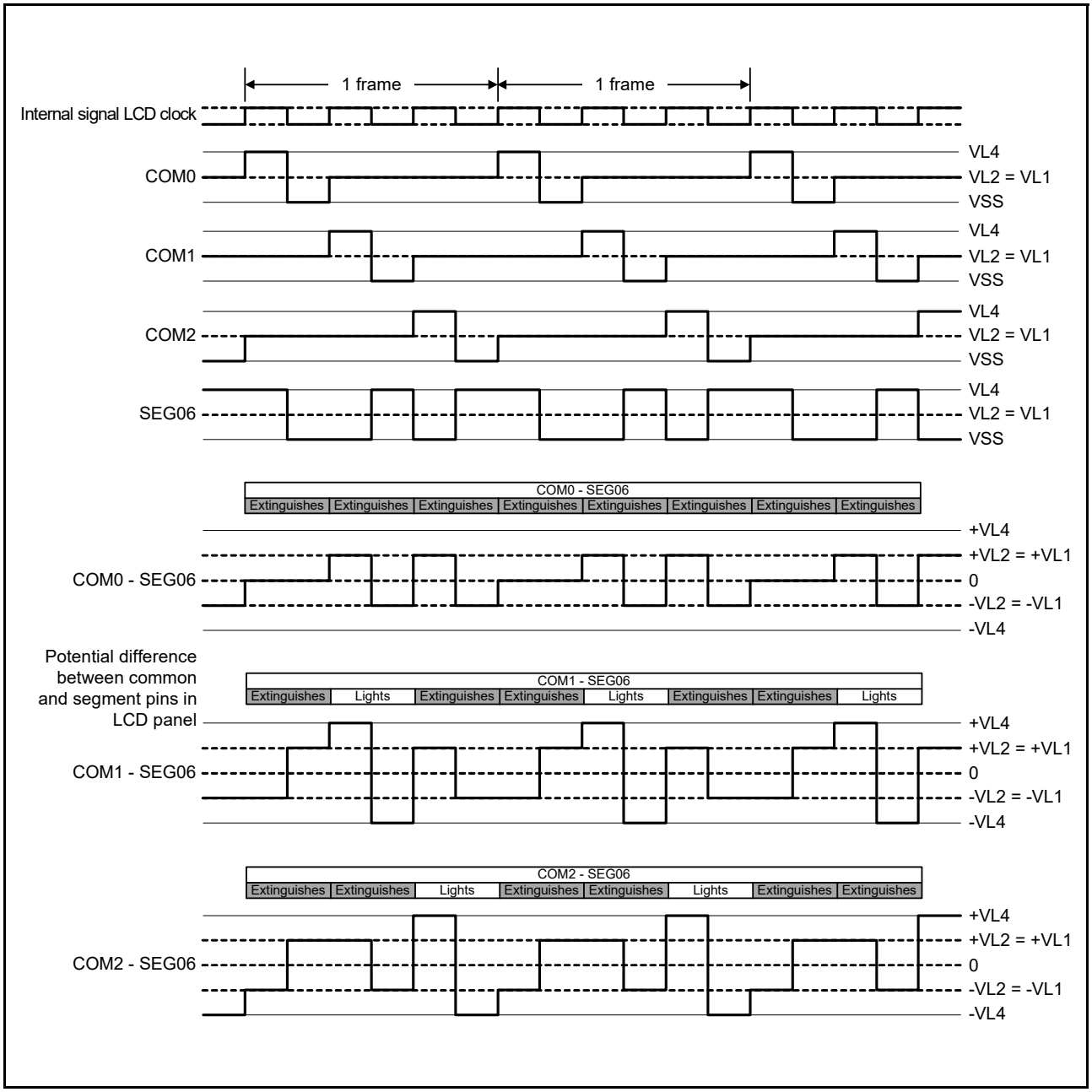


Figure 49.26 Three-time-slice LCD drive waveform examples between SEG06 and each common signals using 1/2 bias method

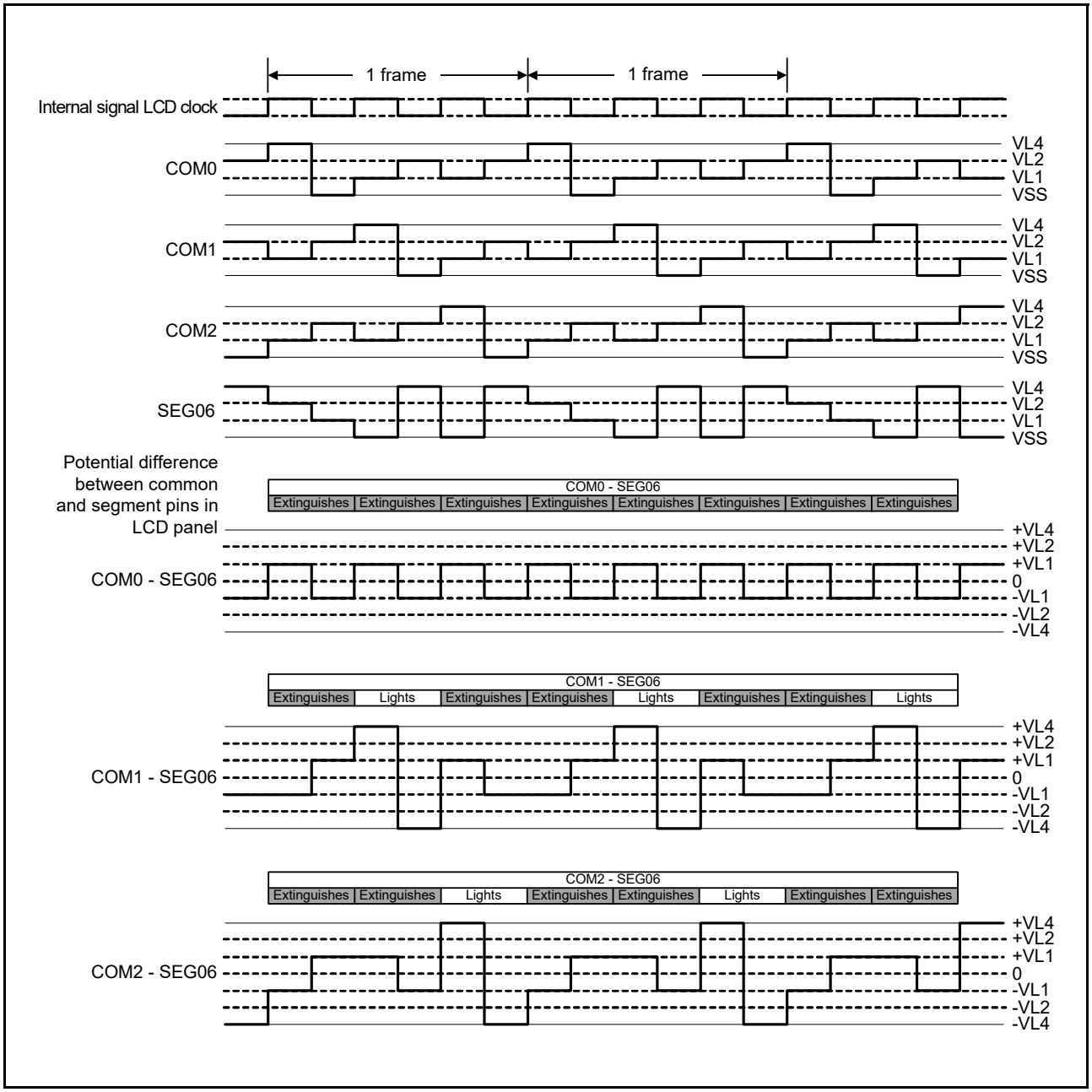


Figure 49.27 Three-time-slice LCD drive waveform examples between SEG06 and each common signals using 1/3 bias method

49.9.4 Four-Time-Slice Display Example

Figure 49.29 shows how a 12-digit LCD panel with the display pattern shown in Figure 49.28 is connected to the segment signals (SEG00 to SEG23) and the common signals (COM0 to COM3). This example displays "123456.789012" in the LCD panel. The contents of the display data register correspond to this display.

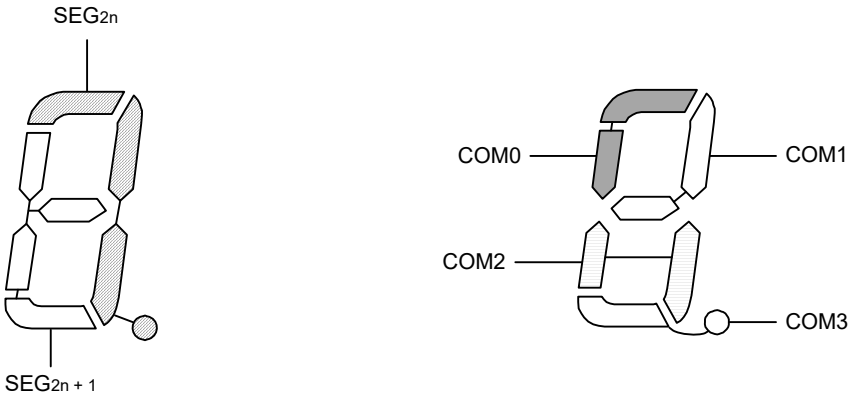
The following description focuses on numeral "6." (6.) displayed in the seventh digit. To display "6." in the LCD panel, the select or deselect voltage must be applied to the SEG12 and SEG13 pins at the select timing of the common signals COM0 to COM3. See Figure 49.28 for the relationship between the segment signals and LCD segments.

Table 49.18 Example of select (1) and deselect (0) data (COM0 to COM3)

	Segment	
Common	**SEG12**	**SEG13**
COM0	Select	Select
COM1	Deselect	Select
COM2	Select	Select
COM3	Select	Select

According to Table 49.18, the display data register location that corresponds to SEG12 must contain "1101b".

Figure 49.30 shows examples of LCD drive waveforms between the SEG12 signal and each common signal. When the select voltage is applied to SEG12 at the timing of COM0, an alternating rectangle waveform, +VLCD/–VLCD, is generated to turn on the associated LCD segment.



Note: The mounted segment output pins vary depending on the product.

Figure 49.28 Four-time-slice LCD display pattern and electrode connections

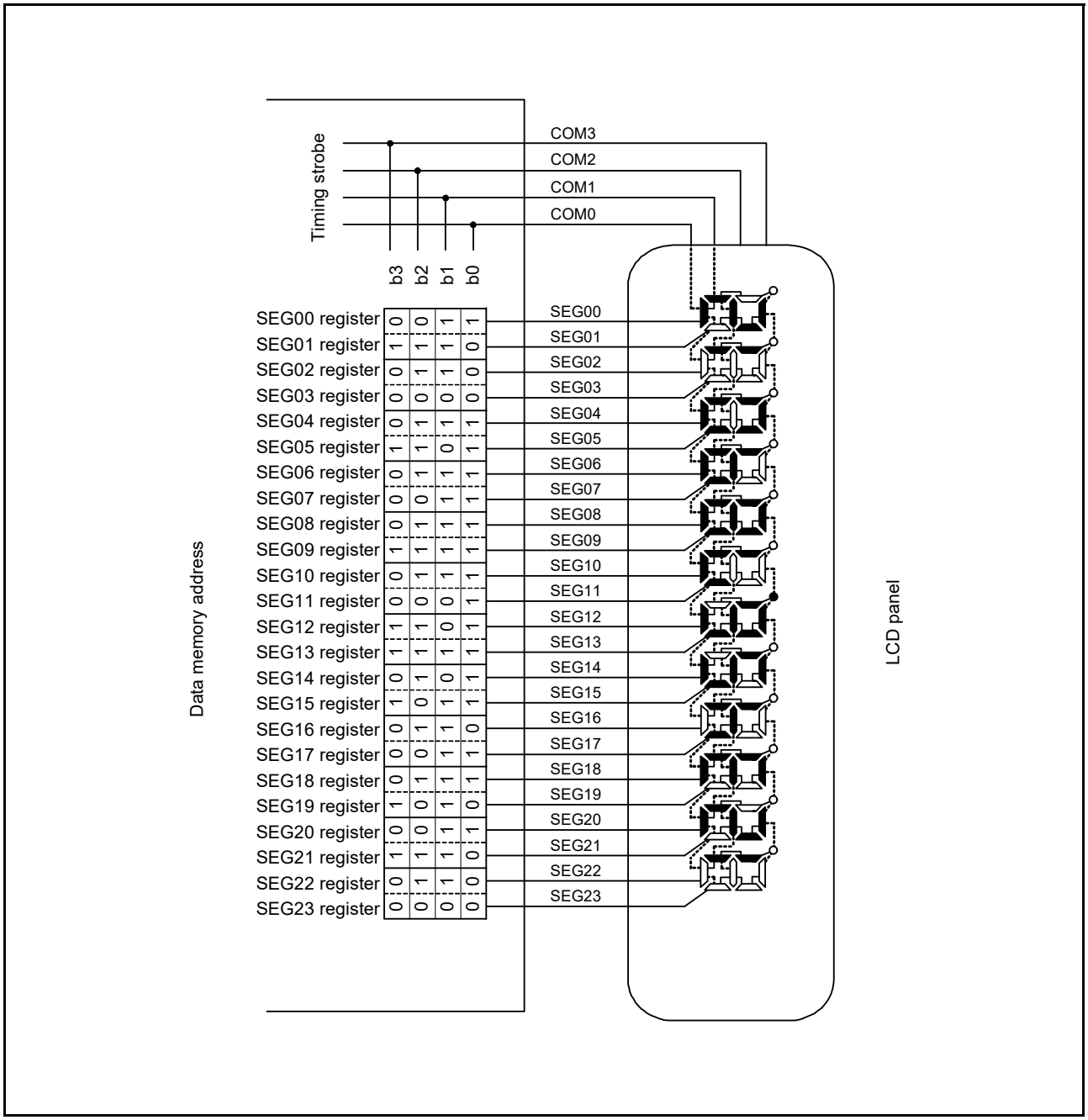


Figure 49.29 Example of connecting four-time-slice LCD panel

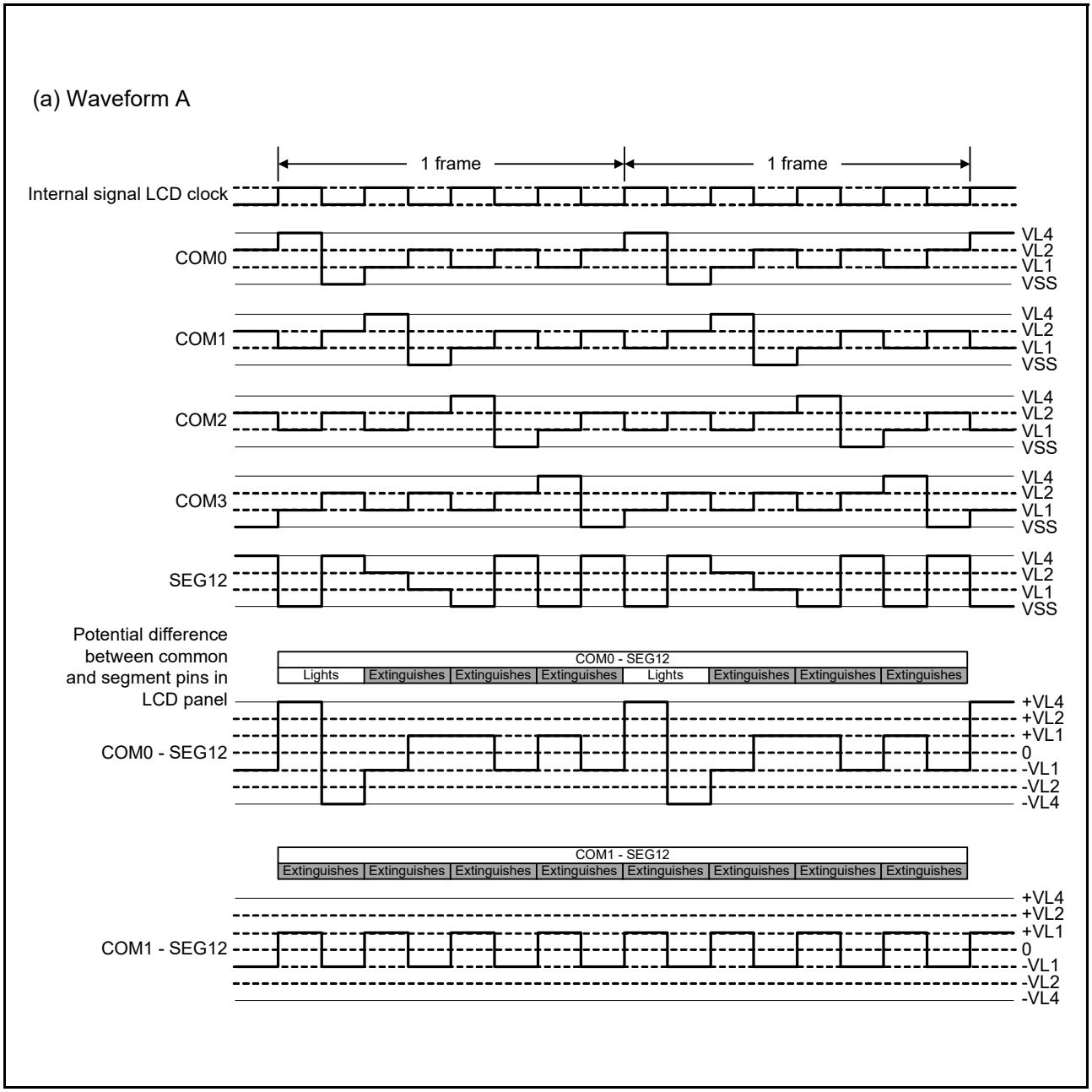


Figure 49.30 Four-time-slice LCD drive waveform examples between SEG12 and each common signals using 1/3 bias method (1/2)

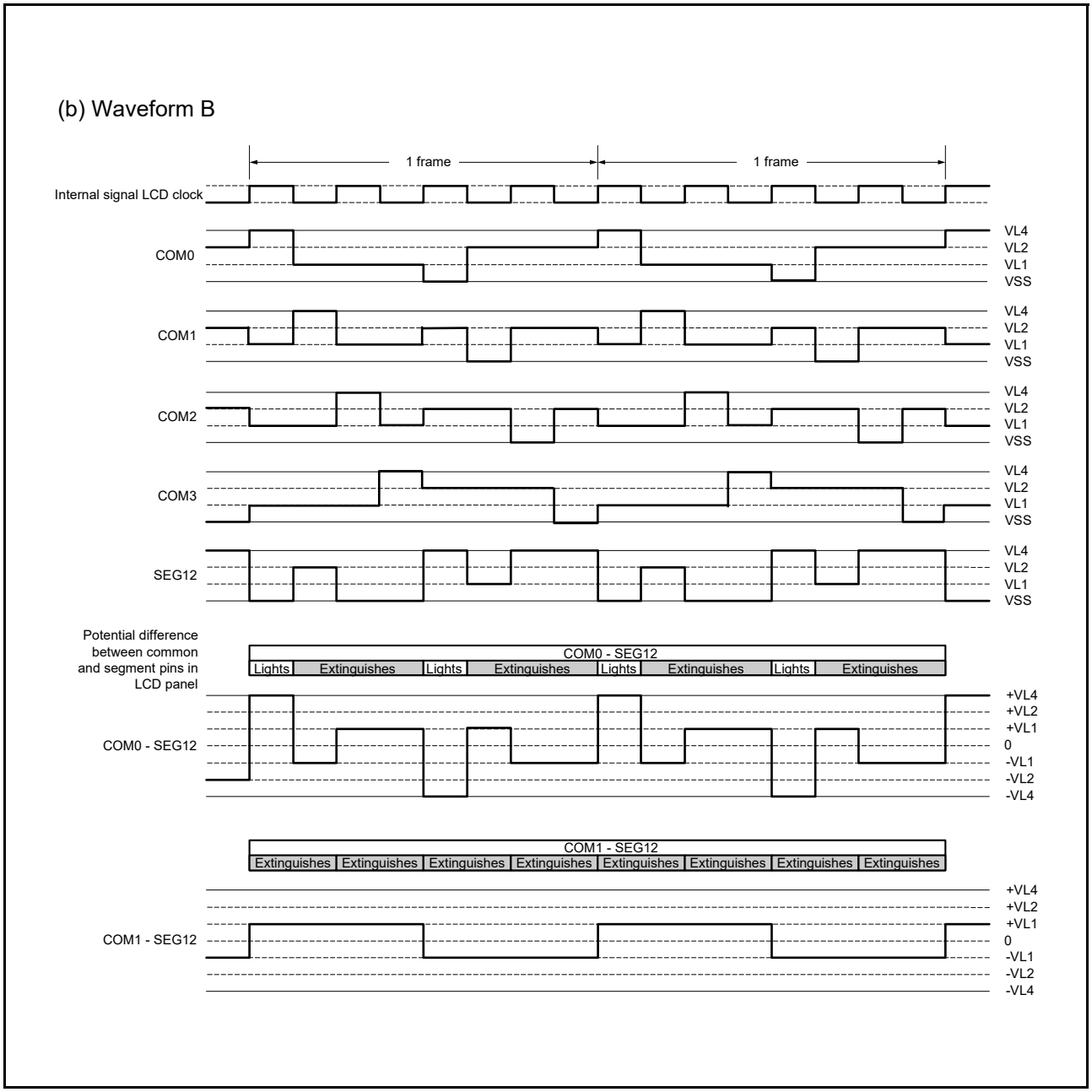


Figure 49.31 Four-time-slice LCD drive waveform examples between SEG12 and each common signals using 1/3 bias method (2/2)

49.9.5 Eight-Time-Slice Display Example

Figure 49.33 shows how a 15 × 8 dot LCD panel with the display pattern shown in Figure 49.32 is connected to the segment signals (SEG04 to SEG18) and the common signals (COM0 to COM7). This example displays "123" in the LCD panel. The contents of the display data register correspond to this display.

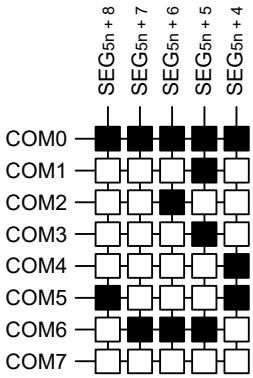
The following description focuses on numeral "3" ( ) displayed in the first digit. To display "3" in the LCD panel, the select or deselect voltage must be applied to the SEG04 to SEG08 pins at the select timing of the common signals COM0 to COM7. See Figure 49.32 for the relationship between the segment signals and LCD segments.

Table 49.19 Example of select (1) and deselect (0) data (COM0 to COM7)

Common	Segment				
	SEG04	**SEG05**	**SEG06**	**SEG07**	**SEG08**
COM0	Select	Select	Select	Select	Select
COM1	Deselect	Select	Deselect	Deselect	Deselect
COM2	Deselect	Deselect	Select	Deselect	Deselect
COM3	Deselect	Select	Deselect	Deselect	Deselect
COM4	Select	Deselect	Deselect	Deselect	Deselect
COM5	Select	Deselect	Deselect	Deselect	Select
COM6	Deselect	Select	Select	Select	Deselect
COM7	Deselect	Deselect	Deselect	Deselect	Deselect

According to Table 49.19, the display data register location that corresponds to SEG04 must contain "00110001b".

Figure 49.34 and Figure 49.35 show examples of LCD drive waveforms between the SEG04 signal and each common signal. When the select voltage is applied to SEG04 at the timing of COM0, a waveform is generated to turn on the associated LCD segment.



Note: The mounted segment output pins vary depending on the product.

Figure 49.32 Eight-time-slice LCD display pattern and electrode connections

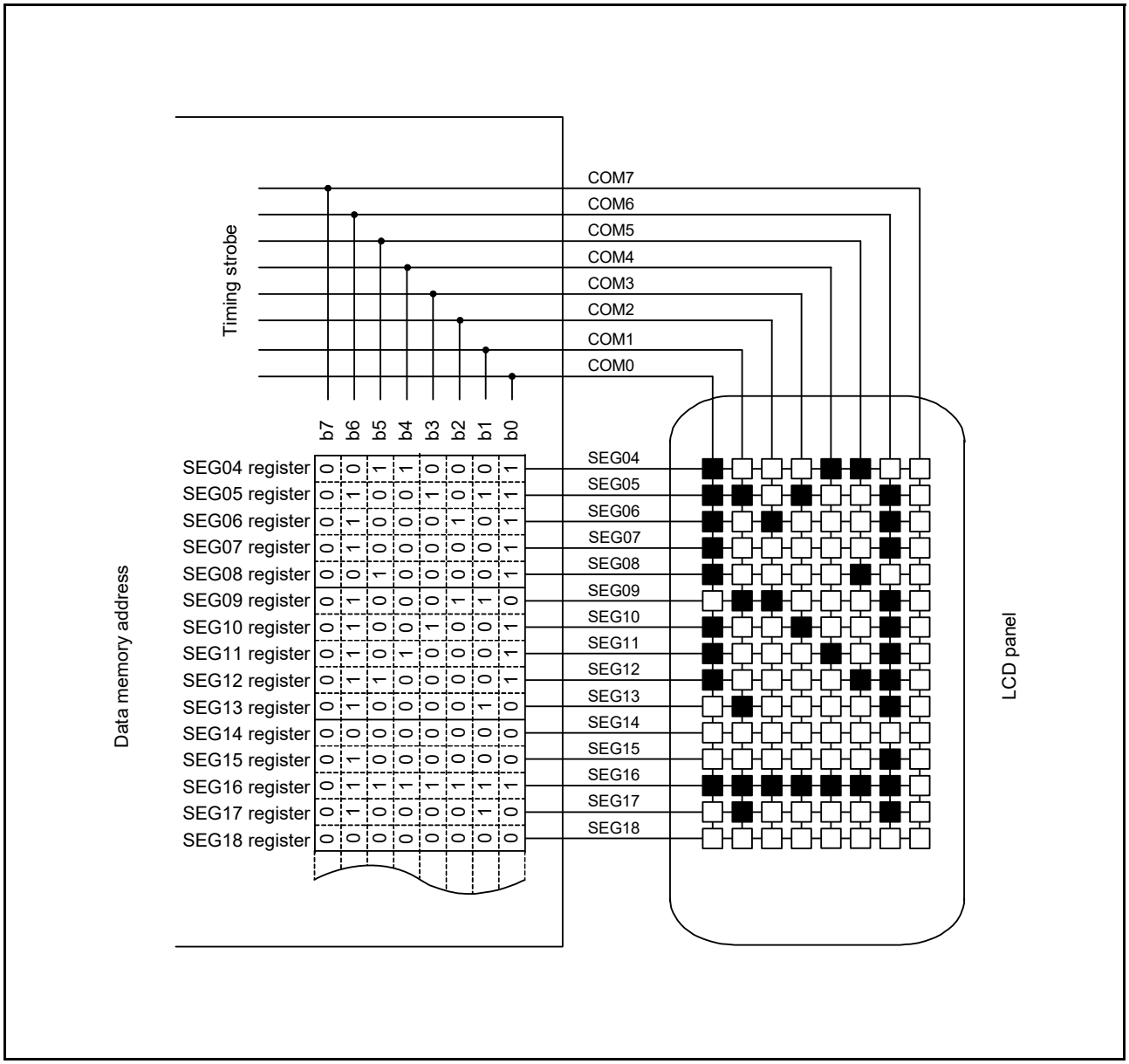


Figure 49.33 Example of connecting eight-time-slice LCD panel

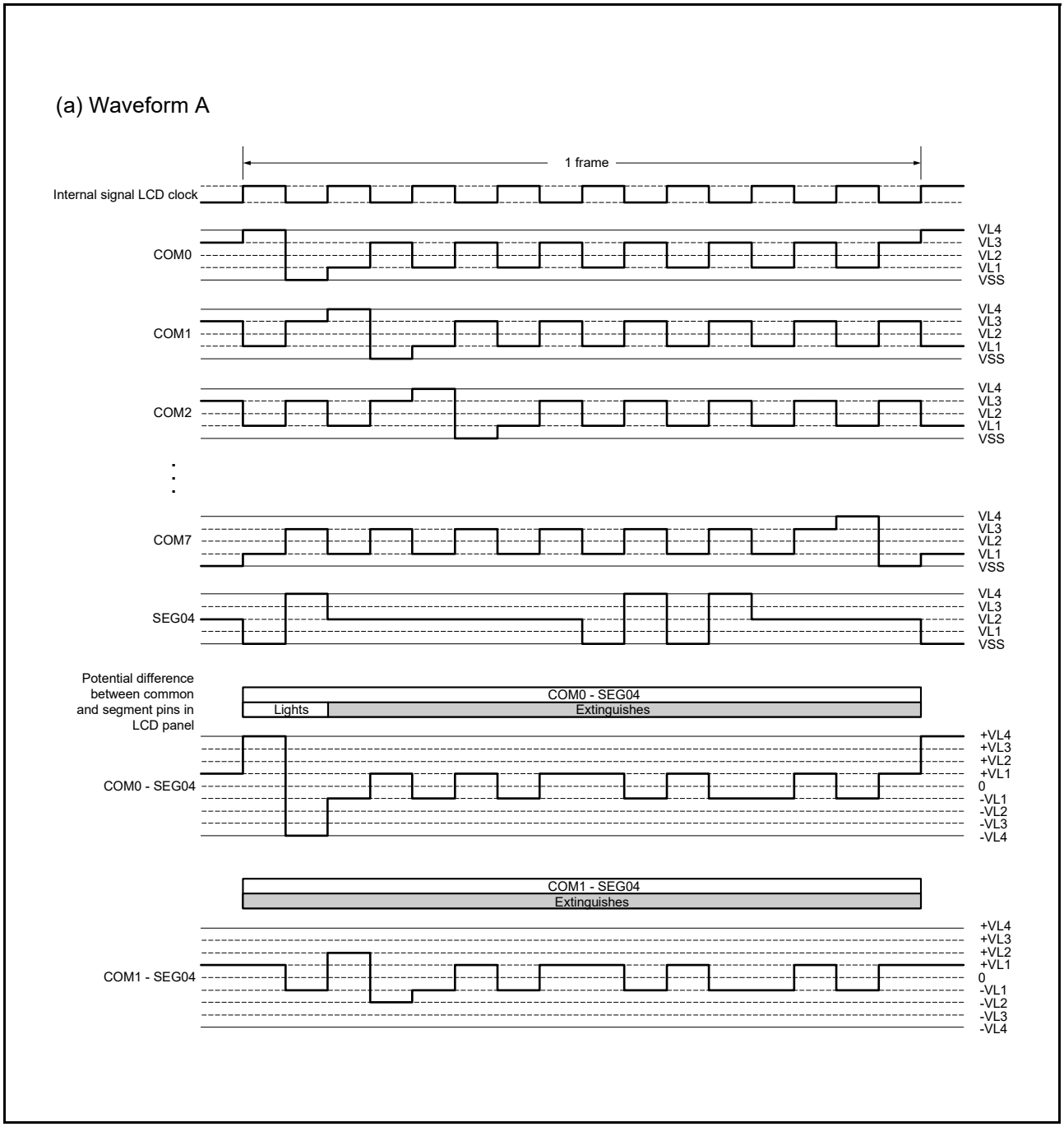


Figure 49.34 Eight-time-slice LCD drive waveform examples between SEG04 and each common signals using 1/4 bias method (1/2)

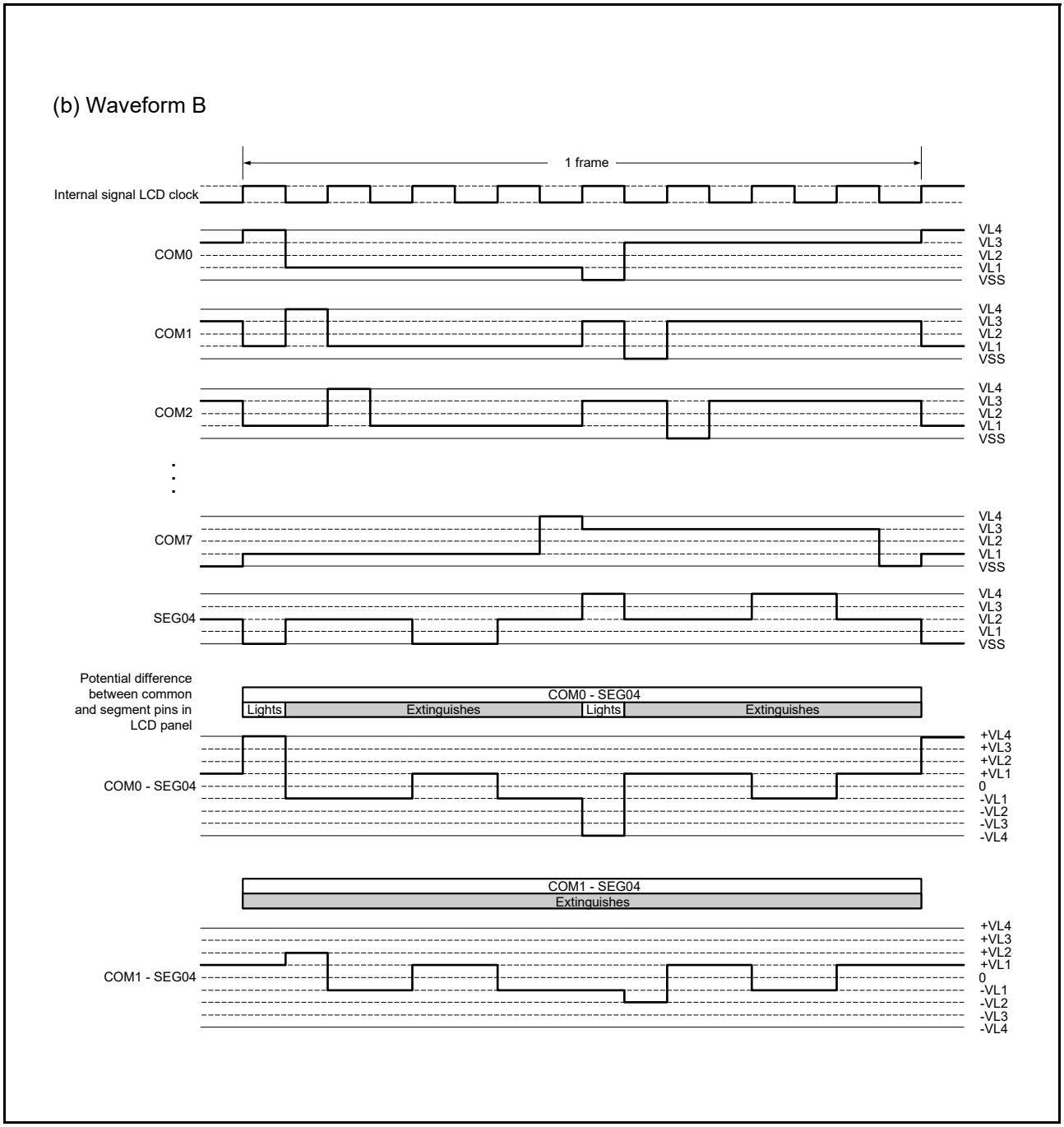


Figure 49.35 Eight-time-slice LCD drive waveform examples between SEG04 and each common signals using 1/4 bias method (2/2)

50. Secure Cryptographic Engine (SCE5)

50.1 Overview

The MCU incorporates a Secure Cryptographic Engine (SCE5) module to provide security functions. The module consists of an access management circuit, encryption engine, and random number generator. In combination with the Renesas Synergy Software Package (SSP) Crypto library, the SCE5 can prevent eavesdropping (confidentiality), falsification of information (integrity), and impersonation (authenticity).

The SCE5 module can only be used with the SSP Crypto library. For details, see the Crypto Framework and the SCE Crypto Driver sections in the *Renesas Synergy™ Software Package (SSP) User's Manual*.

Table 50.1 shows the SCE5 specifications and Figure 50.1 shows the SCE5 block diagram.

Table 50.1 SCE5 specifications

Item	Description
Access control	Access management circuit • In case of irregular access to the SCE5 due to a falsified program or runaway execution of a program, this circuit blocks all subsequent accesses and stops the output of data from the SCE5.
Encryption engine	Advanced Encryption Standard (AES): Compliant with NIST FIPS PUB 197 algorithm • Key sizes: 128 or 256 bits • Block size: 128 bits • Chaining modes - ECB, CBC, CTR: Compliant with NIST SP 800-38A - GCM: Compliant with NIST SP 800-38D - XTS: Compliant with NIST SP 800-38E. - GCTR • Throughput for 128-bit data - 44 PCLKA cycles for 128-bit key*1 - 61 PCLKA cycles for 256-bit key*1. AES-GCM • AES-GCM is realized by combining AES-GCTR and GHASH. Key management • Wrapped keys are only valid within the SCE5.
Generation of random numbers	32-bit true random number generator
Unique ID	• An ID unique to the MCU (unique ID) is accessible from the access management circuit through the dedicated bus • Combining the unique ID with the key generation information prevents illicit copying of data to another MCU.
Privileged mode	• The privileged mode access signal is connected to the access management circuit and is used to limit control of the SCE5 module to privileged mode only.
Low power consumption	Setting of the module-stop state is possible

Note 1. This does not include the overhead for calling functions of the SSP Crypto library.

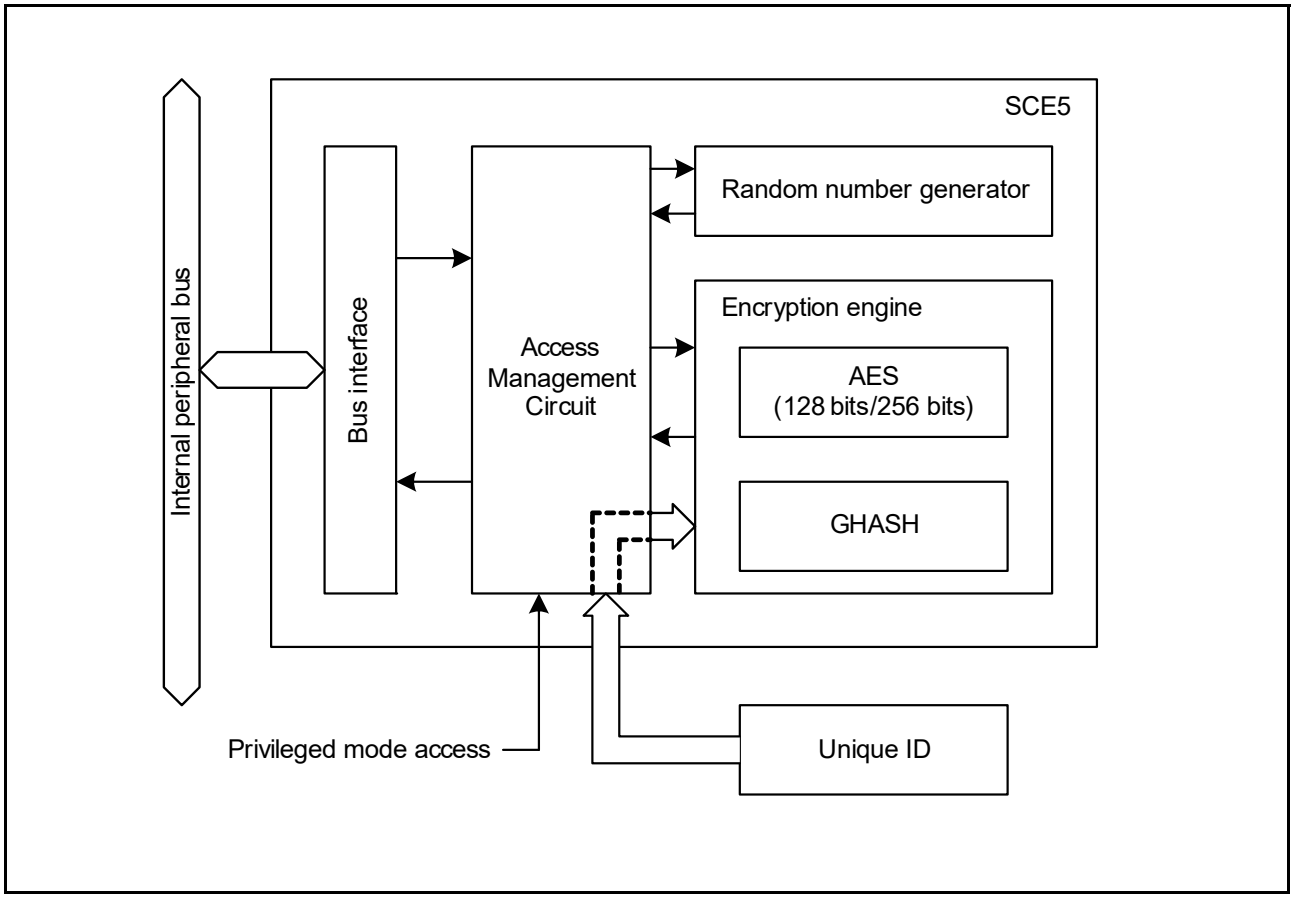


Figure 50.1 SCE5 block diagram

50.2 Operation

50.2.1 Encryption Engine

The encryption engine performs the following operation in hardware, as seen in Figure 50.2:

- Plaintext to ciphertext encryption
- Ciphertext to plaintext decryption.

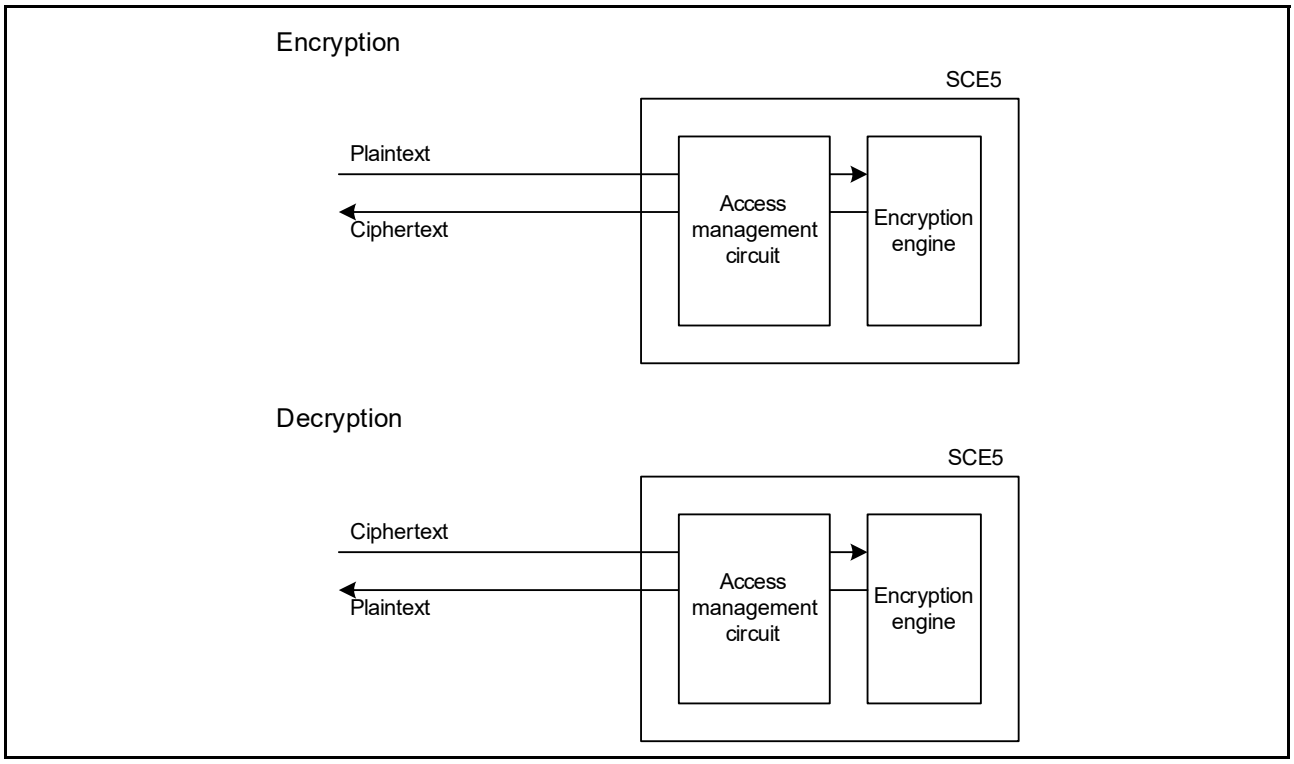


Figure 50.2 Encryption and decryption processes by encryption engine

50.2.2 Encryption and Decryption

To encrypt or decrypt data:

1. Input the data to encrypt or decrypt in the SCE5.
   The SCE5 converts the plaintext data to ciphertext or ciphertext data to plaintext.
2. Read the converted data.

The encryption engine has an input buffer and an output buffer, enabling encryption/decryption to proceed in parallel with data input/output. Figure 50.3 shows the encryption engine timing.

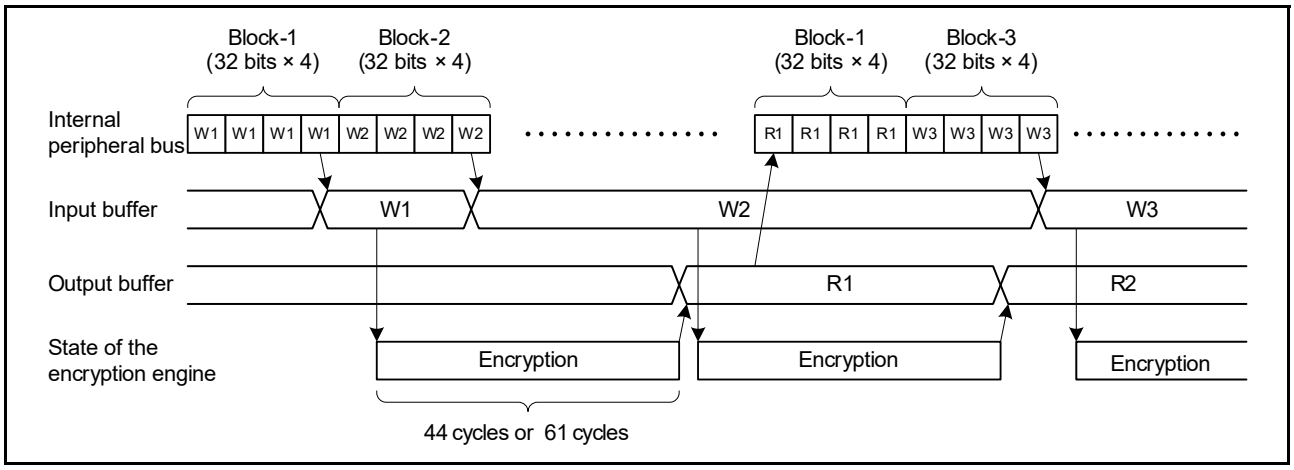


Figure 50.3 Encryption and decryption timing (AES)

50.3 Usage Notes

50.3.1 Software Standby Mode

If the MCU enters Software Standby mode while the encryption engine is processing, proper processing cannot be resumed after Software Standby mode is exited. Therefore, it is necessary to enter the Software Standby mode while the encryption engine is not running.

50.3.2 Settings for the Module-Stop Function

SCE5 operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The SCE5 module is initially stopped after reset. Releasing the module-stop state enables access to the registers.

51. Internal Voltage Regulator

51.1 Overview

The MCU includes a linear regulator (LDO) that supplies voltage to the internal circuits and memory, except for I/O and the analog domain.

51.2 Operation

Table 51.1 lists the LDO mode pin settings, and Figure 51.1 shows the LDO mode settings. The internal voltage is generated from VCC.

Table 51.1 LDO mode pin settings

Pin	Settings
All VCC pins	• 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 pin	• Connect each pin to VSS through a 4.7-µF multilayer ceramic capacitor. Place the capacitor close to the pin.

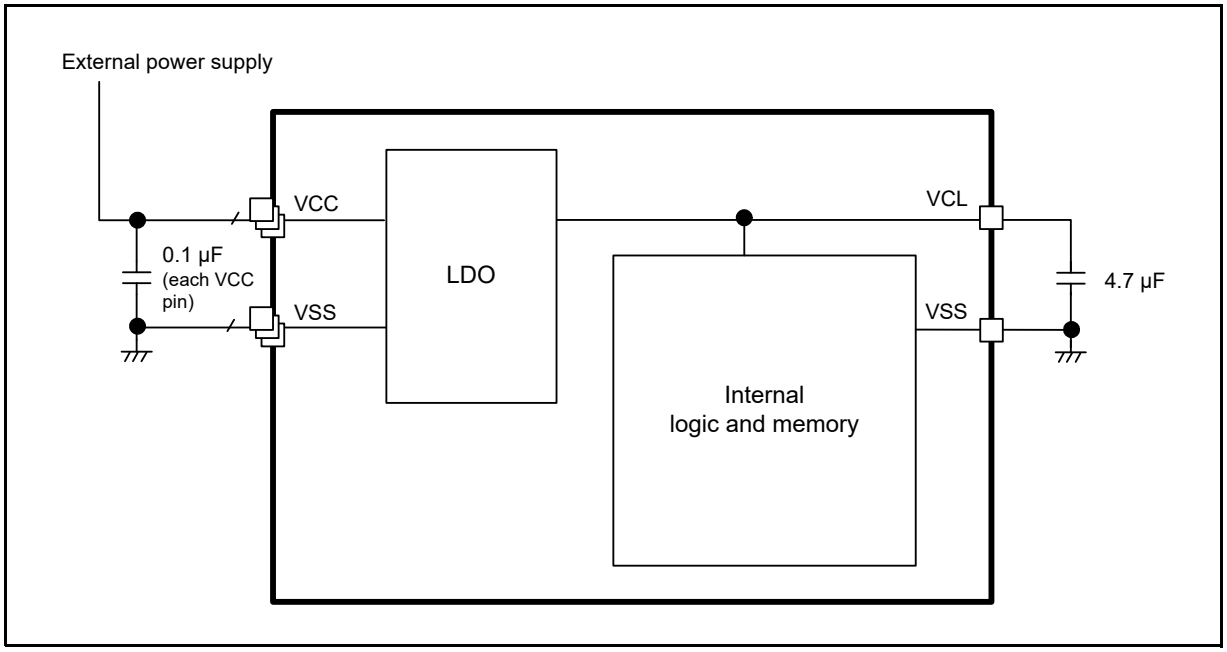


Figure 51.1 LDO mode settings

52. Electrical Characteristics

Unless otherwise specified, the electrical characteristics of the MCU are defined under the following conditions:

VCC*1 = AVCC0 = VCC_USB*2 = VCC_USB_LDO*2 = 1.6 to 5.5V, VREFH = VREFH0 = 1.6 to AVCC0, VBATT = 1.6 to 3.6V, VSS = AVSS0 = VREFL = VREFL0 = VSS_USB = 0V, Ta = T_{opr}

Note 1. The typical condition is set to VCC = 3.3V.
Note 2. When USBFS is not used.

Figure 52.1 shows the timing conditions.

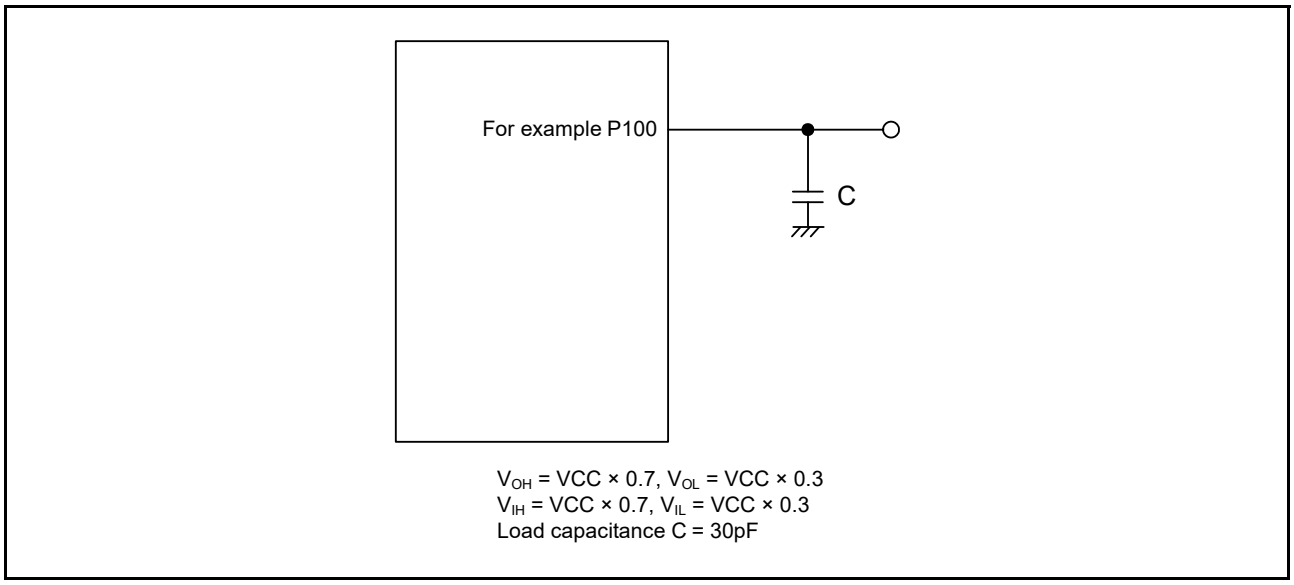


Figure 52.1 Input or output timing measurement conditions

The measurement conditions of timing specification in each peripherals are recommended for the best peripheral operation. However, make sure to adjust driving abilities of each pin to meet your conditions.

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 specification of each function is not guaranteed.

52.1 Absolute Maximum Ratings

Table 52.1 Absolute maximum ratings

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 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
		VREFH		V
VBATT power supply voltage		VBATT	–0.5 to +6.5	V
Analog power supply voltage		AVCC0	–0.5 to +6.5	V
USB power supply voltage		VCC_USB	–0.5 to +6.5	V
		VCC_USB_LDO	–0.5 to +6.5	V
Analog input voltage	When AN000 to AN015 are used	V_{AN}	–0.3 to AVCC0 + 0.3	V
	When AN016 to AN027 are used		–0.3 to VCC + 0.3	V
LCD voltage	VL1 voltage	V_{L1}	–0.3 to +2.8	V
	VL2 voltage	V_{L2}	–0.3 to +6.5	V
	VL3 voltage	V_{L3}	–0.3 to +6.5	V
	VL4 voltage	V_{L4}	–0.3 to +6.5	V
Operating temperature*2 *3 *4		T_{opr}	–40 to +85	°C
			–40 to +105	°C
Storage temperature		T_{stg}	–55 to +125	°C

Note 1. Ports P205, P206, P400 to P404, P407, P511, P512 are 5V-tolerant.
Note 2. See section 52.2.1, Tj/Ta Definition.
Note 3. Contact Renesas Electronics sales office for information on derating operation under Ta = +85°C to +105°C. Derating is the systematic reduction of load for improved reliability.
Note 4. The upper limit of operating temperature is 85°C or 105°C, depending on the product. For details, refer to 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 of high frequency characteristics between the VCC and VSS pins, between the AVCC0 and AVSS0 pins, between the VCC_USB and VSS_USB pins, between the VREFH0 and VREFL0 pins, and between the VREFH and VREFL pins. Place capacitors of about 0.1 µF as close as possible to every power supply pin and use the shortest and heaviest possible traces. Also, connect capacitors as stabilization capacitance.
Connect the VCL pin to a VSS pin by a 4.7 µF capacitor. The capacitor must be placed close to the pin.
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.

Table 52.2 Recommended operating conditions

Parameter	Symbol	Value	Min	Typ	Max	Unit
Power supply voltages	VCC*1, *2	When USBFS is not used	1.6	-	5.5	V
		When USBFS is used USB Regulator Disable	VCC_USB	-	3.6	V
		When USBFS is used USB Regulator Enable	VCC_USB _LDO	-	5.5	V
	VSS		-	0	-	V
USB power supply voltages	VCC_USB	When USBFS is not used	-	VCC	-	V
		When USBFS is used USB Regulator Disable (Input)	3.0	3.3	3.6	V
	VCC_USB_LDO	When USBFS is not used	-	VCC	-	V
		When USBFS is used USB Regulator Disable	-	VCC	-	V
		When USBFS is used USB Regulator Enable	3.8	-	5.5	V
	VSS_USB		-	0	-	V
VBATT power supply voltage	VBATT	When the battery backup function is not used	-	VCC	-	V
		When the battery backup function is used	1.6	-	3.6	V
Analog power supply voltages	AVCC0*1, *2		1.6	-	5.5	V
	AVSS0		-	0	-	V
	VREFH0	When used as ADC14 Reference	1.6	-	AVCC0	V
	VREFL0		-	0	-	V
	VREFH	When used as DAC12 Reference	1.6	-	AVCC0	V
	VREFL		-	0	-	V

Note 1. Use AVCC0 and VCC under the following conditions:
AVCC0 and VCC can be set individually within the operating range when VCC ≥ 2.2 V and AVCC0 ≥ 2.2 V
AVCC0 = VCC when VCC < 2.2 V or AVCC0 < 2.2 V

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 pin.

52.2 DC Characteristics

52.2.1 Tj/Ta Definition

Table 52.3 DC Characteristics
Conditions: Products with operating temperature (T_a) –40 to +105°C

Parameter	Symbol	Typ	Max	Unit	Test conditions
Permissible junction temperature	Tj	-	125	°C	High-speed mode Middle-speed mode Low-voltage mode Low-speed mode Subosc-speed mode
			105*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 85°C or 105°C, depending on the product. For details, see section 1.3, Part Numbering. If the part number shows the operation temperature as 85°C, then Tj max is 105°C, otherwise it is 125°C.

52.2.2 I/O V_{IH}, V_{IL}

Table 52.4 I/O V_{IH}, V_{IL} (1)
Conditions: VCC = 2.7 to 5.5 V, AVCC0 = 2.7 to 5.5 V, VBATT = 1.6 to 3.6 V, VSS = AVSS0 = 0 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Schmitt trigger input voltage	IIC*1 (except for SMBus)	V_{IH}	VCC × 0.7	-	5.8	V	-
		V_{IL}	-	-	VCC × 0.3		
		ΔV_T	VCC × 0.05	-	-		
	RES, NMI Other peripheral input pins excluding IIC	V_{IH}	VCC × 0.8	-	-		
		V_{IL}	-	-	VCC × 0.2		
		ΔV_T	VCC × 0.1	-	-		
Input voltage (except for Schmitt trigger input pin)	IIC (SMBus)*2	V_{IH}	2.2	-	-		VCC = 3.6 to 5.5 V
		V_{IH}	2.0	-	-		VCC = 2.7 to 3.6 V
		V_{IL}	-	-	0.8		-
	5V-tolerant ports*3	V_{IH}	VCC × 0.8	-	5.8		
		V_{IL}	-	-	VCC × 0.2		
	P000 to P015	V_{IH}	AVCC0 × 0.8	-	-		
		V_{IL}	-	-	AVCC0 × 0.2		
	EXTAL D00 to D15 Input ports pins except for P000 to P015	V_{IH}	VCC × 0.8	-	-		
		V_{IL}	-	-	VCC × 0.2		
When V_{BATT} power supply is selected	P402, P403, P404	V_{IH}	V_{BATT} × 0.8	-	V_{BATT} + 0.3		
		V_{IL}	-	-	V_{BATT} × 0.2		
		ΔV_T	V_{BATT} × 0.05	-	-		

Note 1. SCL0_A, SDA0_A, SCL1_A, SDA1_A, SCL2, SDA2, SDA0_B (total 7 pins).
Note 2. SCL0_A, SDA0_A, SCL0_B, SDA0_B, SCL1_A, SDA1_A, SCL1_B, SDA1_B, SCL2, SDA2 (total 10 pins).
Note 3. P205, P206, P400 to P404, P407, P511, P512 (total 10 pins).

Table 52.5 I/O V_{IH}, V_{IL} (2)

Conditions: VCC = 1.6 to 2.7 V, AVCC0 = 1.6 to 2.7 V, VBATT = 1.6 to 3.6 V, VSS = AVSS0 = 0 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Schmitt trigger input voltage	RES, NMI Peripheral input pins	V_{IH}	VCC × 0.8	-	-	V	-
		V_{IL}	-	-	VCC × 0.2		
		ΔV_T	VCC × 0.01	-	-		
Input voltage (except for Schmitt trigger input pin)	5V-tolerant ports*1	V_{IH}	VCC × 0.8	-	5.8		
		V_{IL}	-	-	VCC × 0.2		
	P000 to P015	V_{IH}	AVCC0 × 0.8	-	-		
		V_{IL}	-	-	AVCC0 × 0.2		
	EXTAL D0 to D15 Input ports pins except for P000 to P015	V_{IH}	VCC × 0.8	-	-		
		V_{IL}	-	-	VCC × 0.2		
When V_{BATT} power supply is selected	P402, P403, P404	V_{IH}	V_{BATT} × 0.8	-	V_{BATT} + 0.3		
		V_{IL}	-	-	V_{BATT} × 0.2		
		ΔV_T	V_{BATT} × 0.01	-	-		

Note 1. P205, P206, P400 to P404, P407, P511, P512 (total 10 pins)

52.2.3 I/O I_{OH}, I_{OL}

Table 52.6 I/O I_{OH}, I_{OL}
Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max	Unit
Permissible output current (average value per pin)	Ports P000 to P015, Ports P212, P213	-	I_{OH}	-	-	–4.0	mA
			I_{OL}	-	-	4.0	mA
	Ports P408, P409	Low drive*1	I_{OH}	-	-	–4.0	mA
			I_{OL}	-	-	4.0	mA
		Middle drive*2 VCC = 2.7 to 3.0 V	I_{OH}	-	-	–8.0	mA
			I_{OL}	-	-	8.0	mA
		Middle drive*2 VCC = 3.0 to 5.5 V	I_{OH}	-	-	–20.0	mA
			I_{OL}	-	-	20.0	mA
	Ports P100 to P115, P201 to P204, P300 to P315, P500 to P503, P600 to P606, P608 to P614, P800 to P809, P900 to P902 (total 67 pins)	Low drive*1	I_{OH}	-	-	–4.0	mA
			I_{OL}	-	-	4.0	mA
		Middle drive*2	I_{OH}	-	-	–4.0	mA
			I_{OL}	-	-	8.0	mA
	Other output pin*3	Low drive*1	I_{OH}	-	-	–4.0	mA
			I_{OL}	-	-	4.0	mA
		Middle drive*2	I_{OH}	-	-	–8.0	mA
			I_{OL}	-	-	8.0	mA
Permissible output current (Max value per pin)	Ports P000 to P015, Ports P212, P213	-	I_{OH}	-	-	–4.0	mA
			I_{OL}	-	-	4.0	mA
	Ports P408, P409	Low drive*1	I_{OH}	-	-	–4.0	mA
			I_{OL}	-	-	4.0	mA
		Middle drive*2 VCC = 2.7 to 3.0 V	I_{OH}	-	-	–8.0	mA
			I_{OL}	-	-	8.0	mA
		Middle drive*2 VCC = 3.0 to 5.5 V	I_{OH}	-	-	–20.0	mA
			I_{OL}	-	-	20.0	mA
	Ports P100 to P115, P201 to P204, P300 to P315, P500 to P503, P600 to P606, P608 to P614, P800 to P809, P900 to P902 (total 67 pins)	Low drive*1	I_{OH}	-	-	–4.0	mA
			I_{OL}	-	-	4.0	mA
		Middle drive*2	I_{OH}	-	-	–4.0	mA
			I_{OL}	-	-	8.0	mA
	Other output pin*3	Low drive*1	I_{OH}	-	-	–4.0	mA
			I_{OL}	-	-	4.0	mA
		Middle drive*2	I_{OH}	-	-	–8.0	mA
			I_{OL}	-	-	8.0	mA
Permissible output current (max value total pins)	Total of ports P000 to P015		$\Sigma I_{OH\ (max)}$	-	-	–30	mA
			$\Sigma I_{OL\ (max)}$	-	-	30	mA
	Total of all output pin		$\Sigma I_{OH\ (max)}$	-	-	–60	mA
			$\Sigma I_{OL\ (max)}$	-	-	60	mA

Caution: To protect the reliability of the MCU, the output current values should not exceed the values in this table. The average output current indicates the average value of current measured during 100 µs.

Note 1. This is the value when low driving ability is selected with the Port Drive Capability bit in PmnPFS register.
Note 2. This is the value when middle driving ability is selected with the Port Drive Capability bit in PmnPFS register.
Note 3. Except for ports P200, P214, P215, which are input ports.

52.2.4 I/O V_{OH}, V_{OL}, and Other Characteristics

Table 52.7 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	IIC*1, *2		V_{OL}	-	-	0.4	V	I_{OL} = 3.0 mA
			V_{OL}	-	-	0.6		I_{OL} = 6.0 mA
	Ports P408, P409*2, *3		V_{OH}	VCC – 1.0	-	-		I_{OH} = –20 mA
			V_{OL}	-	-	1.0		I_{OL} = 20 mA
	Ports P000 to P015	Low drive	V_{OH}	AVCC0 – 0.8	-	-		I_{OH} = –2.0 mA
			V_{OL}	-	-	0.8		I_{OL} = 2.0 mA
		Middle drive	V_{OH}	AVCC0 – 0.8	-	-		I_{OH} = –4.0 mA
			V_{OL}	-	-	0.8		I_{OL} = 4.0 mA
	Other output pins*4	Low drive	V_{OH}	VCC – 0.8	-	-		I_{OH} = –2.0 mA
			V_{OL}	-	-	0.8		I_{OL} = 2.0 mA
		Middle drive*5	V_{OH}	VCC – 0.8	-	-		I_{OH} = –4.0 mA
			V_{OL}	-	-	0.8		I_{OL} = 4.0 mA

Note 1. SCL0_A, SDA0_A, SCL0_B, SDA0_B, SCL1_A, SDA1_A, SCL1_B, SDA1_B, SCL2, SDA2 (total 10 pins).
Note 2. This is the value when middle driving ability is selected with the Port Drive Capability bit in PmnPFS register.
Note 3. Based on characterization data, not tested in production.
Note 4. Except for ports P200, P214, P215, which are input ports.
Note 5. Except for P212, P213.

Table 52.8 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	IIC*1, *2		V_{OL}	-	-	0.4	V	I_{OL} = 3.0 mA
			V_{OL}	-	-	0.6		I_{OL} = 6.0 mA
	Ports P408, P409*2, *3		V_{OH}	VCC – 1.0	-	-		I_{OH} = –20 mA VCC = 3.3 V
			V_{OL}	-	-	1.0		I_{OL} = 20 mA VCC = 3.3 V
	Ports P000 to P015	Low drive	V_{OH}	AVCC0 – 0.5	-	-		I_{OH} = –1.0 mA
			V_{OL}	-	-	0.5		I_{OL} = 1.0 mA
		Middle drive	V_{OH}	AVCC0 – 0.5	-	-		I_{OH} = –2.0 mA
			V_{OL}	-	-	0.5		I_{OL} = 2.0 mA
	Other output pins*4	Low drive	V_{OH}	VCC – 0.5	-	-		I_{OH} = –1.0 mA
			V_{OL}	-	-	0.5		I_{OL} = 1.0 mA
		Middle drive*5	V_{OH}	VCC – 0.5	-	-		I_{OH} = –2.0 mA
			V_{OL}	-	-	0.5		I_{OL} = 2.0 mA

Note 1. SCL0_A, SDA0_A, SCL0_B, SDA0_B, SCL1_A, SDA1_A, SCL1_B, SDA1_B, SCL2, SDA2 (total 10 pins).
Note 2. This is the value when middle driving ability is selected with the Port Drive Capability bit in PmnPFS register.
Note 3. Based on characterization data, not tested in production.
Note 4. Except for ports P200, P214, P215, which are input ports.
Note 5. Except for P212, P213.

Table 52.9 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 P015	Low drive	V_{OH}	AVCC0 – 0.3	-	-	V	I_{OH} = –0.5 mA
			V_{OL}	-	-	0.3		I_{OL} = 0.5 mA
		Middle drive	V_{OH}	AVCC0 – 0.3	-	-		I_{OH} = –1.0 mA
			V_{OL}	-	-	0.3		I_{OL} = 1.0 mA
	Other output pins*1	Low drive	V_{OH}	VCC – 0.3	-	-		I_{OH} = –0.5 mA
			V_{OL}	-	-	0.3		I_{OL} = 0.5 mA
		Middle drive*2	V_{OH}	VCC – 0.3	-	-		I_{OH} = –1.0 mA
			V_{OL}	-	-	0.3		I_{OL} = 1.0 mA

Note 1. Except for ports P200, P214, P215, which are input ports.
Note 2. Except for P212, P213.

Table 52.10 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, P200, P214, P215	$\lvert I_{in} \rvert$	-	-	1.0	µA	V_{in} = 0 V V_{in} = VCC
Three-state leakage current (off state)	5V-tolerant ports	$\lvert I_{TSI} \rvert$	-	-	1.0	µA	V_{in} = 0 V V_{in} = 5.8 V
	Other ports (except for ports P200, P214, P215 and 5 V tolerant)		-	-	1.0		V_{in} = 0 V V_{in} = VCC
Input pull-up resistor	All Ports (except for ports P200, P214, P215)	R_U	10	20	50	kΩ	V_{in} = 0 V
Input capacitance	USB_DP, USB_DM, P100 to P103, P111, P112, P200	C_{in}	-	-	30	pF	V_{in} = 0 V f = 1 MHz T_a = 25°C
	Other input pins		-	-	15		

52.2.5 I/O Pin Output Characteristics of Low Drive Capacity

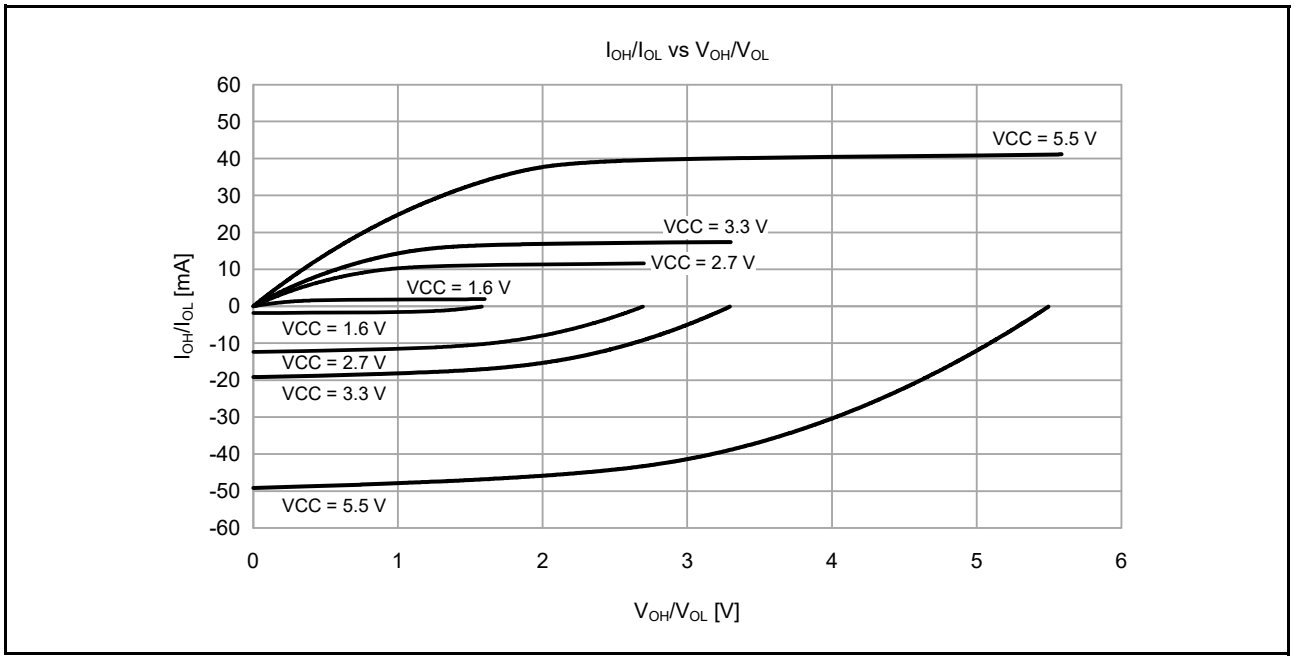


Figure 52.2 **V_{OH}/V_{OL} and I_{OH}/I_{OL} voltage characteristics at Ta = 25°C when low drive output is selected (reference data)**

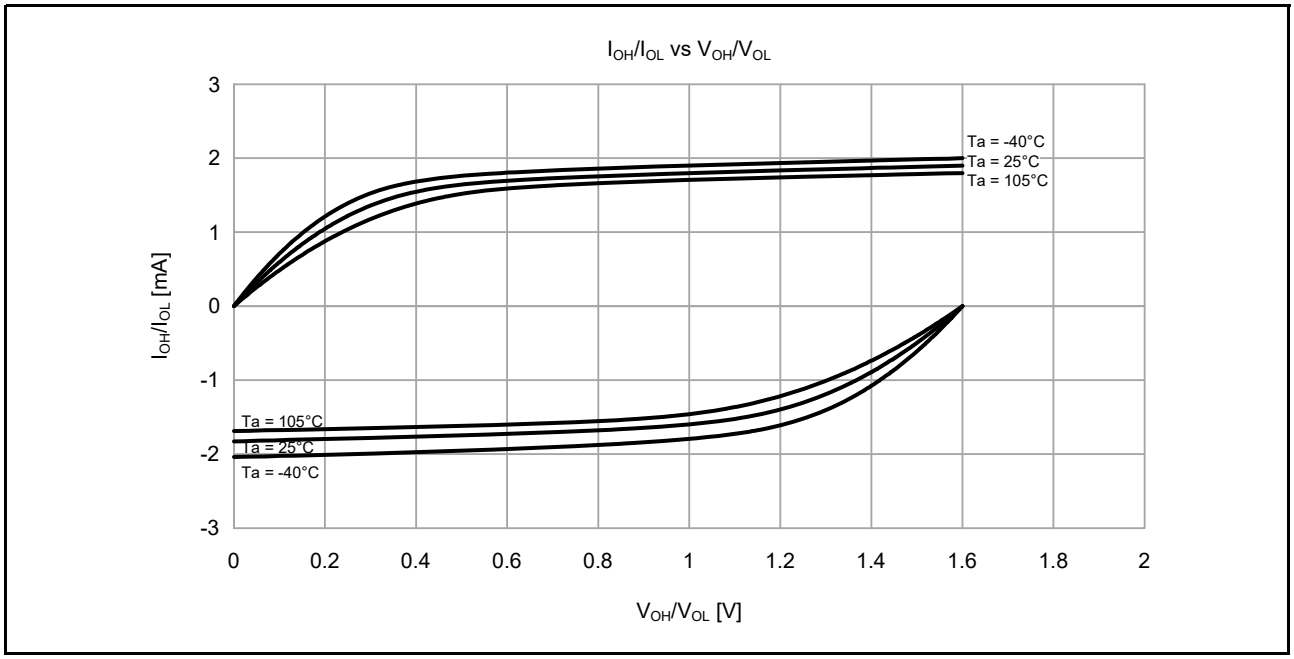


Figure 52.3 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 1.6 V when low drive output is selected (reference data)**

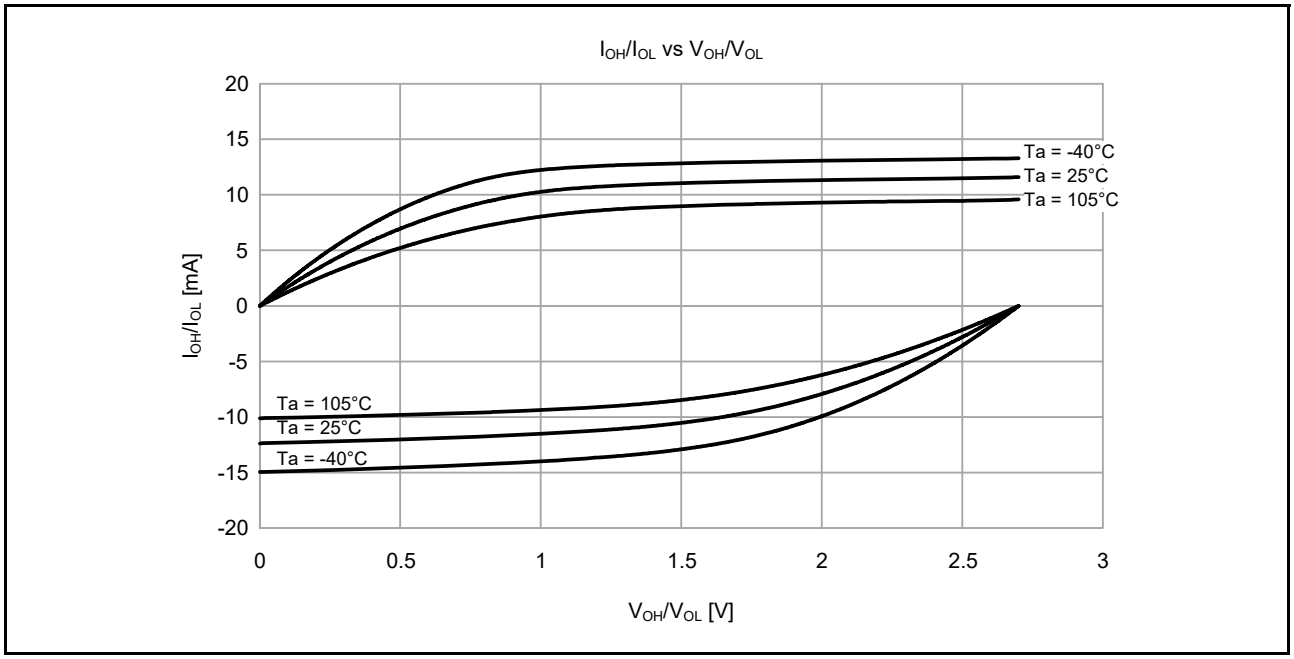


Figure 52.4 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 2.7 V when low drive output is selected (reference data)**

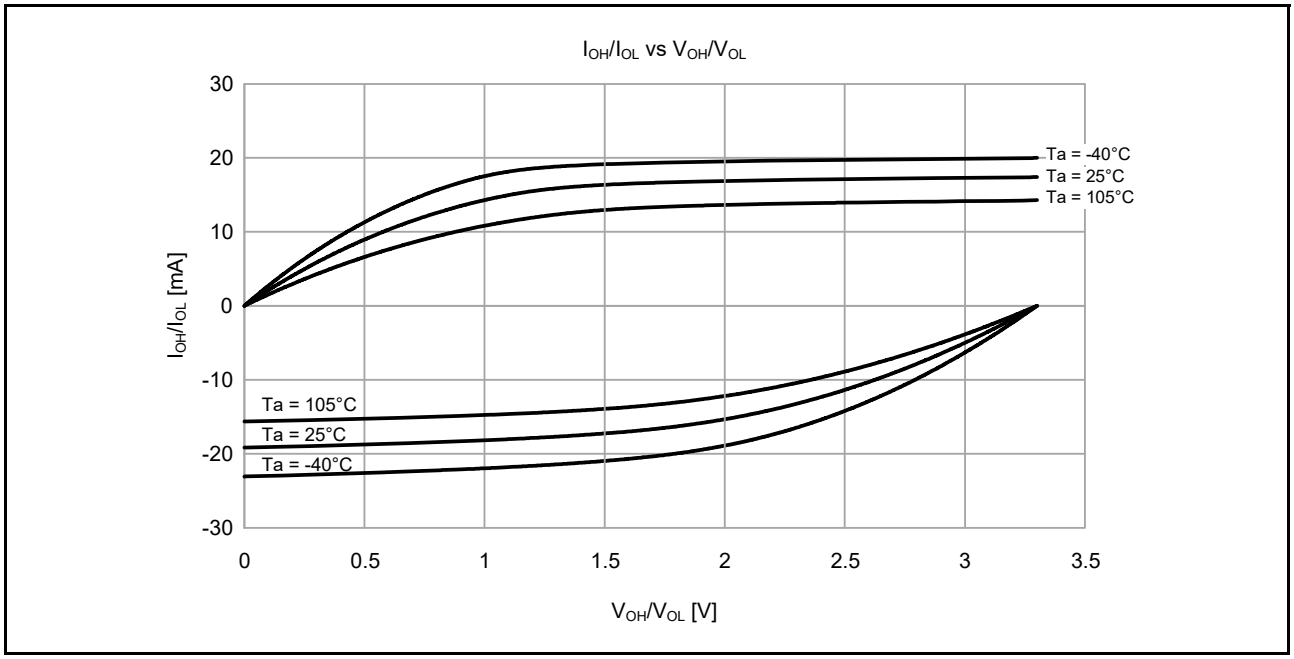


Figure 52.5 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 3.3 V when low drive output is selected (reference data)**

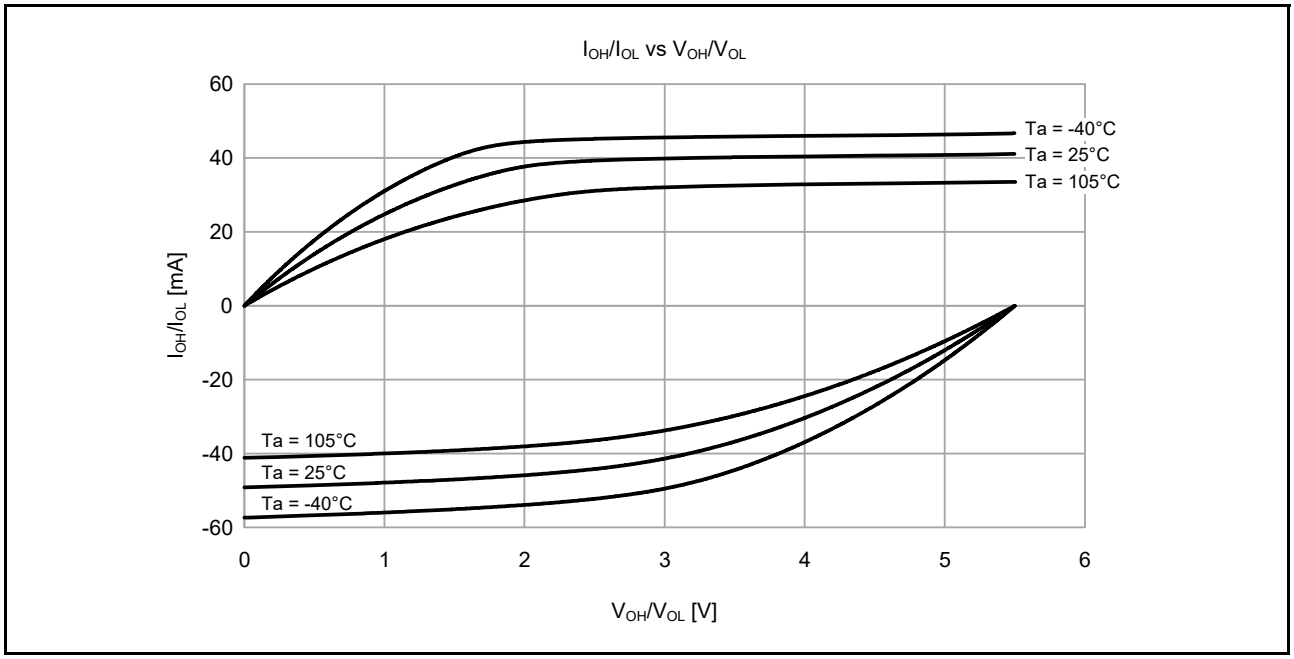


Figure 52.6 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 5.5 V when low drive output is selected (reference data)**

52.2.6 I/O Pin Output Characteristics of Middle Drive Capacity

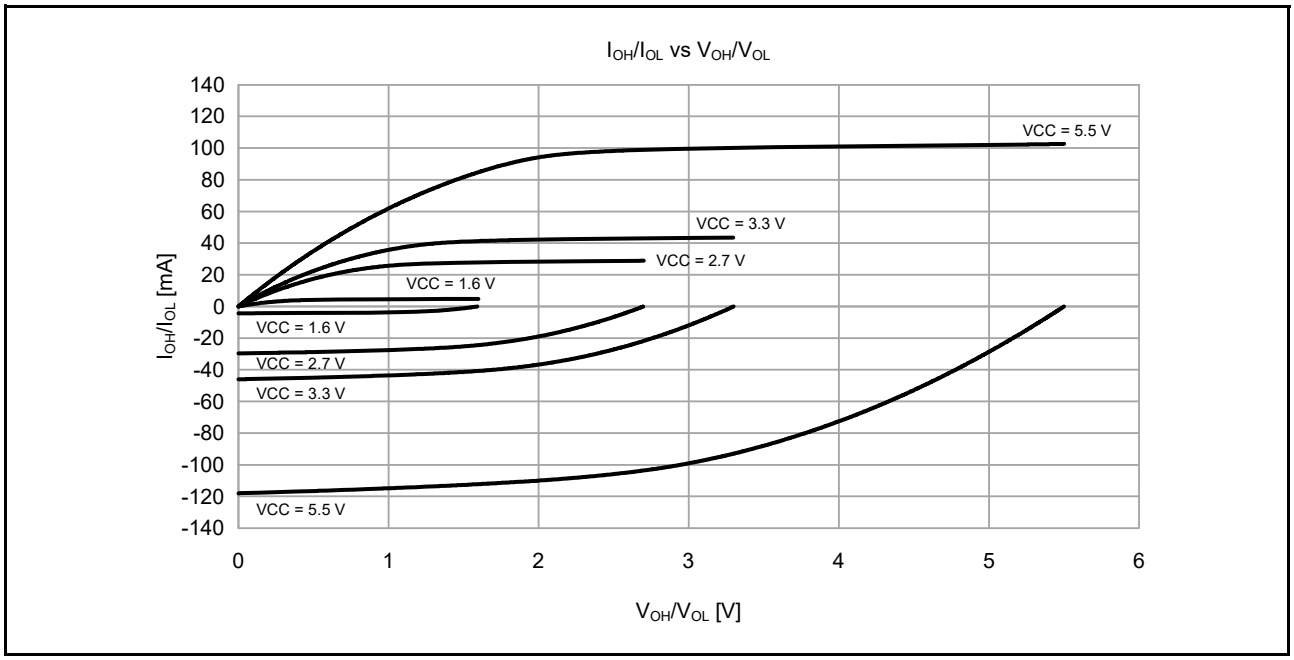


Figure 52.7 **V_{OH}/V_{OL} and I_{OH}/I_{OL} voltage characteristics at Ta = 25°C when middle drive output is selected (reference data)**

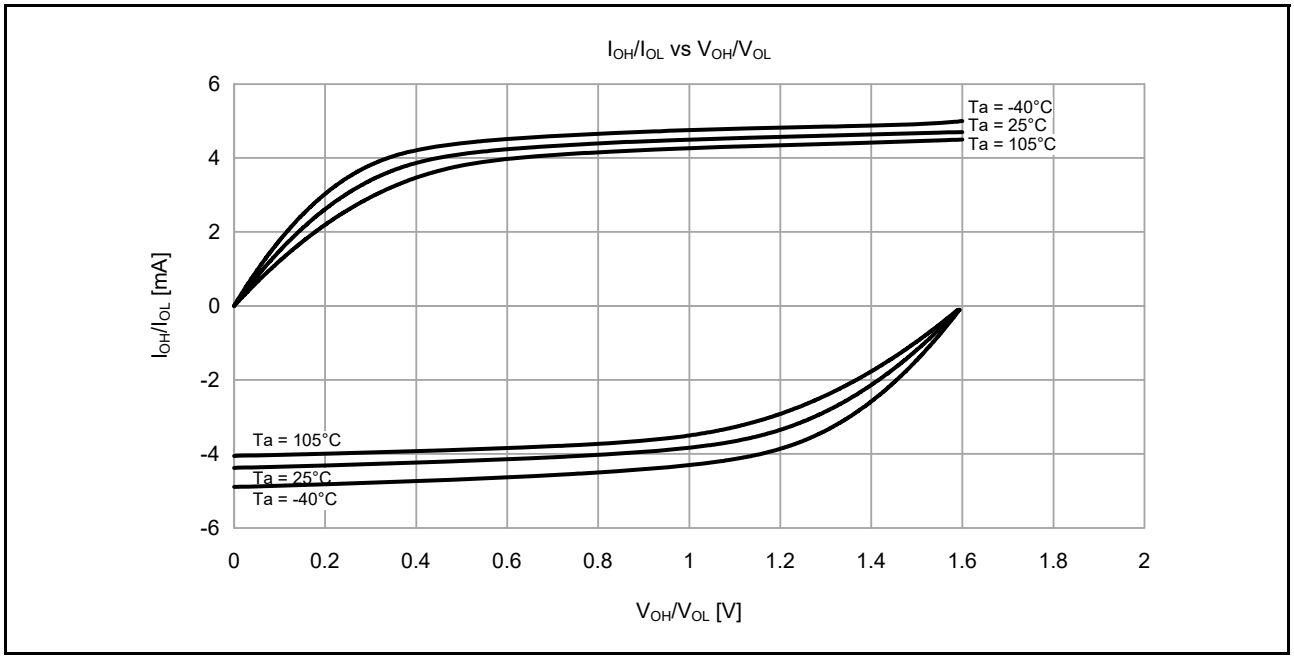


Figure 52.8 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 1.6 V when middle drive output is selected (reference data)**

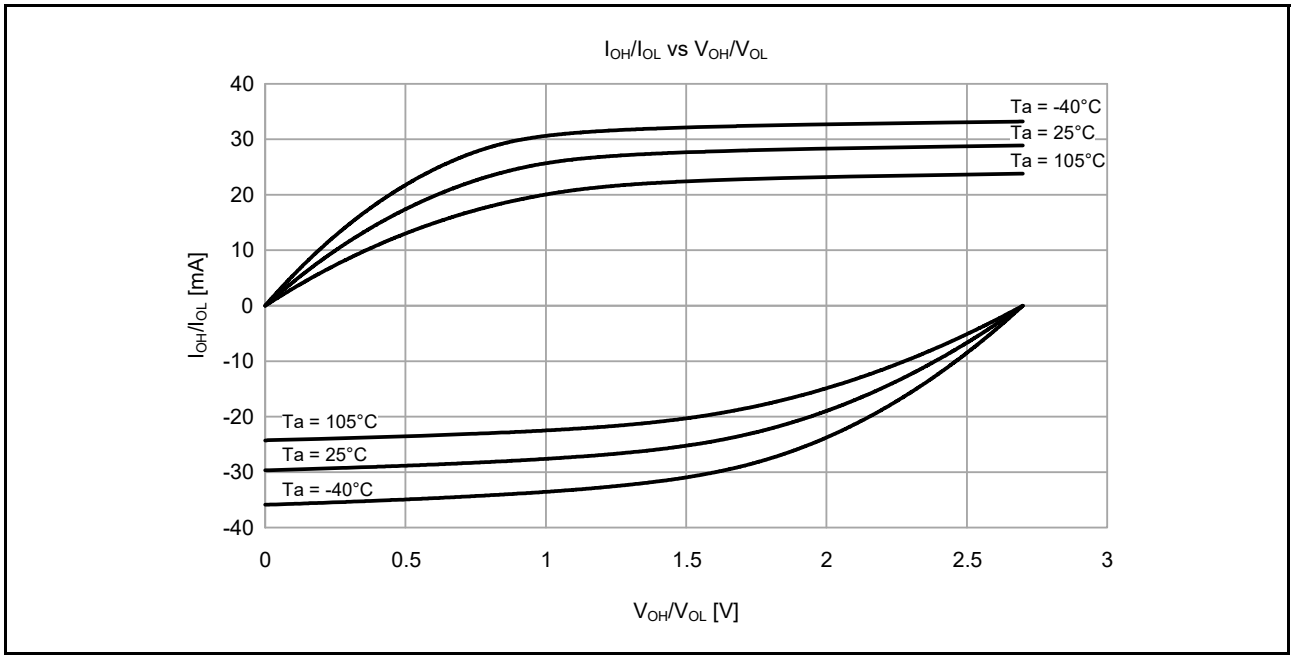


Figure 52.9 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 2.7 V when middle drive output is selected (reference data)**

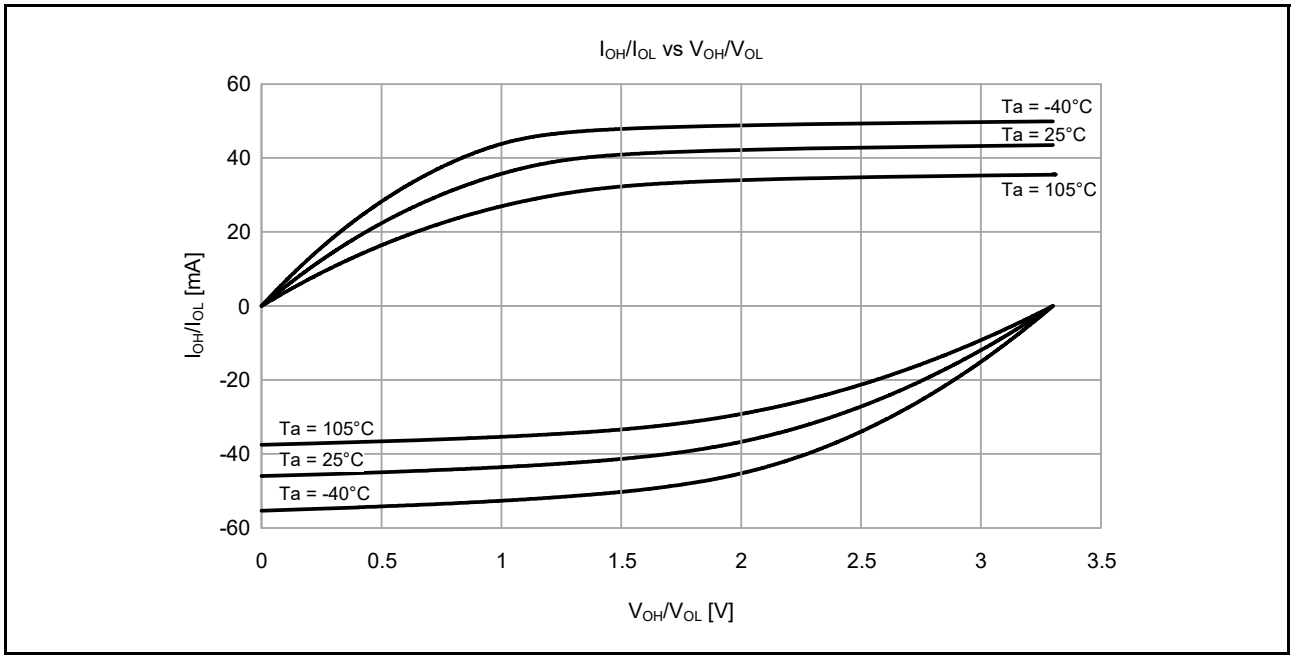


Figure 52.10 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 3.3 V when middle drive output is selected (reference data)**

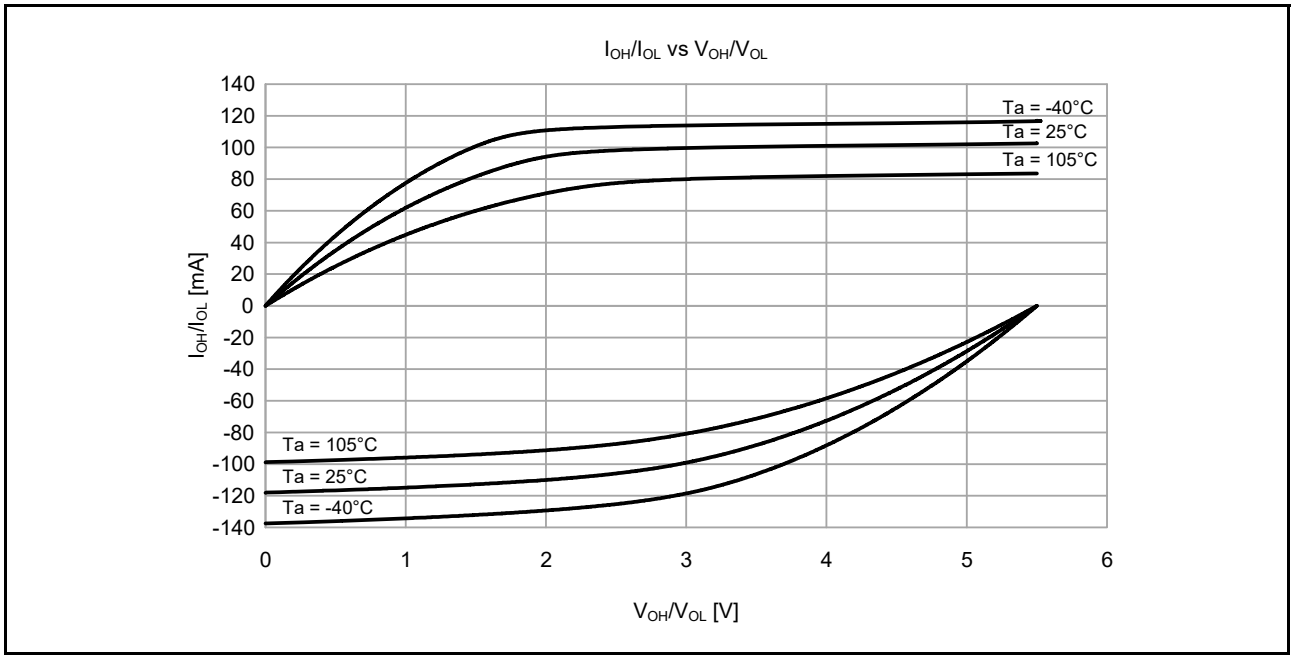


Figure 52.11 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 5.5 V when middle drive output is selected (reference data)**

52.2.7 P408, P409 I/O Pin Output Characteristics of Middle Drive Capacity

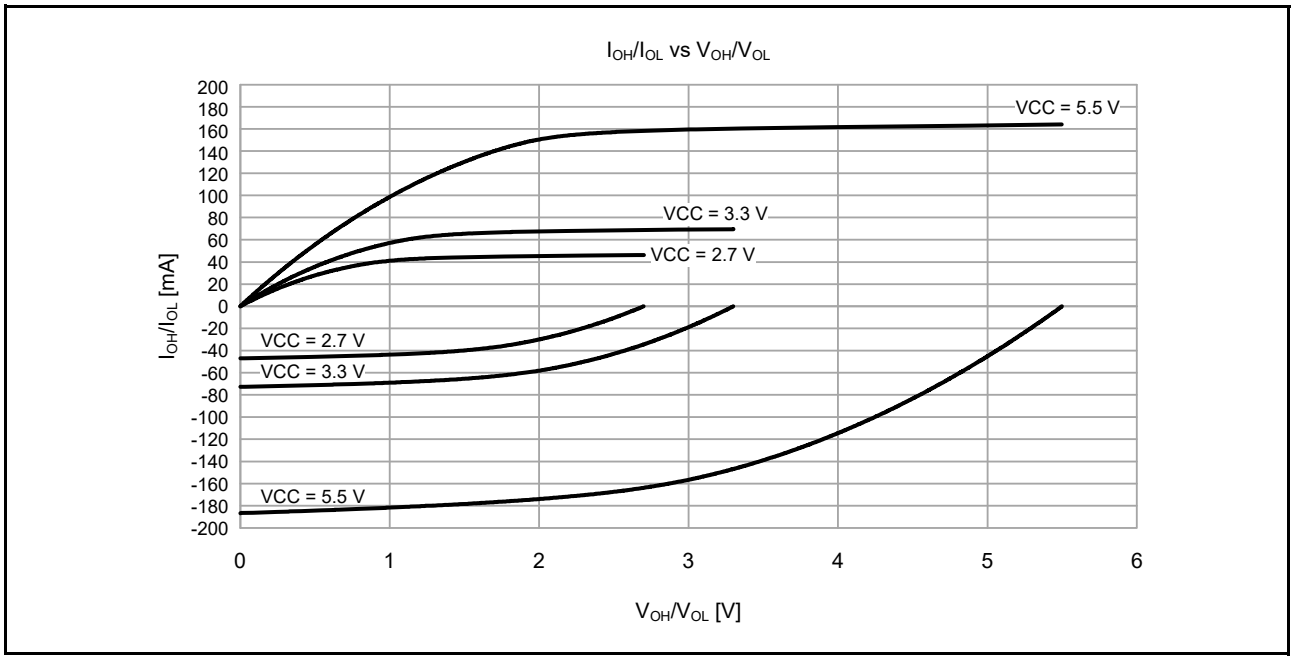


Figure 52.12 V_{OH}/V_{OL} and I_{OH}/I_{OL} voltage characteristics at Ta = 25°C when middle drive output is selected (reference data)

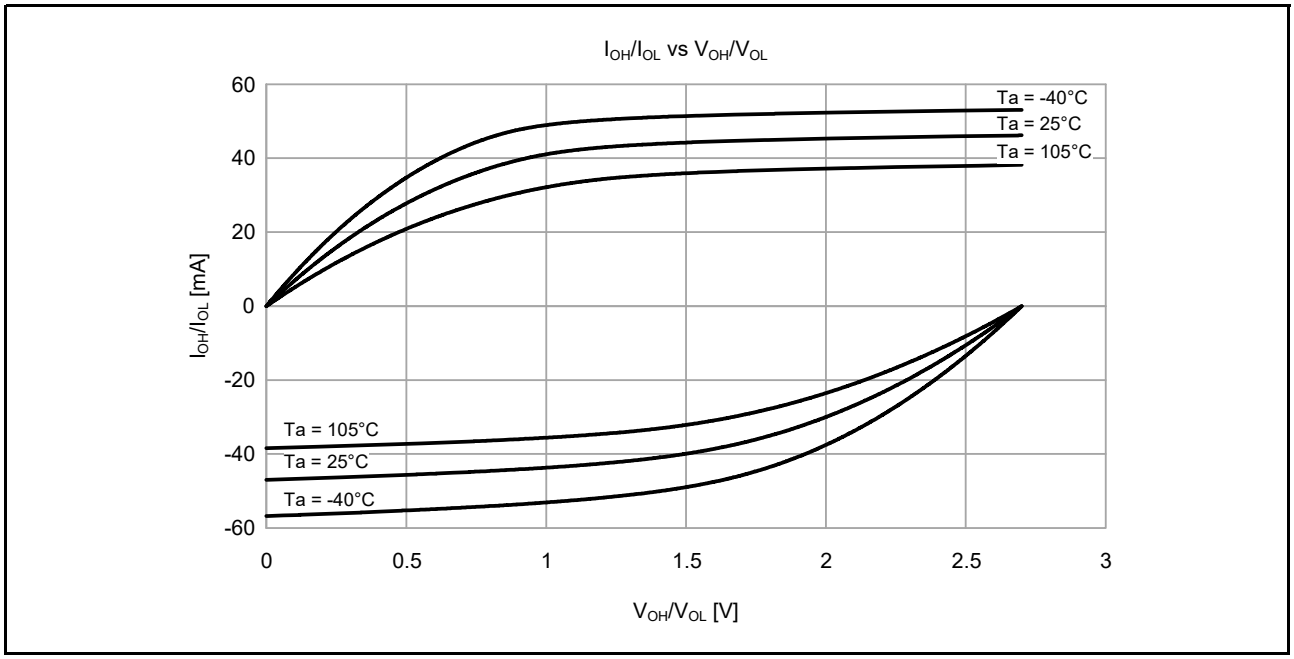


Figure 52.13 V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 2.7 V when middle drive output is selected (reference data)

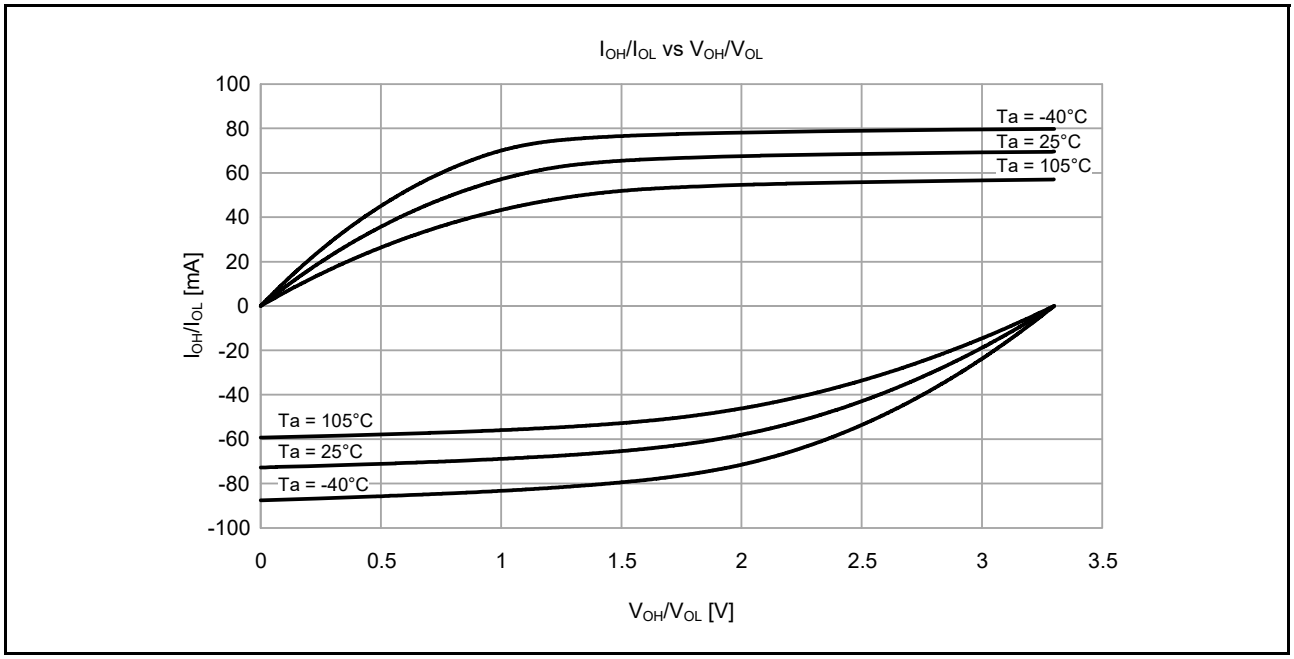


Figure 52.14 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 3.3 V when middle drive output is selected (reference data)**

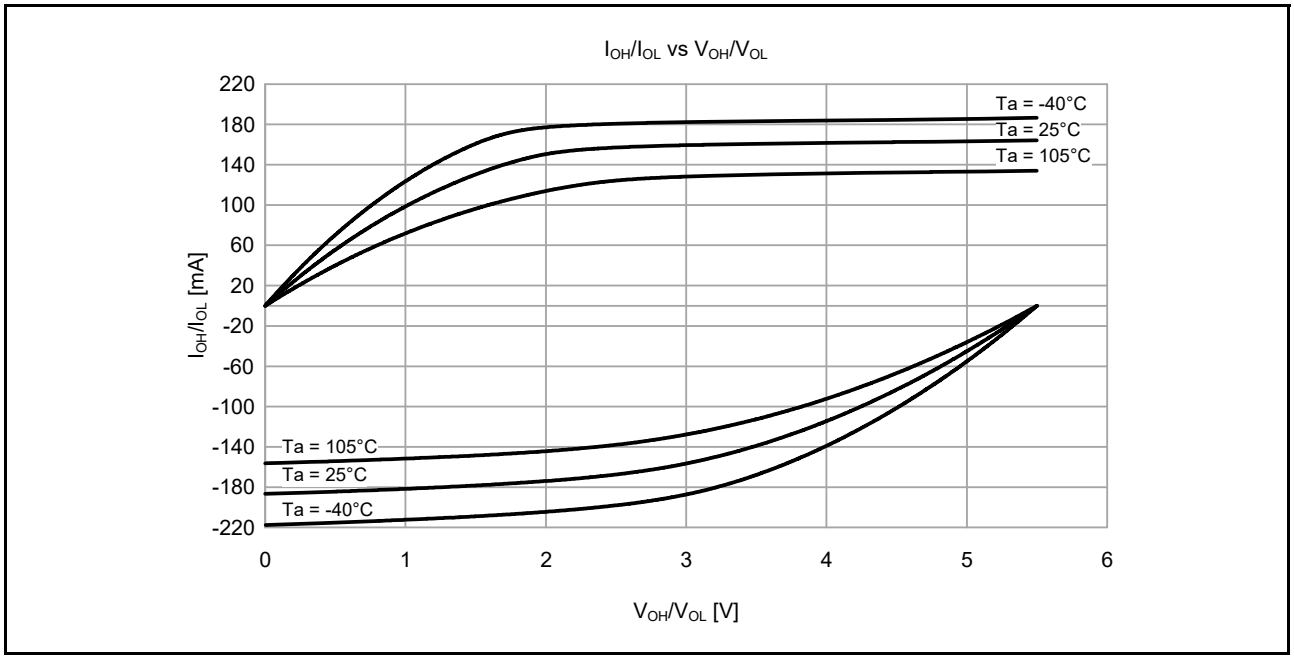


Figure 52.15 **V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 5.5 V when middle drive output is selected (reference data)**

52.2.8 IIC I/O Pin Output Characteristics

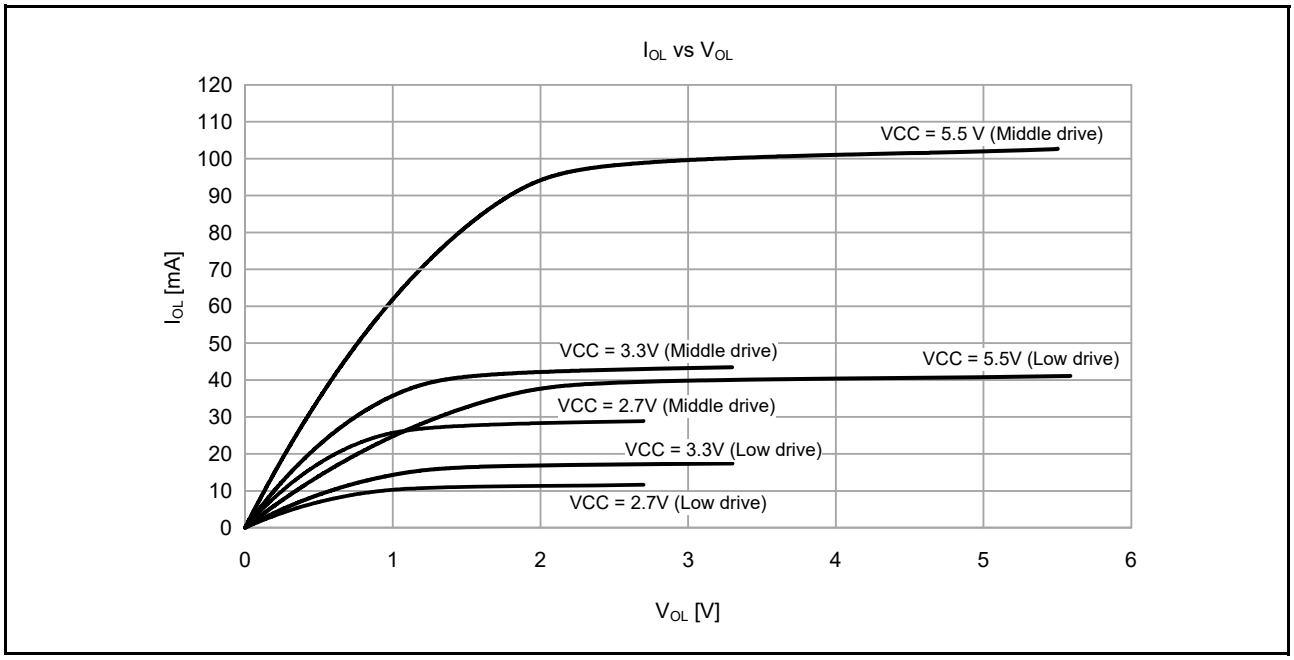


Figure 52.16 V_{OH}/V_{OL} and I_{OH}/I_{OL} voltage characteristics at Ta = 25°C

52.2.9 Operating and Standby Current

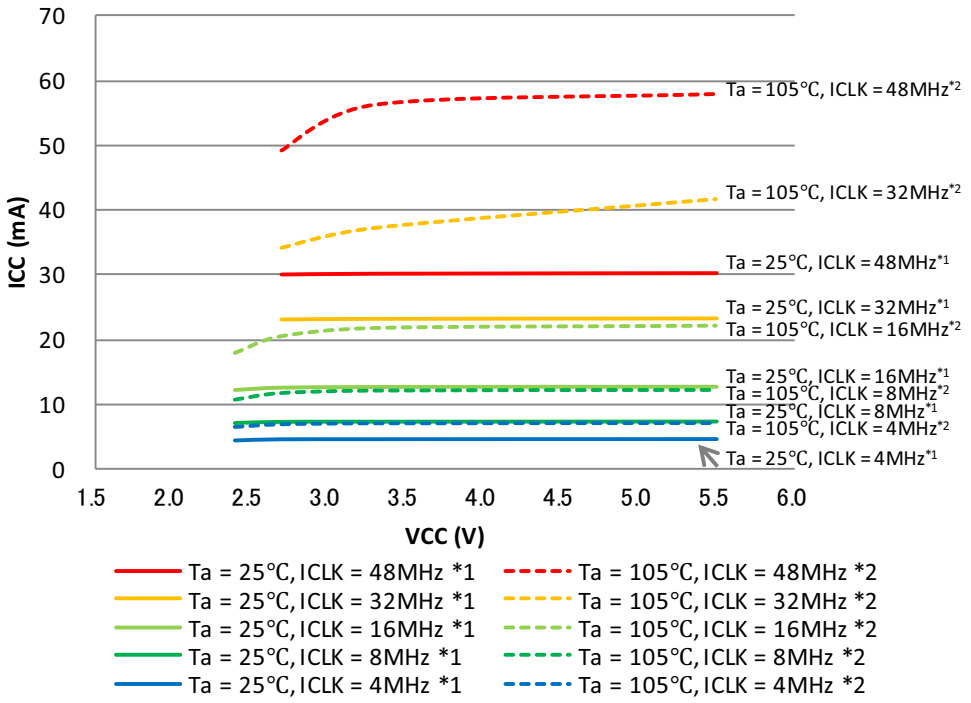
Table 52.11 Operating and standby current (1) (1 of 2)
Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter					Symbol	Typ*10	Max	Unit	Test conditions
Supply current*1	High-speed mode*2	Normal mode	All peripheral clock disabled, while (1) code executing from flash*5	ICLK = 48 MHz	I_{CC}	11.8	-	mA	*7
				ICLK = 32 MHz		8.6	-		
				ICLK = 16 MHz		5.1	-		
				ICLK = 8 MHz		3.4	-		
			All peripheral clock disabled, CoreMark code executing from flash*5	ICLK = 48 MHz		18.6	-		
				ICLK = 32 MHz		12.7	-		
				ICLK = 16 MHz		7.2	-		
				ICLK = 8 MHz		4.5	-		
			All peripheral clock enabled, while (1) code executing from flash*5	ICLK = 48 MHz		30.1	-		*9
				ICLK = 32 MHz		23.2	-		*8
				ICLK = 16 MHz		12.6	-		
				ICLK = 8 MHz		7.3	-		
			All peripheral clock enabled, code executing from SRAM*5	ICLK = 48 MHz		-	75.0		*9
		Sleep mode	All peripheral clock disabled*5	ICLK = 48 MHz		6.4	-		*7
				ICLK = 32 MHz		4.7	-		
				ICLK = 16 MHz		3.2	-		
				ICLK = 8 MHz		2.4	-		
			All peripheral clock enabled*5	ICLK = 48 MHz		24.7	-		*9
				ICLK = 32 MHz		19.2	-		*8
				ICLK = 16 MHz		10.7	-		
				ICLK = 8 MHz		6.4	-		
		Increase during BGO operation*6				2.5	-		-
	Middle-speed mode*2	Normal mode	All peripheral clock disabled, while (1) code executing from flash*5	ICLK = 12 MHz	I_{CC}	3.6	-	mA	*7
				ICLK = 8 MHz		3.0	-		
				ICLK = 1 MHz		1.4	-		
			All peripheral clock disabled, CoreMark code executing from flash*5	ICLK = 12 MHz		5.2	-		
				ICLK = 8 MHz		4.0	-		
				ICLK = 1 MHz		1.6	-		
			All peripheral clock enabled, while (1) code executing from flash*5	ICLK = 12 MHz		9.4	-		*8
				ICLK = 8 MHz		6.9	-		
				ICLK = 1 MHz		2.2	-		
			All peripheral clock enabled, code executing from SRAM*5	ICLK = 12 MHz		-	30.0		
		Sleep mode	All peripheral clock disabled*5	ICLK = 12 MHz		2.2	-		*7
				ICLK = 8 MHz		2.0	-		
				ICLK = 1 MHz		1.3	-		
			All peripheral clock enabled*5	ICLK = 12 MHz		7.9	-		*8
				ICLK = 8 MHz		5.9	-		
				ICLK = 1 MHz		2.1	-		
		Increase during BGO operation*6				2.5	-		-

Table 52.11 Operating and standby current (1) (2 of 2)
Conditions: VCC = AVCC0 = 1.6 to 5.5 V

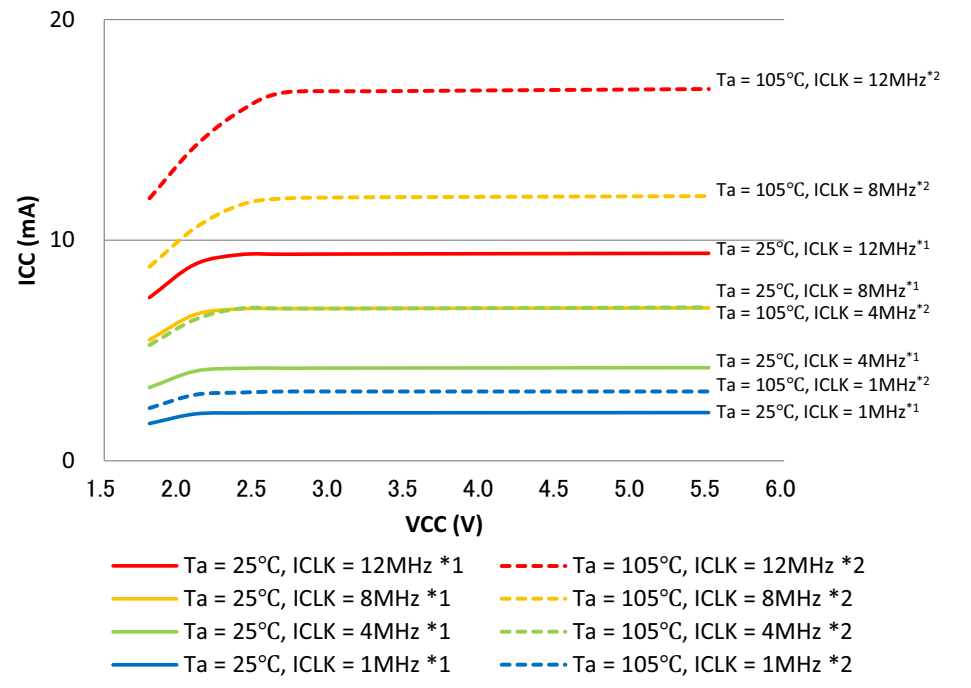
Parameter					Symbol	Typ*10	Max	Unit	Test conditions
Supply current*1	Low-speed mode*3	Normal mode	All peripheral clock disabled, while (1) code executing from flash*5	ICLK = 1 MHz	I_{CC}	0.5	-	mA	*7
			All peripheral clock disabled, CoreMark code executing from flash*5	ICLK = 1 MHz		0.7	-		
			All peripheral clock enabled, while (1) code executing from flash*5	ICLK = 1 MHz		1.5	-		*8
			All peripheral clock enabled, code executing from SRAM*5	ICLK = 1 MHz		-	3.2		
		Sleep mode	All peripheral clock disabled*5	ICLK = 1 MHz		0.4	-		*7
			All peripheral clock enabled*5	ICLK = 1 MHz		1.3	-		*8
	Low-voltage mode*3	Normal mode	All peripheral clock disabled, while (1) code executing from flash*5	ICLK = 4 MHz	I_{CC}	2.5	-	mA	*7
			All peripheral clock disabled, CoreMark code executing from flash*5	ICLK = 4 MHz		3.0	-		
			All peripheral clock enabled, while (1) code executing from flash*5	ICLK = 4 MHz		4.5	-		*8
			All peripheral clock enabled, code executing from SRAM*5	ICLK = 4 MHz		-	11.2		
		Sleep mode	All peripheral clock disabled*5	ICLK = 4 MHz		2.0	-		*7
			All peripheral clock enabled*5	ICLK = 4 MHz		4.0	-		*8
	Subosc-speed mode*4	Normal mode	All peripheral clock disabled, while (1) code executing from flash*5	ICLK = 32.768 kHz	I_{CC}	13.5	-	µA	*8
			All peripheral clock enabled, while (1) code executing from flash*5	ICLK = 32.768 kHz		25.0	-		
			All peripheral clock enabled, code executing from SRAM*5	ICLK = 32.768 kHz		-	214.1		
		Sleep mode	All peripheral clock disabled*5	ICLK = 32.768 kHz		9.5	-		
			All peripheral clock enabled*5	ICLK = 32.768 kHz		21.0	-		

Note 1. Supply current values do not include output charge/discharge current from all pins. The values apply when internal pull-up MOSs are in the off state.
Note 2. The clock source is HOCO.
Note 3. The clock source is MOCO.
Note 4. The clock source is the sub-clock oscillator.
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. FCLK, BCLK, PCLKA, PCLKB, PCLKC and PCLKD are set to divided by 64.
Note 8. FCLK, BCLK, PCLKA, PCLKB, PCLKC and PCLKD are the same frequency as that of ICLK.
Note 9. FCLK, BCLK, and PCLKB are set to divided by 2 and PCLKA, PCLKC and PCLKD are the same frequency as that of ICLK.
Note 10. VCC = 3.3 V.



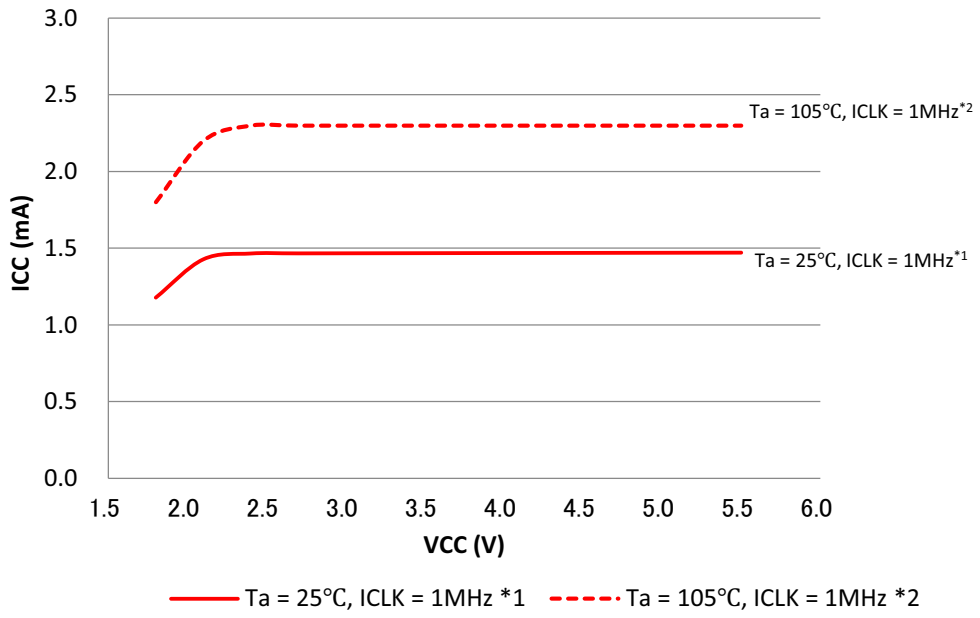
Note 1. All peripheral operations except any BGO operation are operating normally. This is the average of the actual measurements of the sample cores during product evaluation.
Note 2. All peripheral operations except any BGO operation are operating at maximum. This is the average of the actual measurements for the upper-limit samples during product evaluation.

Figure 52.17 Voltage dependency in High-speed operating mode (reference data)



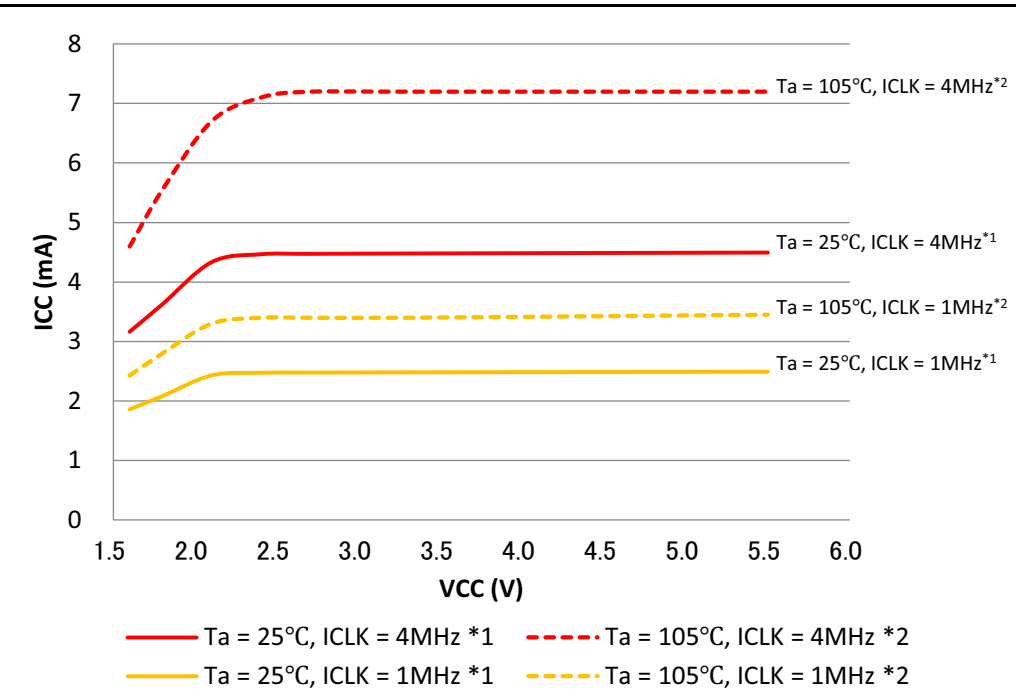
Note 1. All peripheral operations except any BGO operation are operating normally. This is the average of the actual measurements of the sample cores during product evaluation.
Note 2. All peripheral operations except any BGO operation are operating at maximum. This is the average of the actual measurements for the upper-limit samples during product evaluation.

Figure 52.18 Voltage dependency in Middle-speed mode (reference data)



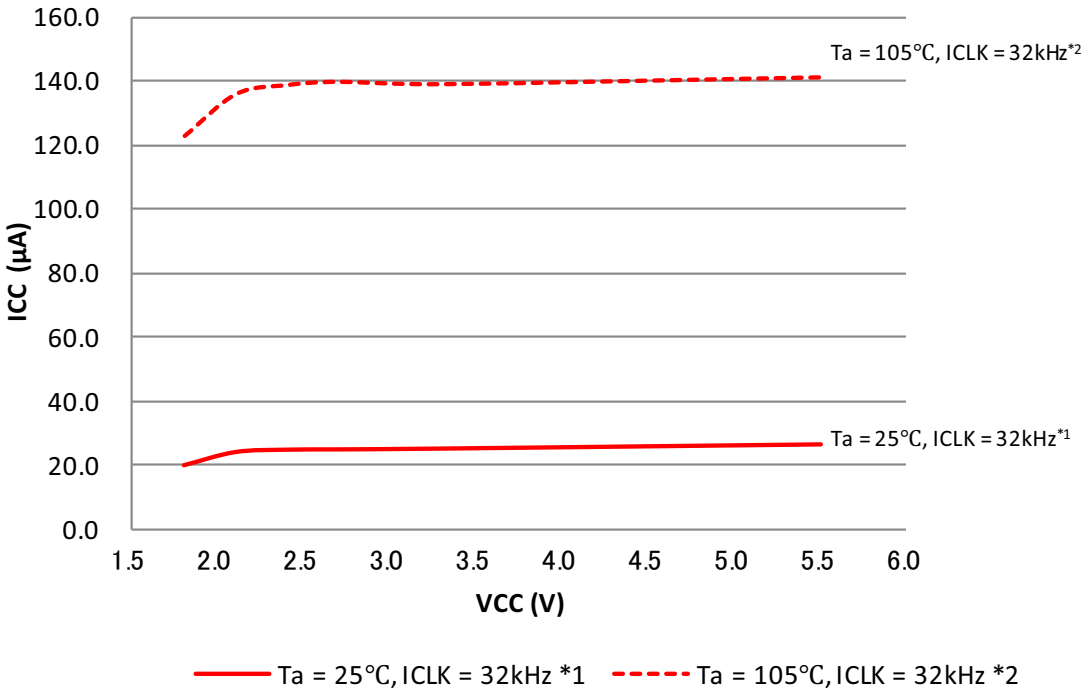
Note 1. All peripheral operations except any BGO operation are operating normally. This is the average of the actual measurements of the sample cores during product evaluation.
Note 2. All peripheral operations except any BGO operation are operating at maximum. This is the average of the actual measurements for the upper-limit samples during product evaluation.

Figure 52.19 Voltage dependency in Low-speed mode (reference data)



Note 1. All peripheral operations except any BGO operation are operating normally. This is the average of the actual measurements of the sample cores during product evaluation.
Note 2. All peripheral operations except any BGO operation are operating at maximum. This is the average of the actual measurements for the upper-limit samples during product evaluation.

Figure 52.20 Voltage dependency in Low-voltage mode (reference data)



Note 1. All peripheral operations except any BGO operation are operating normally. This is the average of the actual measurements of the sample cores during product evaluation.
Note 2. All peripheral operations except any BGO operation are operating at maximum. This is the average of the actual measurements for the upper-limit samples during product evaluation.

Figure 52.21 Voltage dependency in Subosc-speed mode (reference data)

Table 52.12 Operating and standby current (2)
Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter			Symbol	Typ*4	Max	Unit	Test conditions
Supply current*1	Software Standby mode*2	T_a = 25°C	I_{CC}	0.9	6.0	µA	PSMCR.PSMC[1:0] = 01b (48 KB SRAM on)
		T_a = 55°C		1.6	12.2		
		T_a = 85°C		4.8	27.1		
		T_a = 105°C		12.2	66.7		
		T_a = 25°C		1.1	7.5		PSMCR.PSMC[1:0] = 00b (All SRAM on)
		T_a = 55°C		2.2	17.0		
		T_a = 85°C		7.5	43.3		
		T_a = 105°C		19.6	105.9		
	Increment for RTC operation with low-speed on-chip oscillator*3			0.5	-		-
	Increment for RTC operation with sub-clock oscillator*3			0.5	-		SOMCR.SODRV[1:0] are 11b (Low power mode 3)
				1.6	-		SOMCR.SODRV[1:0] are 00b (Normal mode)

Note 1. Supply current values do not include output charge/discharge current from all pins. The values apply when internal pull-up MOSs are in the off state.
Note 2. The IWDT and LVD are not operating.
Note 3. Includes the current of sub-oscillation circuit or low-speed on-chip oscillator.
Note 4. VCC = 3.3 V.

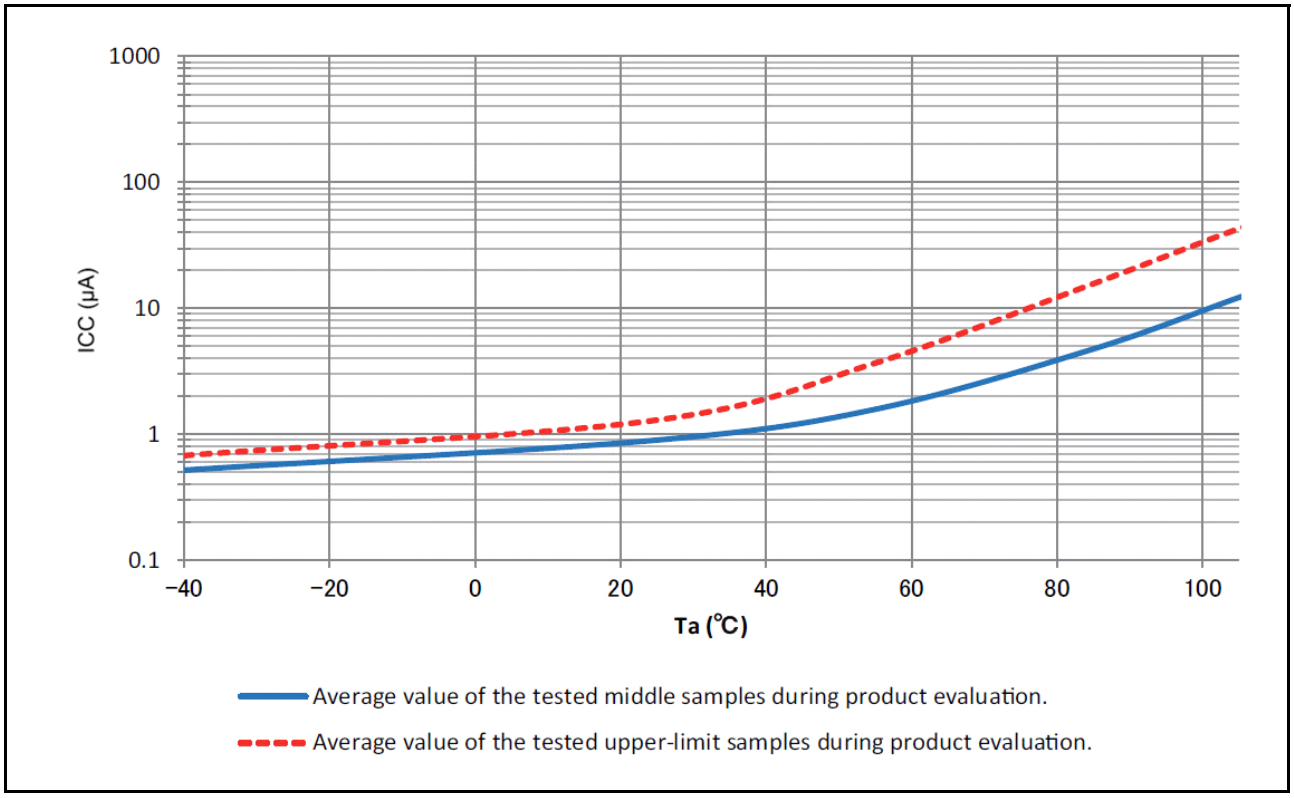


Figure 52.22 Temperature dependency in Software Standby mode 48 KB SRAM on (reference data)

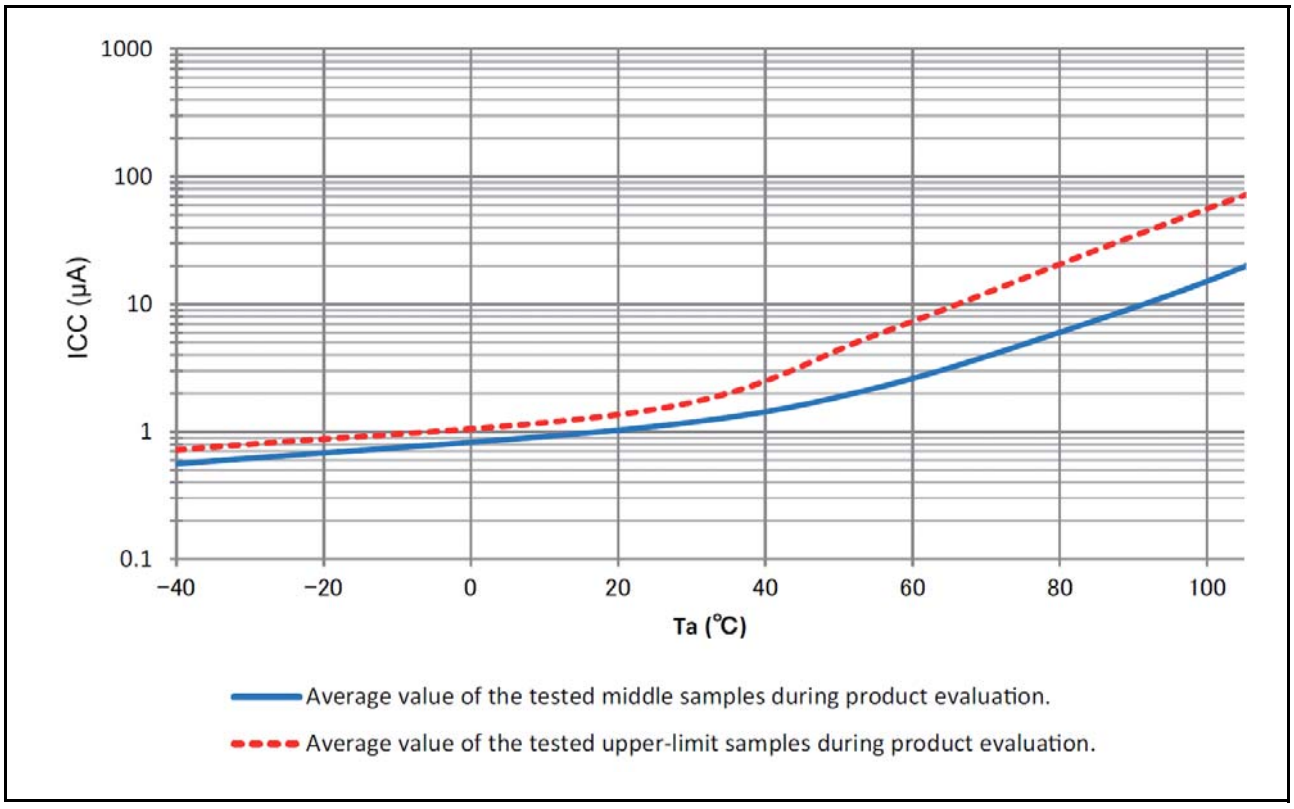
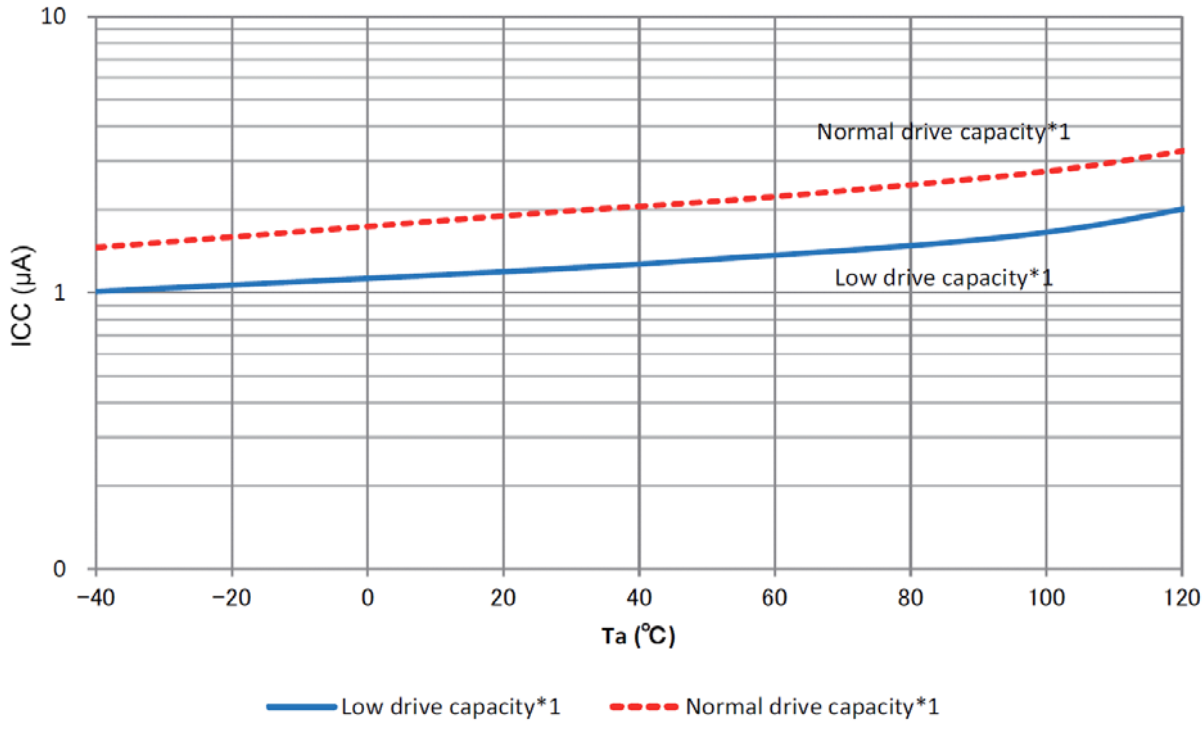


Figure 52.23 Temperature dependency in Software Standby mode all SRAM on (reference data)

Table 52.13 Operating and standby current (3)
Conditions: VCC = AVCC0 = 0V, VBATT = 1.6 to 3.6 V, VSS = AVSS0 = 0V

Parameter			Symbol	Typ	Max	Unit	Test conditions
Supply current*1	RTC operation when VCC is off	T_a = 25°C	I_{CC}	1.1	-	µA	VBATT = 2.0 V SOMCR.SORDRV[1:0] = 11b (Low power mode 3)
		T_a = 55°C		1.2	-		
		T_a = 85°C		1.4	-		
		T_a = 105°C		1.6	-		
		T_a = 25°C		1.2	-		VBATT = 3.3 V SOMCR.SORDRV[1:0] = 11b (Low power mode 3)
		T_a = 55°C		1.3	-		
		T_a = 85°C		1.5	-		
		T_a = 105°C		1.7	-		
		T_a = 25°C		1.8	-		VBATT = 2.0 V SOMCR.SORDRV[1:0] = 00b (Normal mode)
		T_a = 55°C		2.1	-		
		T_a = 85°C		2.4	-		
		T_a = 105°C		2.7	-		
		T_a = 25°C		1.9	-		VBATT = 3.3 V SOMCR.SORDRV[1:0] = 00b (Normal mode)
		T_a = 55°C		2.2	-		
		T_a = 85°C		2.5	-		
		T_a = 105°C		2.8	-		

Note 1. Supply current values do not include output charge/discharge current from all pins. The values apply when internal pull-up MOSs are in the off state.



Note 1. Average value of the tested middle sample during product evaluation.

Figure 52.24 Temperature dependency of RTC operation with VCC off (reference data)

Table 52.14 Operating and standby current (4)
Conditions: VCC = AVCC0 = 1.6 to 5.5 V, VREFH0 = 2.7 V to AVCC0

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Analog power supply current	During A/D conversion (at high-speed conversion)		I_{AVCC}	-	-	3.0	mA	-
	During A/D conversion (at low power conversion)			-	-	1.0	mA	-
	During D/A conversion (per channel)*1			-	0.4	0.8	mA	-
	Waiting for A/D and D/A conversion (all units)*6			-	-	1.0	µA	-
Reference power supply current	During A/D conversion		I_{REFH0}	-	-	150	µA	-
	Waiting for A/D conversion (all units)			-	-	60	nA	-
	During D/A conversion		I_{REFH}	-	50	100	µA	-
	Waiting for D/A conversion (all units)			-	-	100	µA	-
Temperature sensor			I_{TNS}	-	75	-	µA	-
Low power Analog Comparator operating current	Window mode		I_{CMPLP}	-	15	-	µA	-
	Comparator High-speed mode			-	10	-	µA	-
	Comparator Low-speed mode			-	2	-	µA	-
High-Speed Analog Comparator operating current			I_{CMPHS}	-	70	100	µA	AVCC0 ≥ 2.7 V
Operational Amplifier operating current	Low power mode	1 unit operating	I_{AMP}	-	2.5	4.0	µA	-
		2 units operating		-	4.5	8.0	µA	-
		3 units operating		-	6.5	11.0	µA	-
		4 units operating		-	8.5	14.0	µA	-
	High-speed mode	1 unit operating		-	140	220	µA	-
		2 units operating		-	280	410	µA	-
		3 units operating		-	420	600	µA	-
		4 units operating		-	560	780	µA	-
LCD operating current	External resistance division method $f_{LCD} = f_{SUB}$ = 128 Hz, 1/3 bias, and 4-time slice		I_{LCD1}*5	-	0.34	-	µA	-
	Internal voltage boosting method $f_{LCD} = f_{SUB}$ = 128 Hz, 1/3 bias, and 4-time slice		I_{LCD2}*5	-	0.92	-	µA	-
	Capacitor split method $f_{LCD} = f_{SUB}$ = 128 Hz, 1/3 bias, and 4-time slice		I_{LCD3}*5	-	0.19	-	µA	-
USB operating current	During USB communication operation under the following settings and conditions: • Host controller operation is set to Full-speed mode Bulk OUT transfer (64 bytes) × 1, bulk IN transfer (64 bytes) × 1 • Connect peripheral devices via a 1-meter USB cable from the USB port.		I_{USBH}*2	-	4.3 (VCC) 0.9 (VCC_USB)*4	-	mA	-
	During USB communication operation under the following settings and conditions: • Function controller operation is set to Full-speed mode Bulk OUT transfer (64 bytes) × 1, bulk IN transfer (64 bytes) × 1 • Connect the host device via a 1-meter USB cable from the USB port.		I_{USBF}*2	-	3.6 (VCC) 1.1 (VCC_USB)*4	-	mA	-
	During suspended state under the following setting and conditions: • Function controller operation is set to Full-speed mode (pull up the USB_DP pin) • Software standby mode • Connect the host device via a 1-meter USB cable from the USB port.		I_{SUSP}*3	-	0.35 (VCC) 170 (VCC_USB)*4	-	µA	-

Note 1. Includes the reference power supply current in the power supply current value for D/A conversion.
Note 2. Includes current consumed by the USBFS only.
Note 3. Includes current supplied from the pull-up resistor of the USB_DP pin to the pull-down resistor of the host device, in addition to the current consumed by the MCU during the suspended state.
Note 4. When VCC = VCC_USB = 3.3 V.
Note 5. Includes current flowing to the LCD controller only. Does not include current flowing through the LCD panel.
Note 6. When the MSTPCRD.MSTPD16 (14-Bit A/D Converter Module Stop bit) is in the module-stop state.

52.2.10 VCC Rise and Fall Gradient and Ripple Frequency

Table 52.15 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		0.02	-	-		
	SCI/USB Boot mode*2		0.02	-	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 the OFS1.LVDAS bit.

Table 52.16 Rising and falling gradient and ripple frequency characteristics
Conditions: VCC = AVCC0 = VCC_USB = 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 VCC change exceeds VCC ±10%, the allowable voltage change rising/falling gradient dt/dVCC must be met.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Allowable ripple frequency	$f_{r\ (VCC)}$	-	-	10	kHz	Figure 52.25 $V_{r\ (VCC)} \leq VCC \times 0.2$
		-	-	1	MHz	Figure 52.25 $V_{r\ (VCC)} \leq VCC \times 0.08$
		-	-	10	MHz	Figure 52.25 $V_{r\ (VCC)} \leq VCC \times 0.06$
Allowable voltage change rising and falling gradient	dt/dVCC	1.0	-	-	ms/V	When VCC change exceeds VCC ±10%

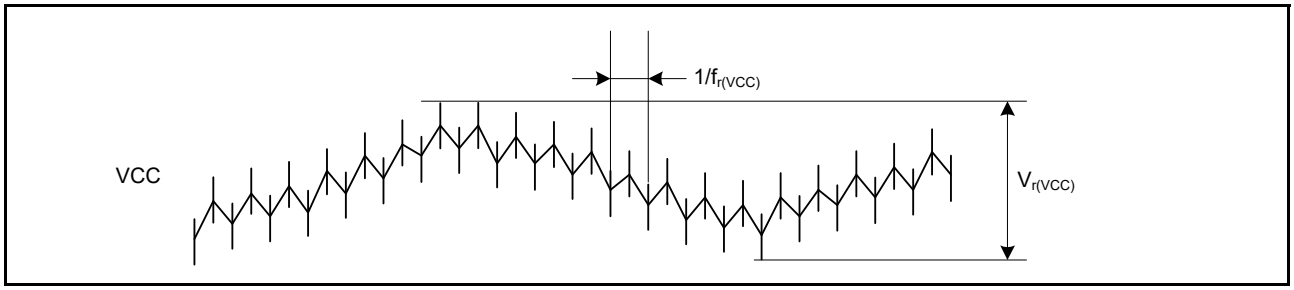


Figure 52.25 Ripple waveform

52.3 AC Characteristics

52.3.1 Frequency

Table 52.17 Operation frequency value in High-speed operating mode

Conditions: VCC = AVCC0 = 2.4 to 5.5 V

Parameter			Symbol	Min	Typ	Max*5	Unit
Operation frequency	System clock (ICLK)*4	2.7 to 5.5 V	f	0.032768	-	48	MHz
		2.4 to 2.7 V		0.032768	-	16	
	FlashIF clock (FCLK)*1, *2, *4	2.7 to 5.5 V		0.032768	-	32	
		2.4 to 2.7 V		0.032768	-	16	
	Peripheral module clock (PCLKA)*4	2.7 to 5.5 V		-	-	48	
		2.4 to 2.7 V		-	-	16	
	Peripheral module clock (PCLKB)*4	2.7 to 5.5 V		-	-	32	
		2.4 to 2.7 V		-	-	16	
	Peripheral module clock (PCLKC)*3, *4	2.7 to 5.5 V		-	-	64	
		2.4 to 2.7 V		-	-	16	
	Peripheral module clock (PCLKD)*4	2.7 to 5.5 V		-	-	64	
		2.4 to 2.7 V		-	-	16	
	External bus clock (BCLK)*4	2.7 to 5.5 V		-	-	24	
		2.4 to 2.7 V		-	-	16	
	EBCLK pin output	2.7 to 5.5 V		-	-	12	
		2.4 to 2.7 V		-	-	8	

Note 1. The lower-limit frequency of FCLK is 1 MHz while programming or erasing the flash memory. When using FCLK 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 FCLK must be ±3.5% while programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.

Note 3. The lower-limit frequency of PCLKC is 4 MHz at 2.4 V or above and 1 MHz at below 2.4 V when the 14-bit A/D converter is in use.

Note 4. See section 9, Clock Generation Circuit for the relationship of frequencies between ICLK, PCLKA, PCLKB, PCLKC, PCLKD, FCLK, and BCLK.

Note 5. The maximum value of operation frequency does not include errors of the internal oscillator. The operation can be guaranteed with the errors of the internal oscillator. For details, on the range for the guaranteed operation, see Table 52.22, Clock timing.

Table 52.18 Operation frequency value in Middle-speed mode

Conditions: VCC = AVCC0 = 1.8 to 5.5 V

Parameter			Symbol	Min	Typ	Max*5	Unit
Operation frequency	System clock (ICLK)*4	2.7 to 5.5 V	f	0.032768	-	12	MHz
		2.4 to 2.7 V		0.032768	-	12	
		1.8 to 2.4 V		0.032768	-	8	
	FlashIF clock (FCLK)*1, *2, *4	2.7 to 5.5 V		0.032768	-	12	
		2.4 to 2.7 V		0.032768	-	12	
		1.8 to 2.4 V		0.032768	-	8	
	Peripheral module clock (PCLKA)*4	2.7 to 5.5 V		-	-	12	
		2.4 to 2.7 V		-	-	12	
		1.8 to 2.4 V		-	-	8	
	Peripheral module clock (PCLKB)*4	2.7 to 5.5 V		-	-	12	
		2.4 to 2.7 V		-	-	12	
		1.8 to 2.4 V		-	-	8	
	Peripheral module clock (PCLKC)*3, *4	2.7 to 5.5 V		-	-	12	
		2.4 to 2.7 V		-	-	12	
		1.8 to 2.4 V		-	-	8	
	Peripheral module clock (PCLKD)*4	2.7 to 5.5 V		-	-	12	
		2.4 to 2.7 V		-	-	12	
		1.8 to 2.4 V		-	-	8	
	External bus clock (BCLK)*4	2.7 to 5.5 V		-	-	12	
		2.4 to 2.7 V		-	-	12	
		1.8 to 2.4 V		-	-	8	
	EBCLK pin output	2.7 to 3.6 V		-	-	12	
		2.4 to 2.7 V		-	-	8	
		1.8 to 2.4 V		-	-	8	

Note 1. The lower-limit frequency of FCLK is 1 MHz while programming or erasing the flash memory. When using FCLK 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 FCLK must be ±3.5% while programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.

Note 3. The lower-limit frequency of PCLKC is 4 MHz at 2.4 V or above and 1 MHz at below 2.4 V when the 14-bit A/D converter is in use.

Note 4. See section 9, Clock Generation Circuit for the relationship of frequencies between ICLK, PCLKA, PCLKB, PCLKC, PCLKD, FCLK, and BCLK.

Note 5. The maximum value of operation frequency does not include errors of the internal oscillator. The operation can be guaranteed with the errors of the internal oscillator. For details on the range for the guaranteed operation, see Table 52.22, Clock timing.

Table 52.19 Operation frequency value in Low-speed mode

Conditions: VCC = AVCC0 = 1.8 to 5.5 V

Parameter			Symbol	Min	Typ	Max*4	Unit
Operation frequency	System clock (ICLK)*3	1.8 to 5.5 V	f	0.032768	-	1	MHz
	FlashIF clock (FCLK)*1, *3	1.8 to 5.5 V		0.032768	-	1	
	Peripheral module clock (PCLKA)*3	1.8 to 5.5 V		-	-	1	
	Peripheral module clock (PCLKB)*3	1.8 to 5.5 V		-	-	1	
	Peripheral module clock (PCLKC)*2, *3	1.8 to 5.5 V		-	-	1	
	Peripheral module clock (PCLKD)*3	1.8 to 5.5 V		-	-	1	
	External bus clock (BCLK)*3	1.8 to 5.5 V		-	-	1	
	EBCLK pin output	1.8 to 5.5 V		-	-	1	

Note 1. The lower-limit frequency of FCLK is 1 MHz while programming or erasing the flash memory.
Note 2. The lower-limit frequency of PCLKC is 1 MHz when the A/D converter is in use.
Note 3. See section 9, Clock Generation Circuit for the relationship of frequencies between ICLK, PCLKA, PCLKB, PCLKC, PCLKD, FCLK, and BCLK.
Note 4. The maximum value of operation frequency does not include errors of the internal oscillator. The operation can be guaranteed with the errors of the internal oscillator. For details on the range for the guaranteed operation, Table 52.22, Clock timing.

Table 52.20 Operation frequency value in Low-voltage mode

Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter			Symbol	Min	Typ	Max*5	Unit
Operation frequency	System clock (ICLK)*4	1.6 to 5.5 V	f	0.032768	-	4	MHz
	FlashIF clock (FCLK)*1, *2, *4	1.6 to 5.5 V		0.032768	-	4	
	Peripheral module clock (PCLKA)*4	1.6 to 5.5 V		-	-	4	
	Peripheral module clock (PCLKB)*4	1.6 to 5.5 V		-	-	4	
	Peripheral module clock (PCLKC)*3, *4	1.6 to 5.5 V		-	-	4	
	Peripheral module clock (PCLKD)*4	1.6 to 5.5 V		-	-	4	
	External bus clock (BCLK)*4	1.6 to 5.5 V		-	-	4	
	EBCLK pin output	1.8 to 5.5 V		-	-	4	
		1.6 to 1.8 V		-	-	2	

Note 1. The lower-limit frequency of FCLK is 1 MHz while programming or erasing the flash memory. When using FCLK 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 FCLK must be ±3.5% while programming or erasing the flash memory. Confirm the frequency accuracy of the clock source.
Note 3. The lower-limit frequency of PCLKC is 4 MHz at 2.4 V or above and 1 MHz at below 2.4 V when the 14-Bit A/D converter is in use.
Note 4. See section 9, Clock Generation Circuit for the relationship of frequencies between ICLK, PCLKA, PCLKB, PCLKC, PCLKD, FCLK, and BCLK.
Note 5. The maximum value of operation frequency does not include errors of the internal oscillator. The operation can be guaranteed with the errors of the internal oscillator. For details on the range for guaranteed operation, see Table 52.22, Clock timing.

Table 52.21 Operation frequency value in Subosc-speed mode
Conditions: VCC = AVCC0 = 1.8 to 5.5 V

Parameter			Symbol	Min	Typ	Max	Unit
Operation frequency	System clock (ICLK)*3	1.8 to 5.5 V	f	27.8528	32.768	37.6832	kHz
	FlashIF clock (FCLK)*1, *3	1.8 to 5.5 V		27.8528	32.768	37.6832	
	Peripheral module clock (PCLKA)*3	1.8 to 5.5 V		-	-	37.6832	
	Peripheral module clock (PCLKB)*3	1.8 to 5.5 V		-	-	37.6832	
	Peripheral module clock (PCLKC)*2, *3	1.8 to 5.5 V		-	-	37.6832	
	Peripheral module clock (PCLKD)*3	1.8 to 5.5 V		-	-	37.6832	
	External bus clock (BCLK)*3	1.8 to 5.5 V		-	-	37.6832	
	EBCLK pin output	1.8 to 5.5 V		-	-	37.6832	

Note 1. Programming and erasing the flash memory is not possible.
Note 2. The 14-bit A/D converter cannot be used.
Note 3. See section 9, Clock Generation Circuit for the relationship of frequencies between ICLK, PCLKA, PCLKB, PCLKC, PCLKD, FCLK, and BCLK.

52.3.2 Clock Timing

Table 52.22 Clock timing (1 of 2)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
EBCLK pin output cycle time	VCC = 2.7 V or above	t_{Bcyc}	83.3	-	-	ns	Figure 52.26
	VCC = 1.8 V or above		125	-	-		
	VCC = 1.6 V or above		500	-	-		
EBCLK pin output high pulse width	VCC = 2.7 V or above	t_{CH}	20	-	-	ns	
	VCC = 1.8 V or above		30	-	-		
	VCC = 1.6 V or above		150	-	-		
EBCLK pin output low pulse width	VCC = 2.7 V or above	t_{CL}	20	-	-	ns	
	VCC = 1.8 V or above		30	-	-		
	VCC = 1.6 V or above		150	-	-		
EBCLK pin output rise time	VCC = 2.7 V or above	t_{Cr}	-	-	15	ns	
	VCC = 2.4 V or above		-	-	25		
	VCC = 1.8 V or above		-	-	30		
	VCC = 1.6 V or above		-	-	50		
EBCLK pin output fall time	VCC = 2.7 V or above	t_{Cf}	-	-	15	ns	
	VCC = 2.4 V or above		-	-	25		
	VCC = 1.8 V or above		-	-	30		
	VCC = 1.6 V or above		-	-	50		
EXTAL external clock input cycle time		t_{Xcyc}	50	-	-	ns	Figure 52.27
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	2.4 ≤ VCC ≤ 5.5
			-	-	8		1.8 ≤ VCC < 2.4
			-	-	1		1.6 ≤ VCC < 1.8
Main clock oscillator oscillation frequency		f_{MAIN}	1	-	20	MHz	2.4 ≤ VCC ≤ 5.5
			1	-	8		1.8 ≤ VCC < 2.4
			1	-	4		1.6 ≤ VCC < 1.8
LOCO clock oscillation frequency		f_{LOCO}	27.8528	32.768	37.6832	kHz	-

Table 52.22 Clock timing (2 of 2)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
LOCO clock oscillation stabilization time		t_{LOCO}	-	-	100	µs	Figure 52.28
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	-
HOCO clock oscillation frequency		f_{HOCO24}	23.64	24	24.36	MHz	Ta = –40 to –20°C 1.8 ≤ VCC ≤ 5.5
			22.68	24	25.32		Ta = –40 to 85°C 1.6 ≤ VCC < 1.8
			23.76	24	24.24		Ta = –20 to 85°C 1.8 ≤ VCC ≤ 5.5
			23.52	24	24.48		Ta = 85 to 105°C 2.4 ≤ VCC ≤ 5.5
		f_{HOCO32}	31.52	32	32.48		Ta = –40 to -20°C 1.8 ≤ VCC ≤ 5.5
			30.24	32	33.76		Ta = –40 to 85°C 1.6 ≤ VCC < 1.8
			31.68	32	32.32		Ta = –20 to 85°C 1.8 ≤ VCC ≤ 5.5
			31.36	32	32.64		Ta = 85 to 105°C 2.4 ≤ VCC ≤ 5.5
		f_{HOCO48}*4	47.28	48	48.72		Ta = –40 to –20°C 1.8 ≤ VCC ≤ 5.5
			47.52	48	48.48		Ta = –20 to 85°C 1.8 ≤ VCC ≤ 5.5
			47.04	48	48.96		Ta = 85°C to 105°C 2.4 ≤ VCC ≤ 5.5
		f_{HOCO64}*5	63.04	64	64.96		Ta = –40 to –20°C 2.4 ≤ VCC ≤ 5.5
			63.36	64	64.64		Ta = –20 to 85°C 2.4 ≤ VCC ≤ 5.5
			62.72	64	65.28		Ta = 85 to 105°C 2.4 ≤ VCC ≤ 5.5
HOCO clock oscillation stabilization time*6, *7	Except Low-Voltage mode	t_{HOCO24} t_{HOCO32}	-	-	37.1	µs	Figure 52.29
		t_{HOCO48}	-	-	43.3		
		t_{HOCO64}	-	-	80.6		
	Low-Voltage mode	t_{HOCO24} t_{HOCO32} t_{HOCO48} t_{HOCO64}	-	-	100.9		
PLL input frequency*2		f_{PLLIN}	4	-	12.5	MHz	-
PLL circuit oscillation frequency*2		f_{PLL}	24	-	64	MHz	-
PLL clock oscillation stabilization time*8		t_{PLL}	-	-	55.5	µs	Figure 52.31
PLL free-running oscillation frequency		f_{PLLFR}	-	8	-	MHz	-
Sub-clock oscillator oscillation frequency		f_{SUB}	-	32.768	-	kHz	-
Sub-clock oscillation stabilization time*3		t_{SUBOSC}	-	0.5	-	s	Figure 52.32

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. The VCC range that the PLL can be used is 2.4 to 5.5 V.
Note 3. After changing the setting of the SOSCCR.SOSTP bit to start sub-clock oscillator operation, only start using the sub-clock oscillator after the sub-clock oscillation stabilization wait time elapsed. Use the oscillator wait time value recommended by the oscillator manufacturer.
Note 4. The 48-MHz HOCO can be used within a VCC range of 1.8 V to 5.5 V.
Note 5. The 64-MHz HOCO can be used within a VCC range of 2.4 V to 5.5 V.
Note 6. This is a characteristic when HOCOCR.HCSTP bit is set to 0 (oscillation) in MOCO stop state.
When HOCOCR.HCSTP bit is set to 0 (oscillation) during MOCO oscillation, this specification is shortened by 1 µs.
Note 7. Whether stabilization time has elapsed can be confirmed by OSCSF.HOCOSF.
Note 8. This is a characteristic when PLLCR.PLLSTP bit is set to 0 (operation) in MOCO stop state.
When PLLCR.PLLSTP bit is set to 0 (operation) during MOCO oscillation, this specification is shortened by 1 µs.

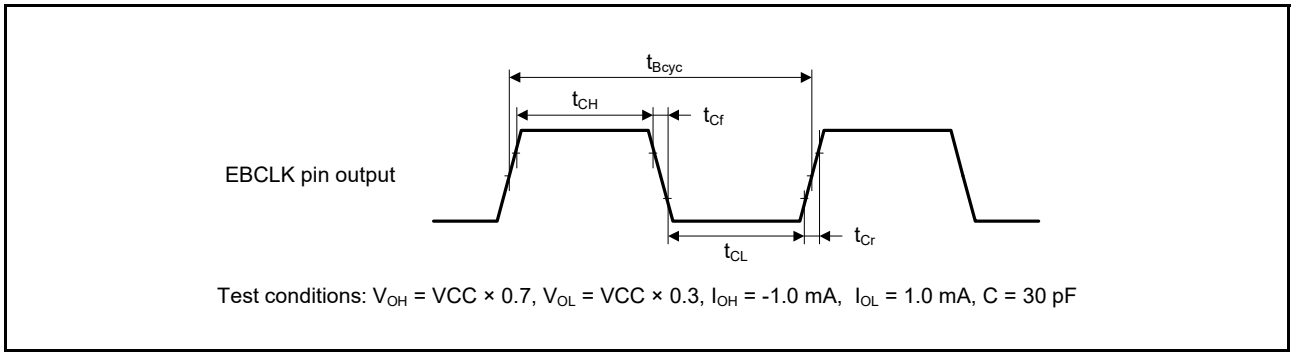


Figure 52.26 EBCLK pin output timing

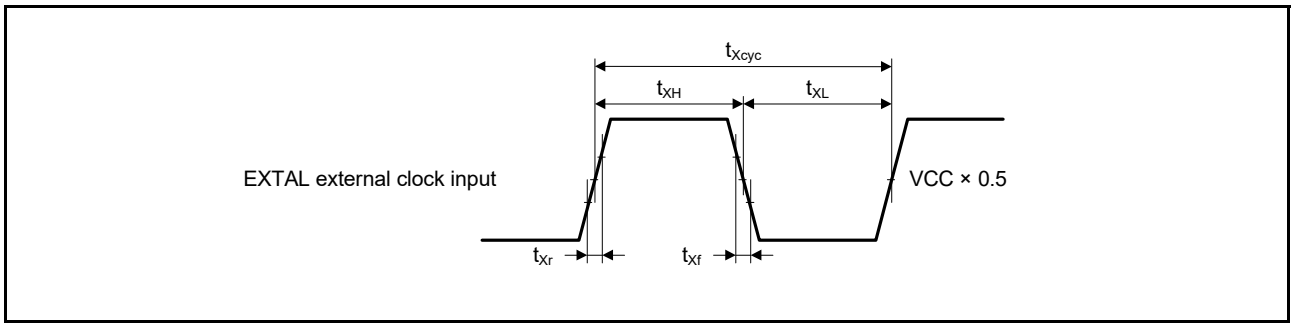


Figure 52.27 EXTAL external clock input timing

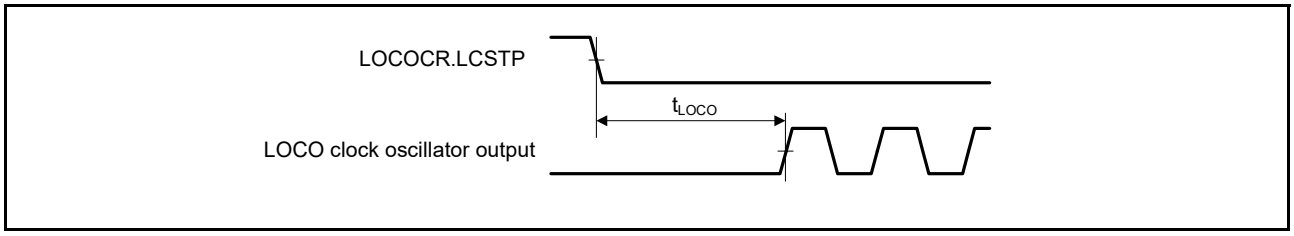


Figure 52.28 LOCO clock oscillation start timing

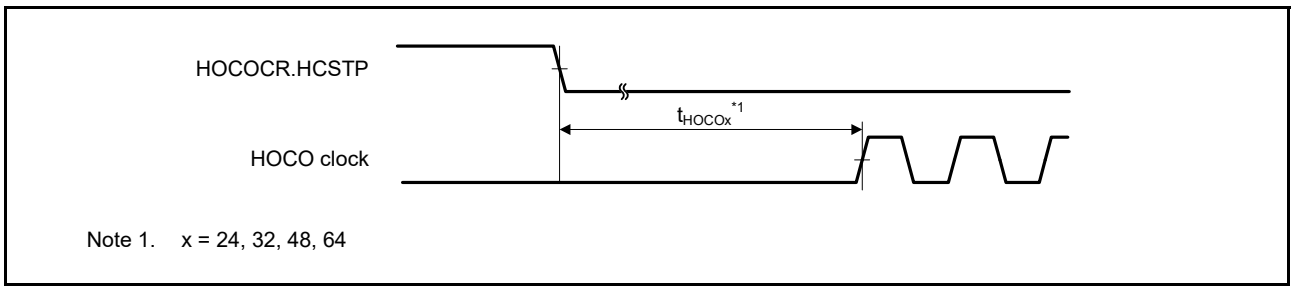


Figure 52.29 HOCO clock oscillation start timing (started by setting HOCOCR.HCSTP bit)

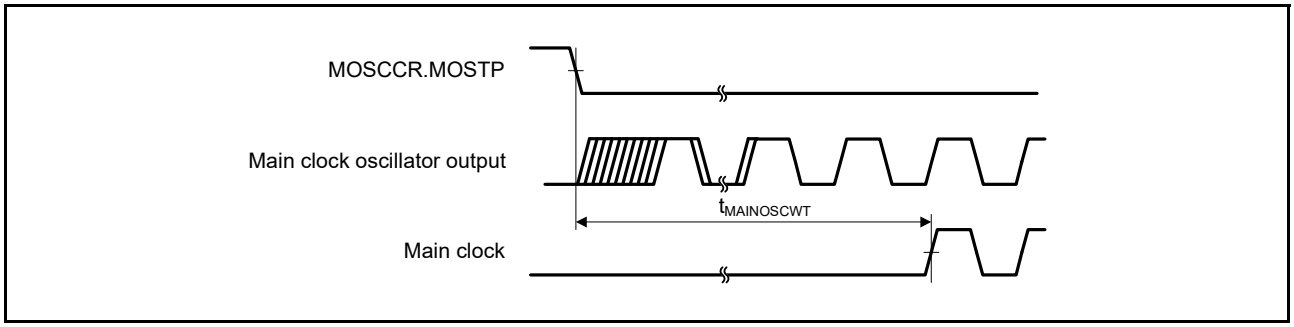


Figure 52.30 Main clock oscillation start timing

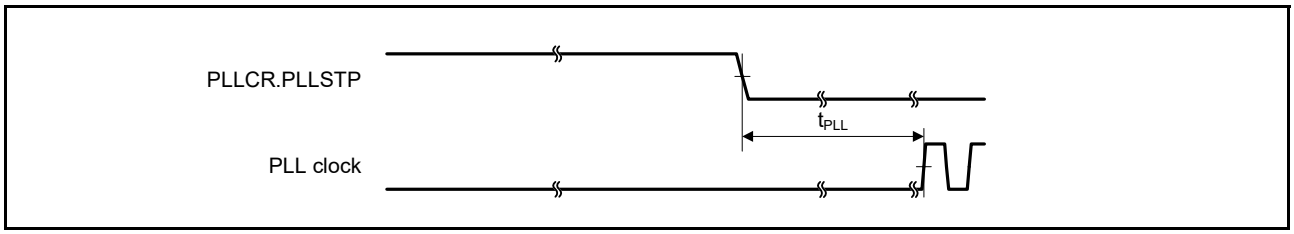


Figure 52.31 PLL clock oscillation start timing (PLL is operated after main clock oscillation has settled)

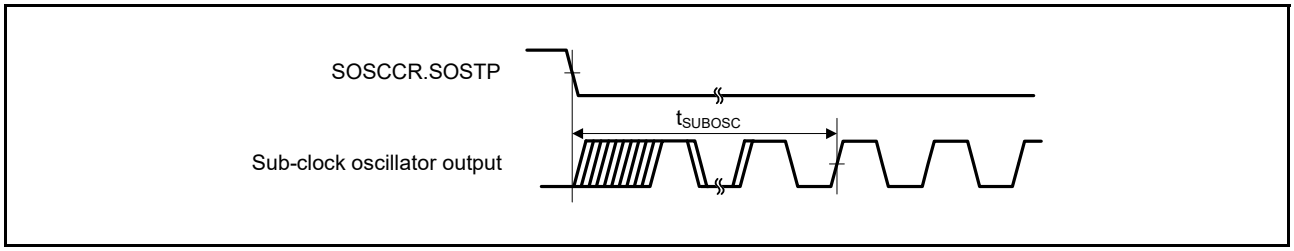


Figure 52.32 Sub-clock oscillation start timing

52.3.3 Reset Timing

Table 52.23 Reset timing

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
RES pulse width	At power-on	t_{RESWP}	3	-	-	ms	Figure 52.33
	Other than above	t_{RESW}	30	-	-	μs	Figure 52.34
Wait time after RES cancellation (at power-on)	LVD0: enable*1	t_{RESWT}	-	0.7	-	ms	Figure 52.33
	LVD0: disable*2		-	0.3	-		
Wait time after RES cancellation (during powered-on state)	LVD0: enable*1	t_{RESWT2}	-	0.5	-	ms	Figure 52.34
	LVD0: disable*2		-	0.05	-		
Wait time after internal reset cancellation (Watchdog timer reset, SRAM parity error reset, SRAM ECC error reset, bus master MPU error reset, bus slave MPU error reset, stack pointer error reset, software reset)	LVD0: enable*1	t_{RESWT3}	-	0.6	-	ms	
	LVD0: disable*2		-	0.15	-		

Note 1. When OFS1.LVDAS = 0.
Note 2. When OFS1.LVDAS = 1.

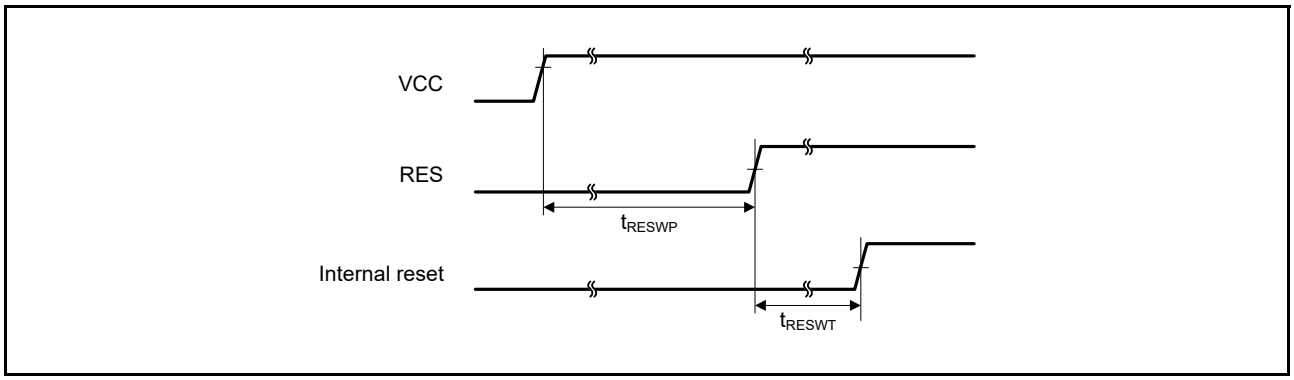


Figure 52.33 Reset input timing at power-on

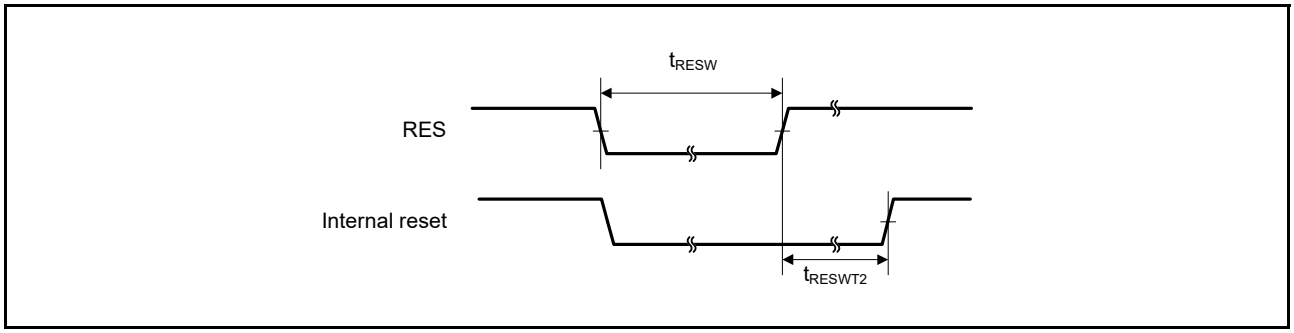


Figure 52.34 Reset input timing (1)

52.3.4 Wakeup Time

Table 52.24 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	Figure 52.35
			System clock source is PLL (48 MHz) with Main clock oscillator*2	t_{SBYPC}	-	2	3	ms	
		External clock input to main clock oscillator	System clock source is main clock oscillator (20 MHz)*3	t_{SBYEX}	-	14	25	µs	
			System clock source is PLL (48 MHz) with Main clock oscillator*3	t_{SBYPE}	-	53	76	µs	
		System clock source is HOCO*4 (HOCO clock is 32 MHz)		t_{SBYHO}	-	43	52	µs	
		System clock source is HOCO*4 (HOCO clock is 48 MHz)		t_{SBYHO}	-	44	52	µs	
		System clock source is HOCO*5 (HOCO clock is 64 MHz)		t_{SBYHO}	-	82	110	µs	
		System clock source is MOCO		t_{SBYMO}	-	16	25	µs	

Note 1. The division ratio of ICK, BCK, FCK, and PCKx 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 05h.
Note 3. The Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 00h.
Note 4. The HOCO Clock Wait Control Register (HOCOWTCR) is set to 05h.
Note 5. The HOCO Clock Wait Control Register (HOCOWTCR) is set to 06h.

Table 52.25 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 (12 MHz)*2	t_{SBYMC}	-	2	3	ms	Figure 52.35
			System clock source is PLL (24 MHz) with Main clock oscillator*2	t_{SBYPC}	-	2	3	ms	
		External clock input to main clock oscillator	System clock source is main clock oscillator (12 MHz)*3	t_{SBYEX}	-	2.9	10	µs	
			System clock source is PLL (24 MHz) with Main clock oscillator*3	t_{SBYPE}	-	49	76	µs	
		System clock source is HOCO*4		t_{SBYHO}	-	38	50	µs	
		System clock source is MOCO		t_{SBYMO}	-	3.5	5.5	µs	

Note 1. The division ratio of ICK, BCK, FCK, and PCKx 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 05h.
Note 3. The Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 00h.
Note 4. The system clock is 12 MHz.

Table 52.26 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 (1 MHz)*2	t_{SBYMC}	-	2	3	ms	Figure 52.35
		External clock input to main clock oscillator	System clock source is main clock oscillator (1 MHz)*3	t_{SBYEX}	-	28	50	µs	
		System clock source is MOCO		t_{SBYMO}	-	25	35	µs	

Note 1. The division ratio of ICK, BCK, FCK, and PCKx 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 05h.
Note 3. The Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 00h.

Table 52.27 Timing of recovery from Low power modes (4)

Parameter				Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Software Standby mode*1	Low-voltage mode	Crystal resonator connected to main clock oscillator	System clock source is main clock oscillator (4 MHz)*2	t_{SBYMC}	-	2	3	ms	Figure 52.35
		External clock input to main clock oscillator	System clock source is main clock oscillator (4 MHz)*3	t_{SBYEX}	-	108	130	µs	
		System clock source is HOCO		t_{SBYHO}	-	108	130	µs	

Note 1. The division ratio of ICK, BCK, FCK, and PCKx is the minimum division ratio within the allowable frequency range. The recovery time is determined by the system clock source. When multiple oscillators are active, the recovery time can be determined by the following expression.
Note 2. The Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 05h.
Note 3. The Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 00h.

Table 52.28 Timing of recovery from Low power modes (5)

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Software Standby mode*1	Subosc-speed mode	System clock source is sub-clock oscillator (32.768 kHz)	t_{SBYSC}	-	0.85	1	ms	Figure 52.35
		System clock source is LOCO (32.768 kHz)	t_{SBYLO}	-	0.85	1.2	ms	

Note 1. The sub-clock oscillator or LOCO itself continues to oscillate in Software Standby mode during Subosc-speed mode.

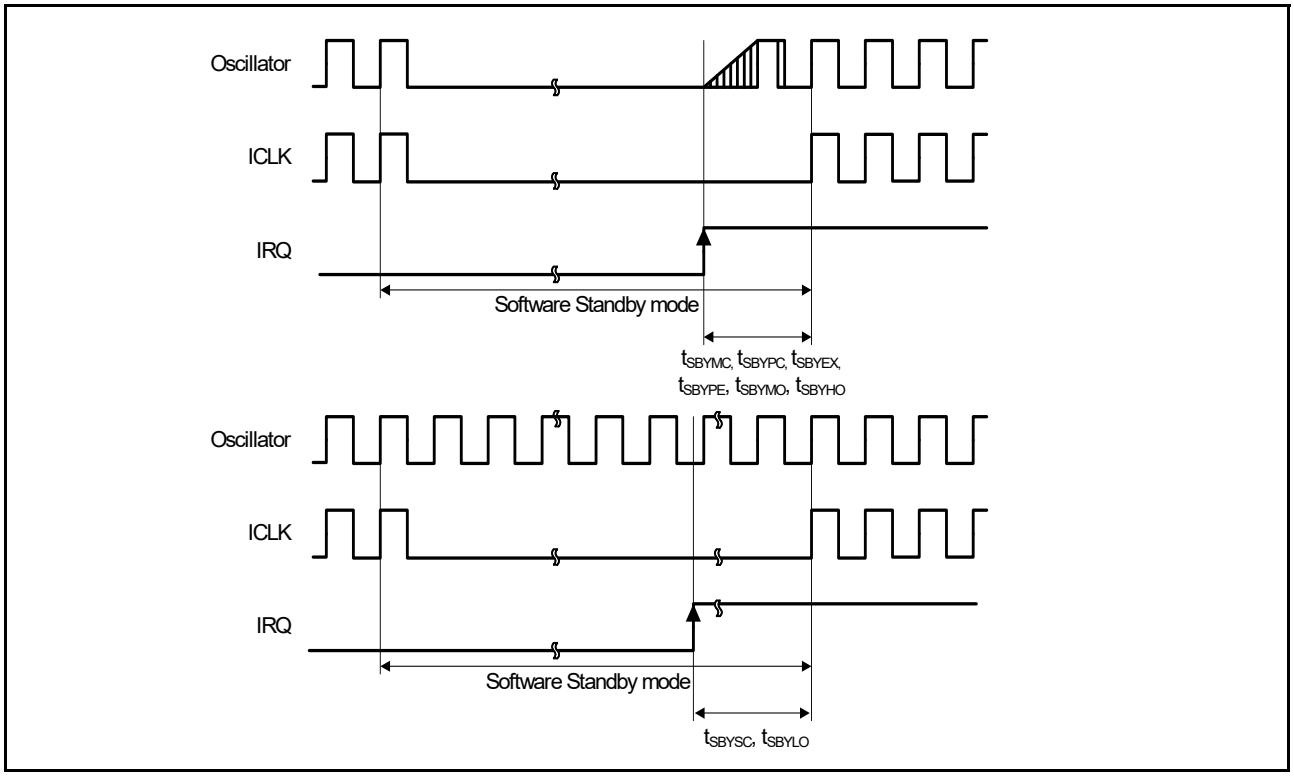


Figure 52.35 Software Standby mode cancellation timing

Table 52.29 Timing of recovery from Low power modes (6)

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}	-	36	45	µs	Figure 52.36
	Middle-speed mode System clock source is MOCO	t_{SNZ}	-	1.3	3.6	µs	
	Low-speed mode System clock source is MOCO	t_{SNZ}	-	10	13	µs	
	Low-voltage mode System clock source is HOCO	t_{SNZ}	-	87	110	µs	

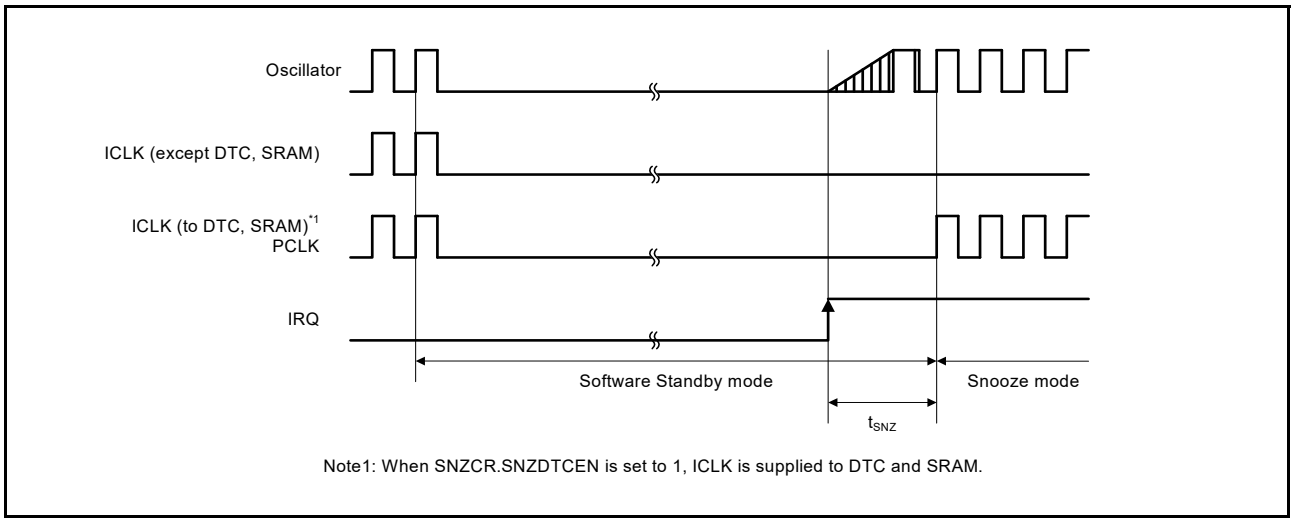


Figure 52.36 Recovery timing from Software Standby mode to Snooze mode

52.3.5 NMI and IRQ Noise Filter

Table 52.30 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 1. t_{Pcyc} indicates the cycle of PCLKB.
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 15).

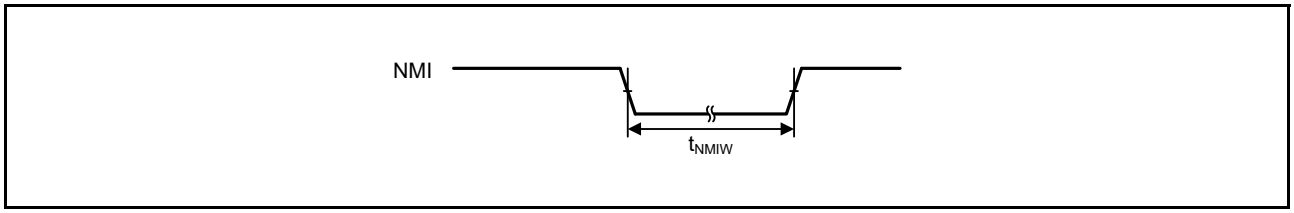


Figure 52.37 NMI interrupt input timing

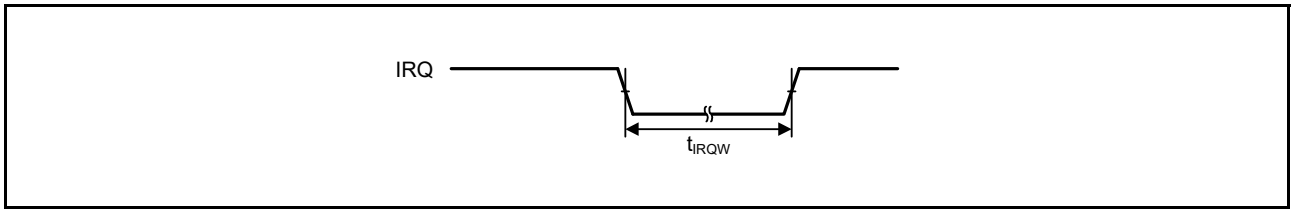


Figure 52.38 IRQ interrupt input timing

52.3.6 Bus Timing

Table 52.31 Bus timing (1)
Conditions: Low drive output is selected in the Port Drive Capability bit in PmnPFS register
VCC = AVCC0 = 2.7 to 5.5 V
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF

Parameter	Symbol	Min	Max	Unit	Test conditions
Address delay	t_{AD}	-	55	ns	Figure 52.39 to Figure 52.42
Byte control delay	t_{BCD}	-	55	ns	
CS delay	t_{CSD}	-	55	ns	
RD delay	t_{RSD}	-	55	ns	
Read data setup time	t_{RDS}	37	-	ns	
Read data hold time	t_{RDH}	0	-	ns	
WR delay	t_{WRD}	-	55	ns	
Write data delay	t_{WDD}	-	55	ns	
Write data hold time	t_{WDH}	0	-	ns	
WAIT setup time	t_{WTS}	37	-	ns	Figure 52.43
WAIT hold time	t_{WTH}	0	-	ns	

Table 52.32 Bus timing (2)
Conditions: Low drive output is selected in the Port Drive Capability bit in the PmnPFS register
VCC = AVCC0 = 2.4 to 2.7 V
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF

Parameter	Symbol	Min	Max	Unit	Test conditions
Address delay	t_{AD}	-	55	ns	Figure 52.39 to Figure 52.42
Byte control delay	t_{BCD}	-	55	ns	
CS delay	t_{CSD}	-	55	ns	
RD delay	t_{RSD}	-	55	ns	
Read data setup time	t_{RDS}	45	-	ns	
Read data hold time	t_{RDH}	0	-	ns	
WR delay	t_{WRD}	-	55	ns	
Write data delay	t_{WDD}	-	55	ns	
Write data hold time	t_{WDH}	0	-	ns	
WAIT setup time	t_{WTS}	45	-	ns	Figure 52.43
WAIT hold time	t_{WTH}	0	-	ns	

Table 52.33 Bus timing (3) (1 of 2)
Conditions: Low drive output is selected in the Port Drive Capability bit in the PmnPFS register
VCC = AVCC0 = 1.8 to 2.4 V
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF

Parameter	Symbol	Min	Max	Unit	Test conditions
Address delay	t_{AD}	-	90	ns	Figure 52.39 to Figure 52.42
Byte control delay	t_{BCD}	-	90	ns	
CS delay	t_{CSD}	-	90	ns	
RD delay	t_{RSD}	-	90	ns	
Read data setup time	t_{RDS}	70	-	ns	
Read data hold time	t_{RDH}	0	-	ns	
WR delay	t_{WRD}	-	90	ns	
Write data delay	t_{WDD}	-	90	ns	
Write data hold time	t_{WDH}	0	-	ns	

Table 52.33 Bus timing (3) (2 of 2)

Conditions: Low drive output is selected in the Port Drive Capability bit in the PmnPFS register
VCC = AVCC0 = 1.8 to 2.4 V
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF

Parameter	Symbol	Min	Max	Unit	Test conditions
WAIT setup time	t_{WTS}	70	-	ns	Figure 52.43
WAIT hold time	t_{WTH}	0	-	ns	

Table 52.34 Bus timing (4)

Conditions: Low drive output is selected in the Port Drive Capability bit in the PmnPFS register
VCC = AVCC0 = 1.6 to 1.8 V
Output load conditions: V_{OH} = VCC × 0.5, V_{OL} = VCC × 0.5, C = 30 pF

Parameter	Symbol	Min	Max	Unit	Test conditions
Address delay	t_{AD}	-	120	ns	Figure 52.39 to Figure 52.42
Byte control delay	t_{BCD}	-	120	ns	
CS delay	t_{CSD}	-	120	ns	
RD delay	t_{RSD}	-	120	ns	
Read data setup time	t_{RDS}	90	-	ns	
Read data hold time	t_{RDH}	0	-	ns	
WR delay	t_{WRD}	-	120	ns	
Write data delay	t_{WDD}	-	120	ns	
Write data hold time	t_{WDH}	0	-	ns	
WAIT setup time	t_{WTS}	90	-	ns	Figure 52.43
WAIT hold time	t_{WTH}	0	-	ns	

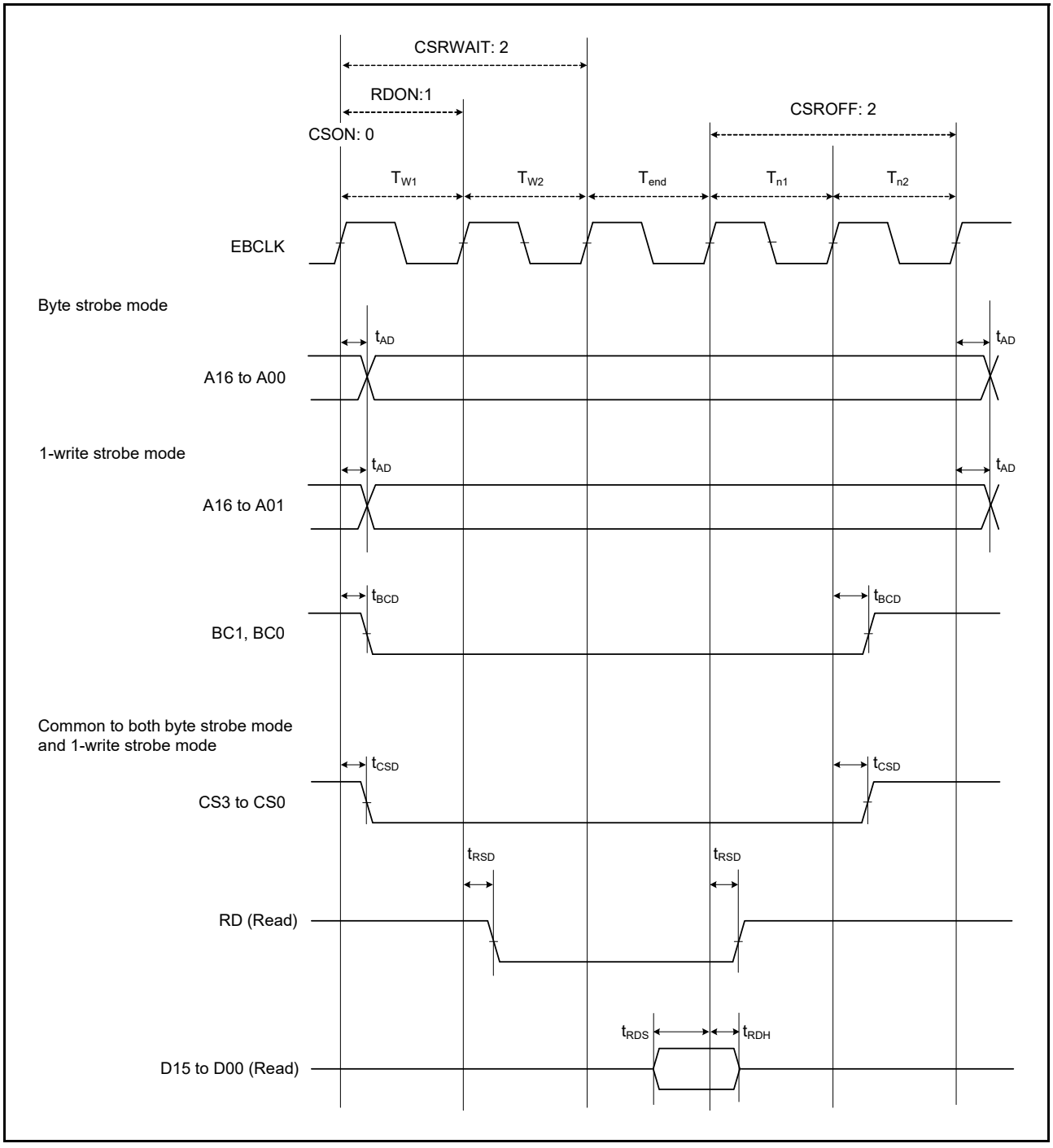
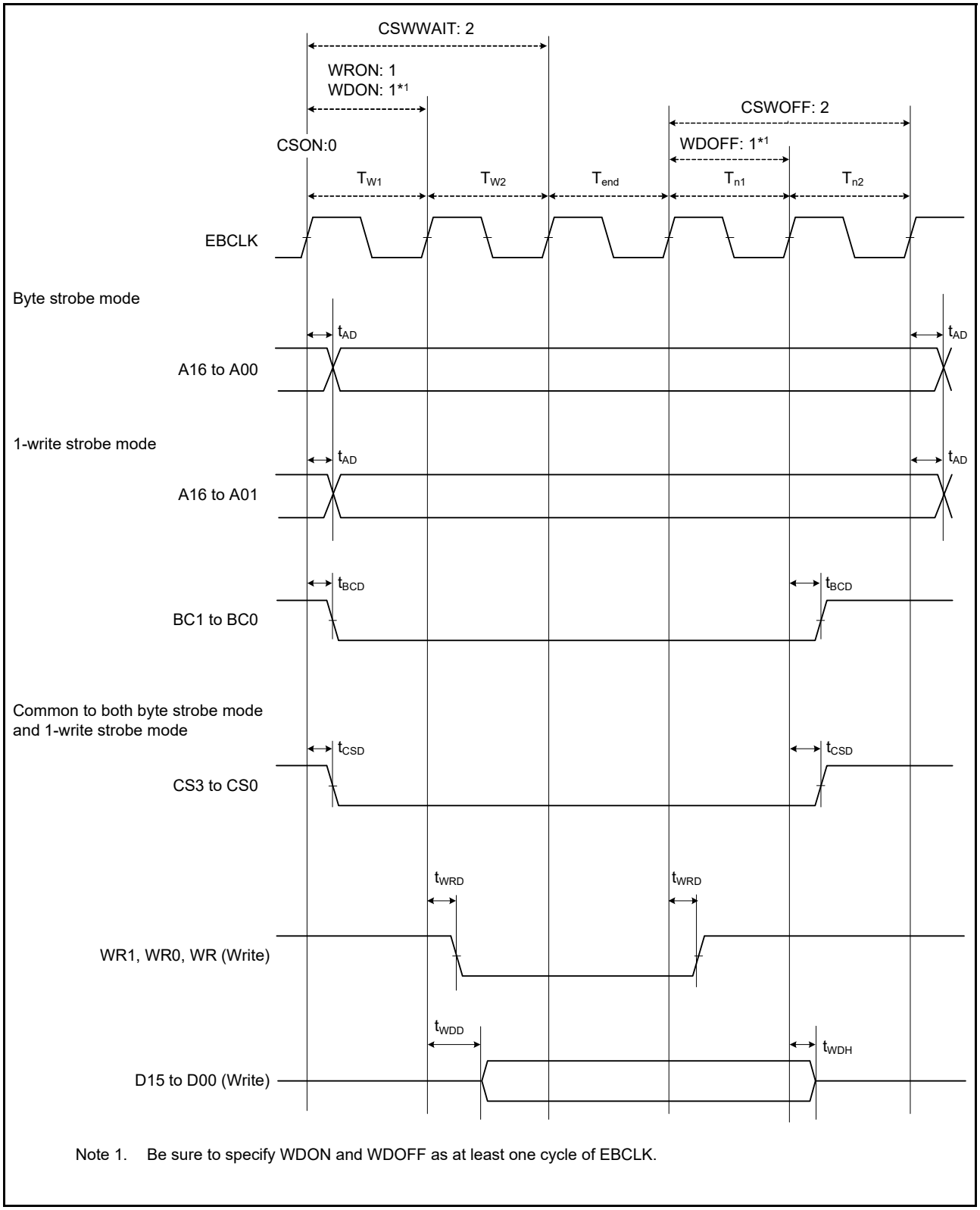


Figure 52.39 External bus timing/normal read cycle (bus clock synchronized)



Note 1. Be sure to specify WDON and WDOFF as at least one cycle of EBCLK.

Figure 52.40 External bus timing/normal write cycle (bus clock synchronized)

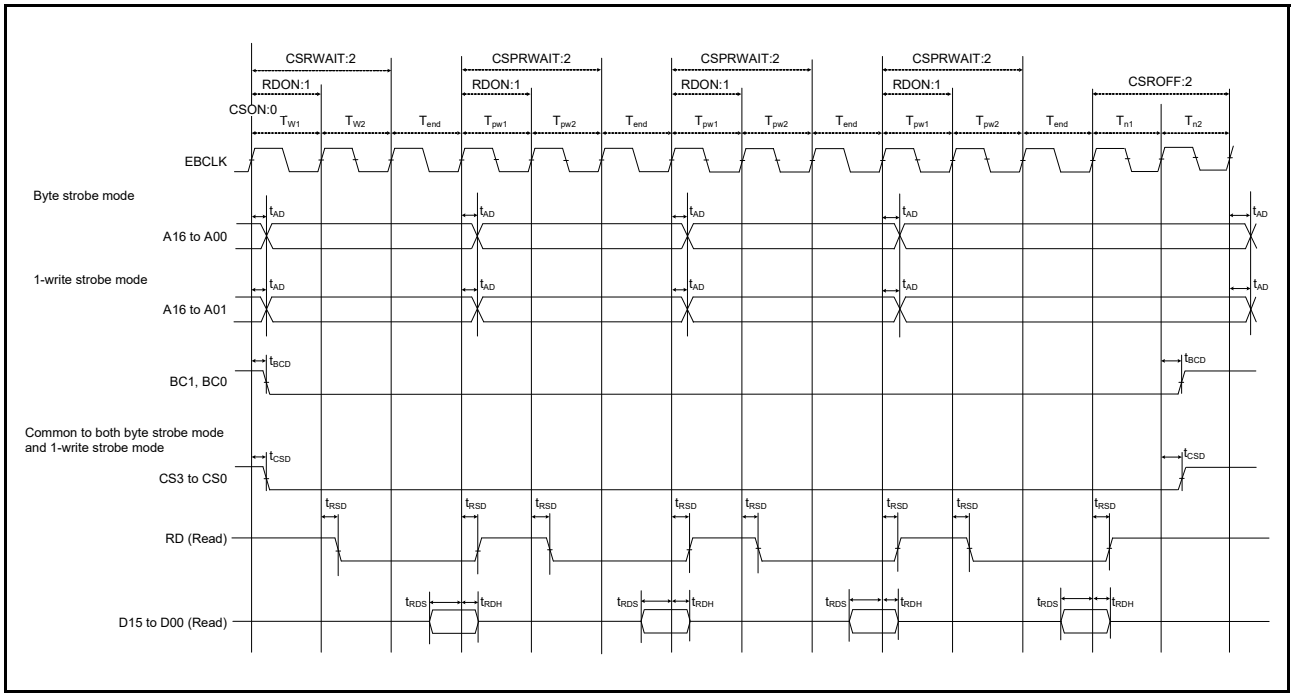
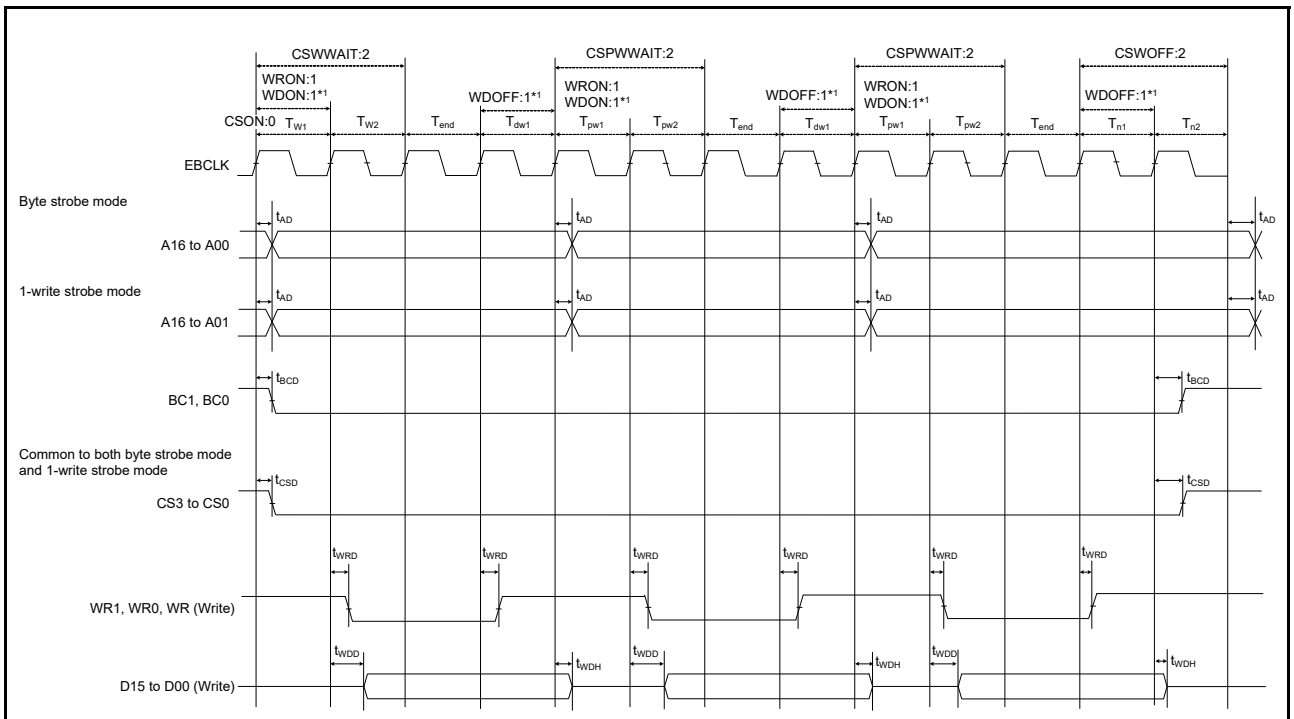


Figure 52.41 External bus timing/page read cycle (bus clock synchronized)



Note 1. Be sure to specify WDON and WDOFF as at least one cycle of EBCLK.

Figure 52.42 External bus timing/page write cycle (bus clock synchronized)

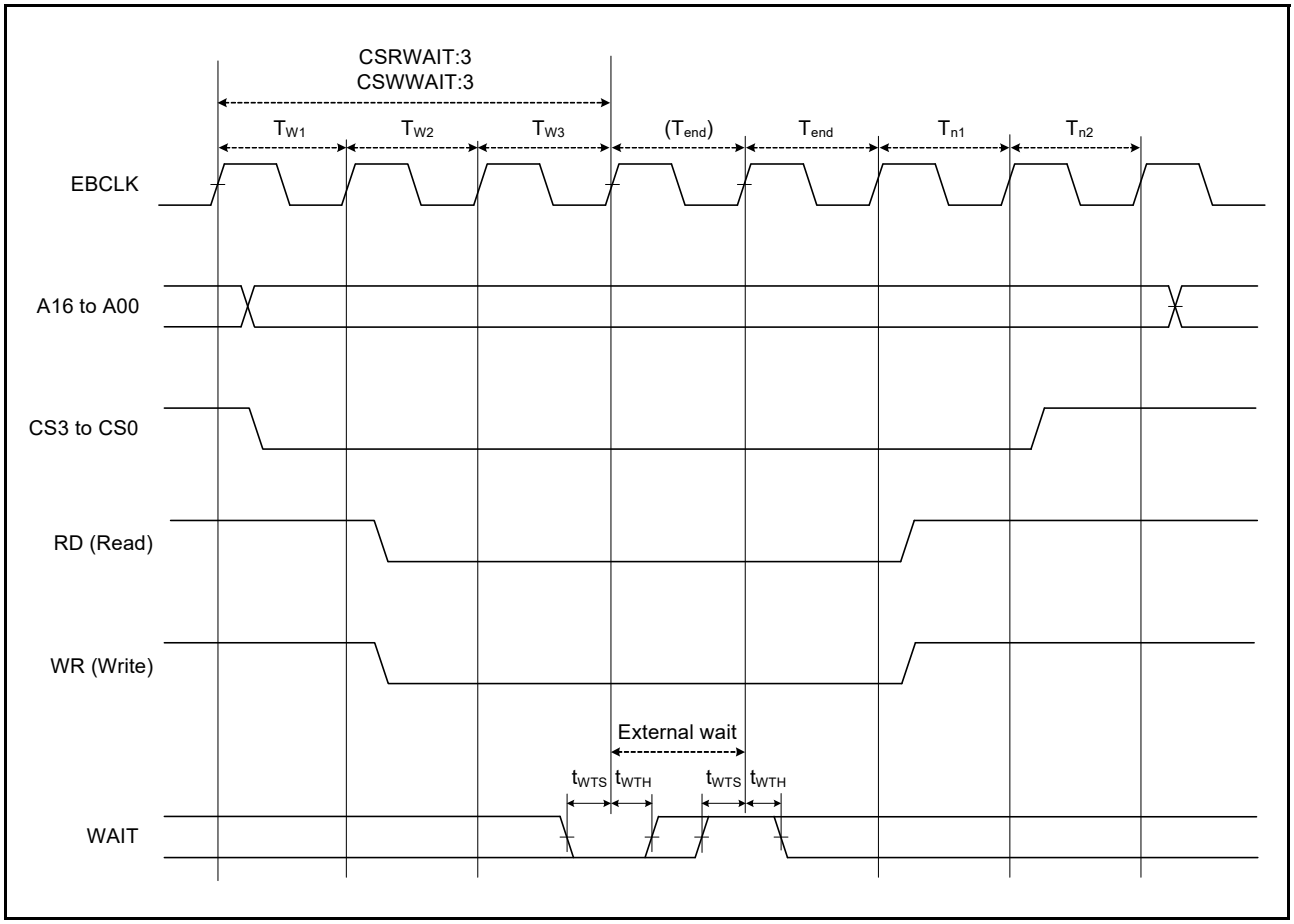


Figure 52.43 External bus timing/external wait control

52.3.7 I/O Ports, POEG, GPT, AGT, KINT, and ADC14 Trigger Timing

Table 52.35 I/O Ports, POEG, GPT, AGT, KINT, and ADC14 trigger timing

Parameter			Symbol	Min	Max	Unit	Test conditions
I/O Ports	Input data pulse width		t_{PRW}	1.5	-	t_{Pcyc}	Figure 52.44
	Input/Output data cycle (P002, P003, P004, P007)		t_{POcyc}	10	-	µs	
POEG	POEG input trigger pulse width		t_{POEW}	3	-	t_{Pcyc}	Figure 52.45
GPT	Input capture pulse width	Single edge	t_{GTICW}	1.5	-	t_{PDcyc}	Figure 52.46
		Dual edge		2.5	-		
AGT	AGTIO, AGTEE input cycle	2.7 V ≤ VCC ≤ 5.5 V	t_{ACYC}*1	250	-	ns	Figure 52.47
		2.4 V ≤ VCC < 2.7 V		500	-	ns	
		1.8 V ≤ VCC < 2.4 V		1000	-	ns	
		1.6 V ≤ VCC < 1.8 V		2000	-	ns	
	AGTIO, AGTEE input high level width, low-level width	2.7 V ≤ VCC ≤ 5.5 V	t_{ACKWH}, t_{ACKWL}	100	-	ns	
		2.4 V ≤ VCC < 2.7 V		200	-	ns	
		1.8 V ≤ VCC < 2.4 V		400	-	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 52.47
		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	
ADC14	14-bit A/D converter trigger input pulse width		t_{TRGW}	1.5	-	t_{Pcyc}	Figure 52.48
KINT	KRn (n = 00 to 07) pulse width		t_{KR}	250	-	ns	Figure 52.49

Note: tPcyc: PCLKB cycle, tPDcyc: PCLKD cycle
Note 1. Constraints on AGTIO input: t_{Pcyc} × 2 (t_{Pcyc}: PCLKB cycle) < t_{ACYC}

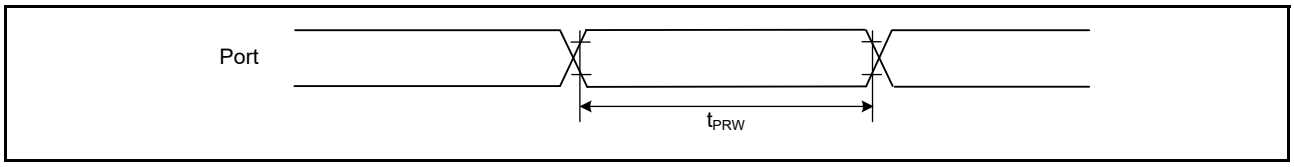


Figure 52.44 I/O ports input timing

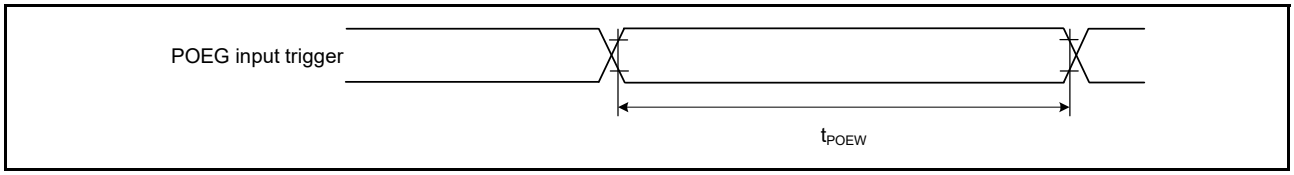


Figure 52.45 POEG input trigger timing

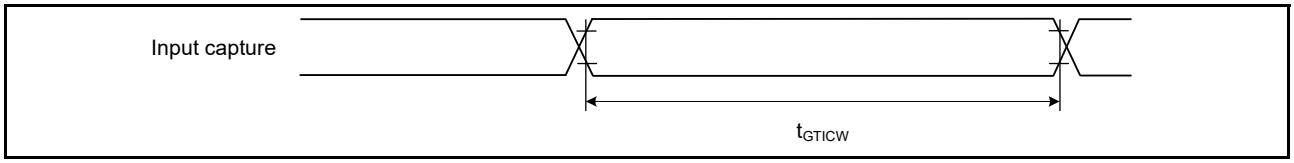


Figure 52.46 GPT input capture timing

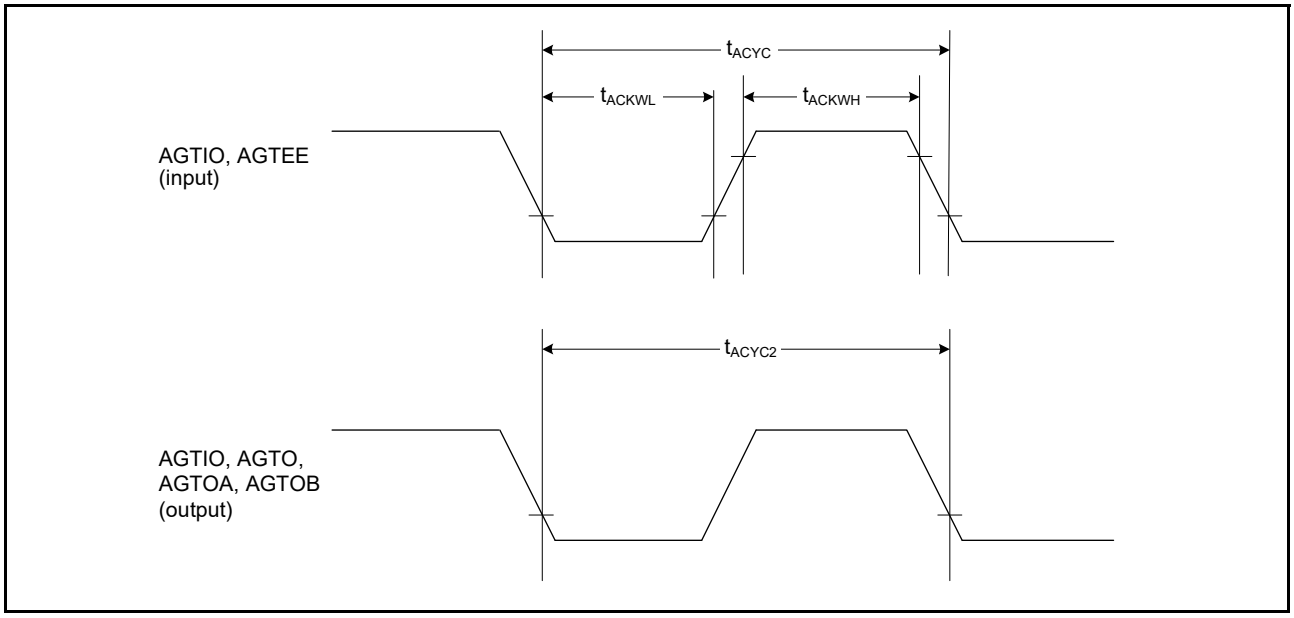


Figure 52.47 AGT I/O timing

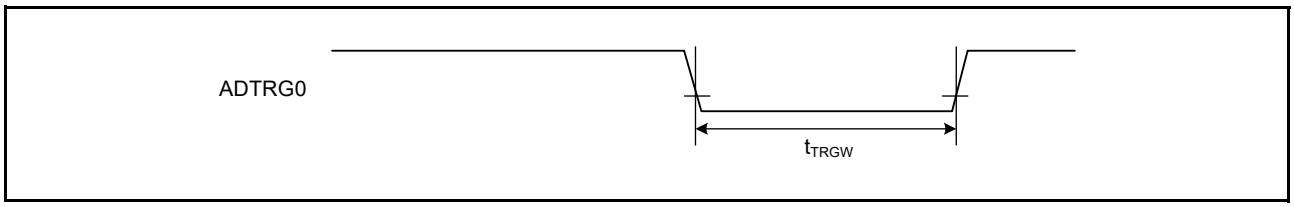


Figure 52.48 ADC14 trigger input timing

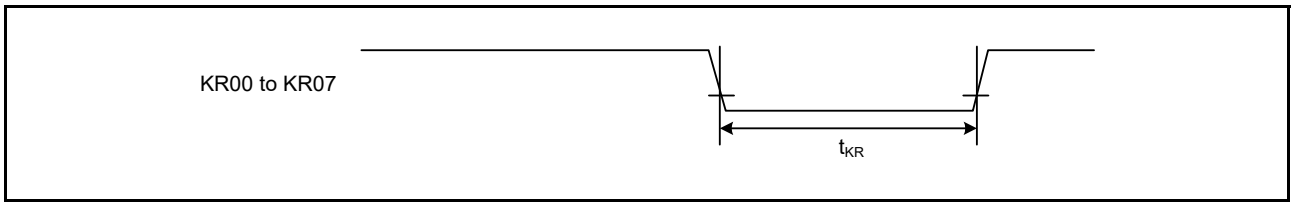


Figure 52.49 Key interrupt input timing

52.3.8 CAC Timing

Table 52.36 CAC timing

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
CAC	CACREF input pulse width	$t_{PBcyc} \leq t_{cac}$*2	t_{CACREF}	4.5 × t_{cac} + 3 × t_{PBcyc}	-	-	ns	-
		$t_{PBcyc} > t_{cac}$*2		5 × t_{cac} + 6.5 × t_{PBcyc}	-	-	ns	

Note 1. t_{PBcyc}: PCLKB cycle.
Note 2. t_{cac}: CAC count clock source cycle.

52.3.9 SCI Timing

Table 52.37 SCI timing (1)
Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter				Symbol	Min	Max	Unit*1	Test conditions
SCI	Input clock cycle	Asynchronous		t_{Scyc}	4	-	t_{Pcyc}	Figure 52.50
		Clock synchronous			6	-		
	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		t_{Scyc}	6	-	t_{Pcyc}	
		Clock synchronous			4	-		
	Output clock pulse width			t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time		1.8 V or above	t_{SCKr}	-	20	ns	
			1.6 V or above		-	30		
	Output clock fall time		1.8 V or above	t_{SCKf}	-	20	ns	
			1.6 V or above		-	30		
	Transmit data delay (master)	Clock synchronous	1.8 V or above	t_{TXD}	-	40	ns	Figure 52.51
			1.6 V or above		-	45		
	Transmit data delay (slave)	Clock synchronous	2.7 V or above		-	55	ns	
			2.4 V or above		-	60		
			1.8 V or above		-	100		
			1.6 V or above		-	125		
	Receive data setup time (master)	Clock synchronous	2.7 V or above	t_{RXS}	45	-	ns	
			2.4 V or above		55	-		
			1.8 V or above		90	-		
			1.6 V or above		105	-		
	Receive data setup time (slave)	Clock synchronous	2.7 V or above		40	-	ns	
			1.6 V or above		45	-		
	Receive data hold time (master)	Clock synchronous		t_{RXH}	5	-	ns	
	Receive data hold time (slave)	Clock synchronous		t_{RXH}	40	-	ns	

Note 1. t_{Pcyc}: PCLKA cycle.

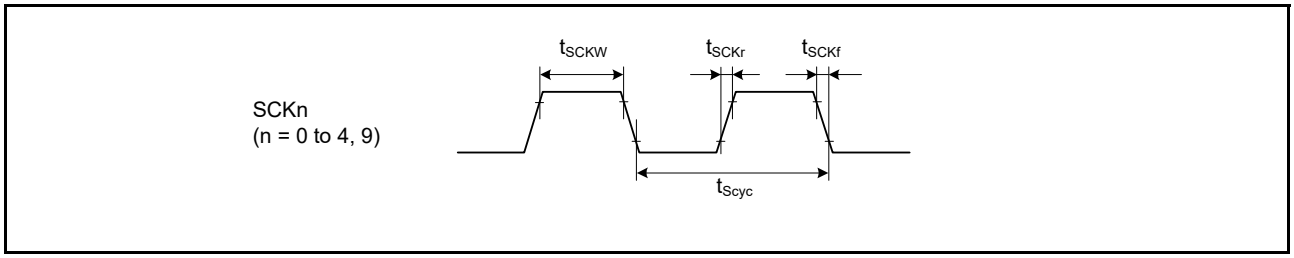


Figure 52.50 SCK clock input timing

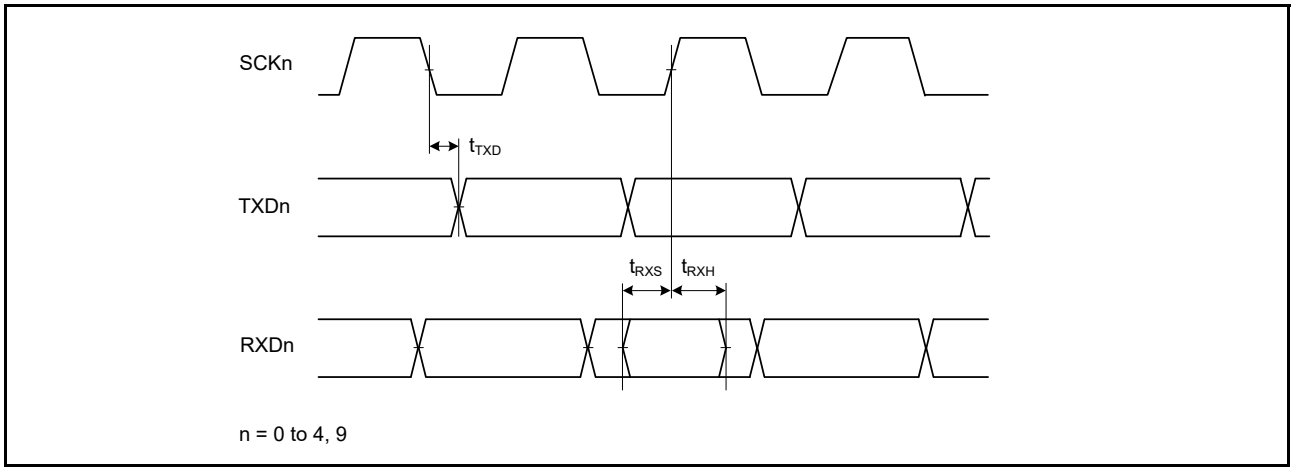


Figure 52.51 SCI input/output timing in clock synchronous mode

Table 52.38 SCI timing (2)
Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter				Symbol	Min	Max	Unit	Test conditions
Simple SPI	SCK clock cycle output (master)			t_{SPcyc}	4	65536	t_{Pcyc}	Figure 52.52
	SCK clock cycle input (slave)				6	65536		
	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 or above	t_{SPCKr}, t_{SPCKf}	-	20	ns	
			1.6 V or above		-	30		
	Data input setup time	Master	2.7 V or above	t_{SU}	45	-	ns	Figure 52.53 to Figure 52.56
			2.4 V or above		55	-		
			1.8 V or above		80	-		
			1.6 V or above		105	-		
		Slave	2.7 V or above		40	-		
			1.6 V or above		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	Master	1.8 V or above	t_{OD}	-	40	ns	
			1.6 V or above		-	50		
		Slave	2.4 V or above		-	65		
			1.8 V or above		-	100		
			1.6 V or above		-	125		
	Data output hold time	Master	2.7 V or above	t_{OH}	–10	-	ns	
			2.4 V or above		–20	-		
			1.8 V or above		–30	-		
			1.6 V or above		–40	-		
		Slave			–10	-		
	Data rise and fall time	Master	1.8 V or above	t_{Dr}, t_{Df}	-	20	ns	
			1.6 V or above		-	30		
		Slave	1.8 V or above		-	20		
			1.6 V or above		-	30		
	Slave access time			t_{SA}	-	10 (PCLKA > 32 MHz), 6 (PCLKA ≤ 32 MHz)	t_{Pcyc}	Figure 52.55 and Figure 52.56 PCLKB = PCLKA
	Slave output release time			t_{REL}	-	10 (PCLKA > 32 MHz), 6 (PCLKA ≤ 32 MHz)	t_{Pcyc}	

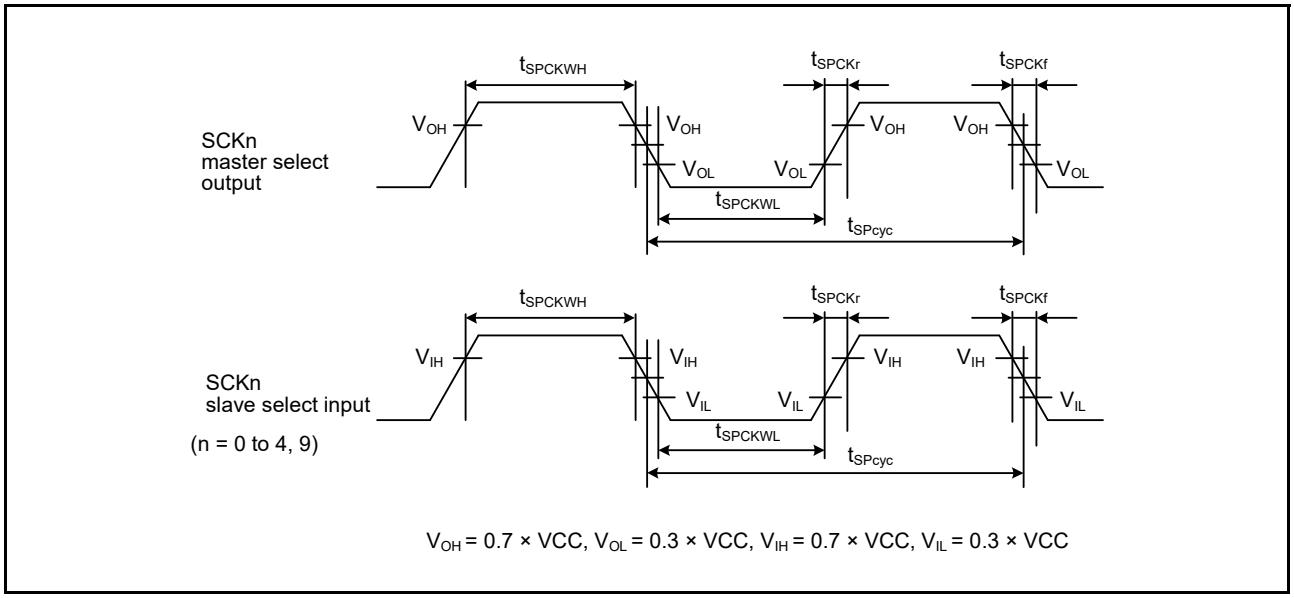


Figure 52.52 SCI simple SPI mode clock timing

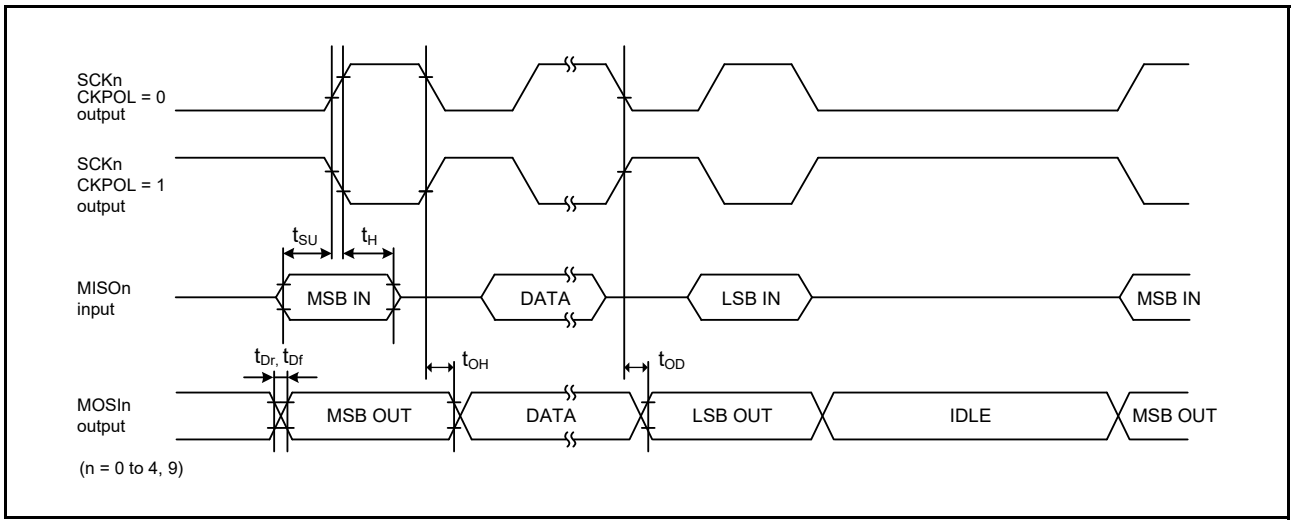


Figure 52.53 SCI simple SPI mode timing (master, CKPH = 1)

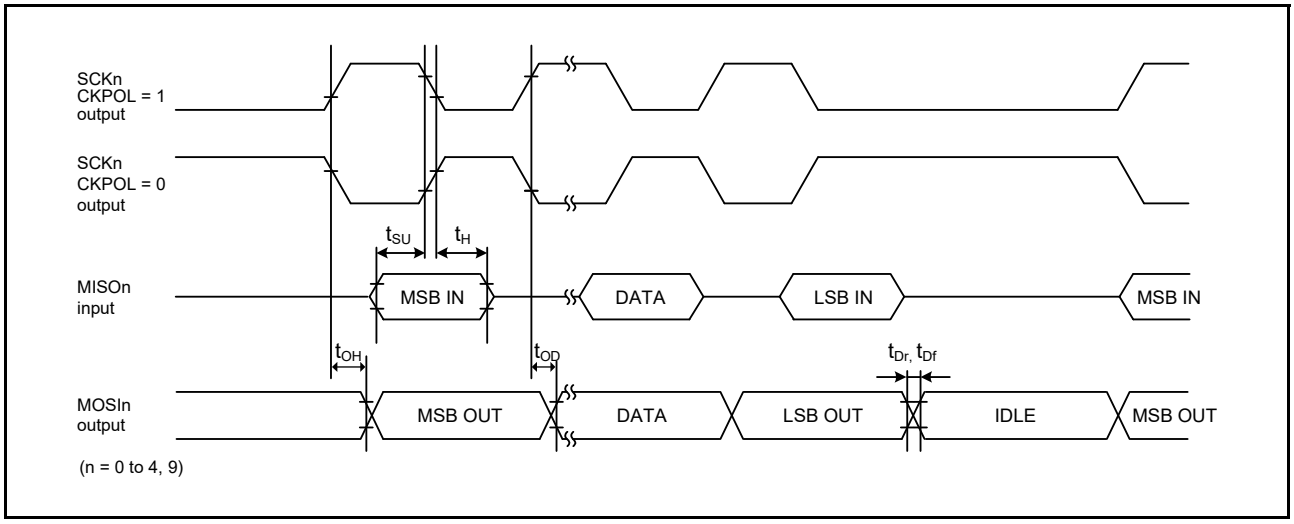


Figure 52.54 SCI simple SPI mode timing (master, CKPH = 0)

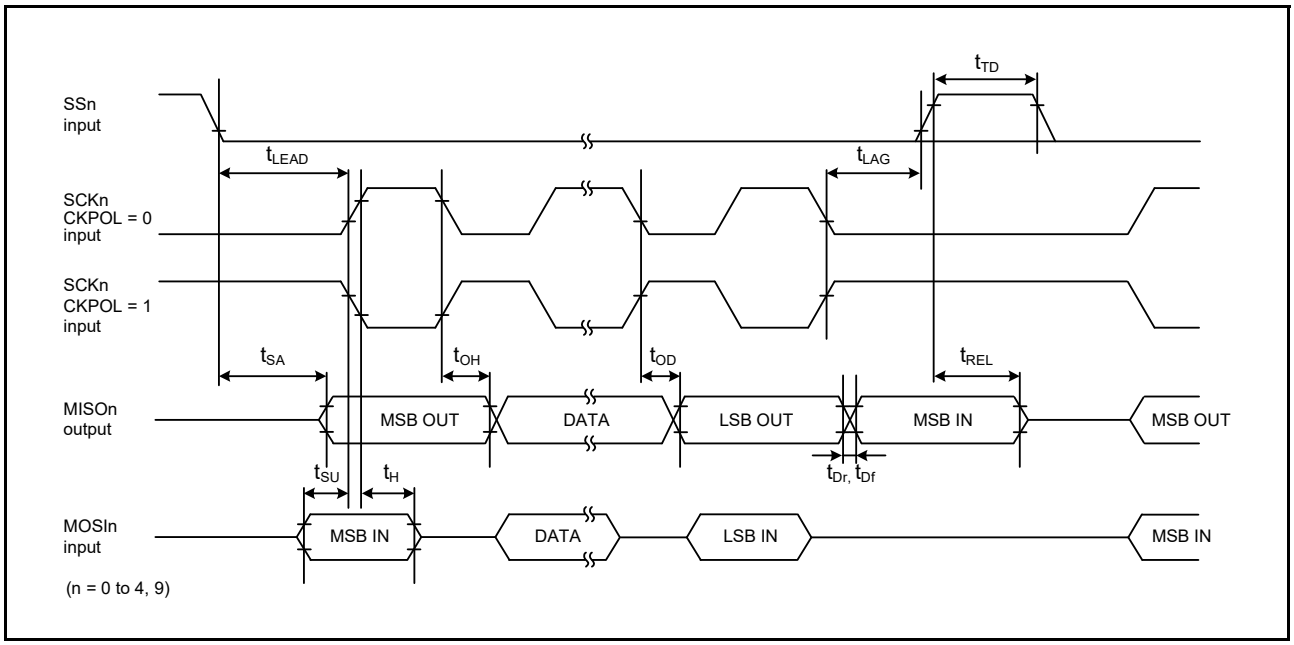


Figure 52.55 SCI simple SPI mode timing (slave, CKPH = 1)

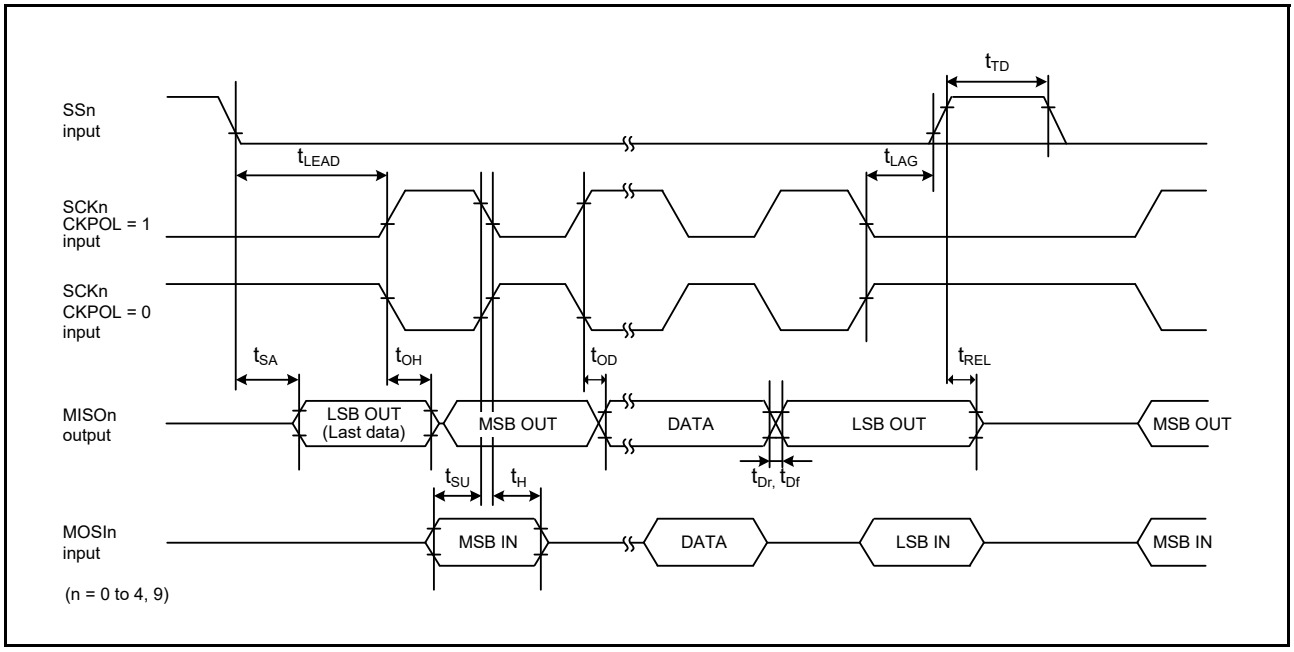


Figure 52.56 SCI simple SPI mode timing (slave, CKPH = 0)

Table 52.39 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 52.57
	SDA input fall time	t_{Sf}	-	300	ns	
	SDA input spike pulse removal time	t_{SP}	0	4 × t_{IICcyc}	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)*3	SDA input rise time	t_{Sr}	-	300	ns	Figure 52.57
	SDA input fall time	t_{Sf}	-	300	ns	
	SDA input spike pulse removal time	t_{SP}	0	4 × t_{IICcyc}	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 in the SMR.CKS[1:0] bits.
Note 2. Cb indicates the total capacity of the bus line.
Note 3. Middle drive output is selected in the Port Drive Capability in the PmnPFS register

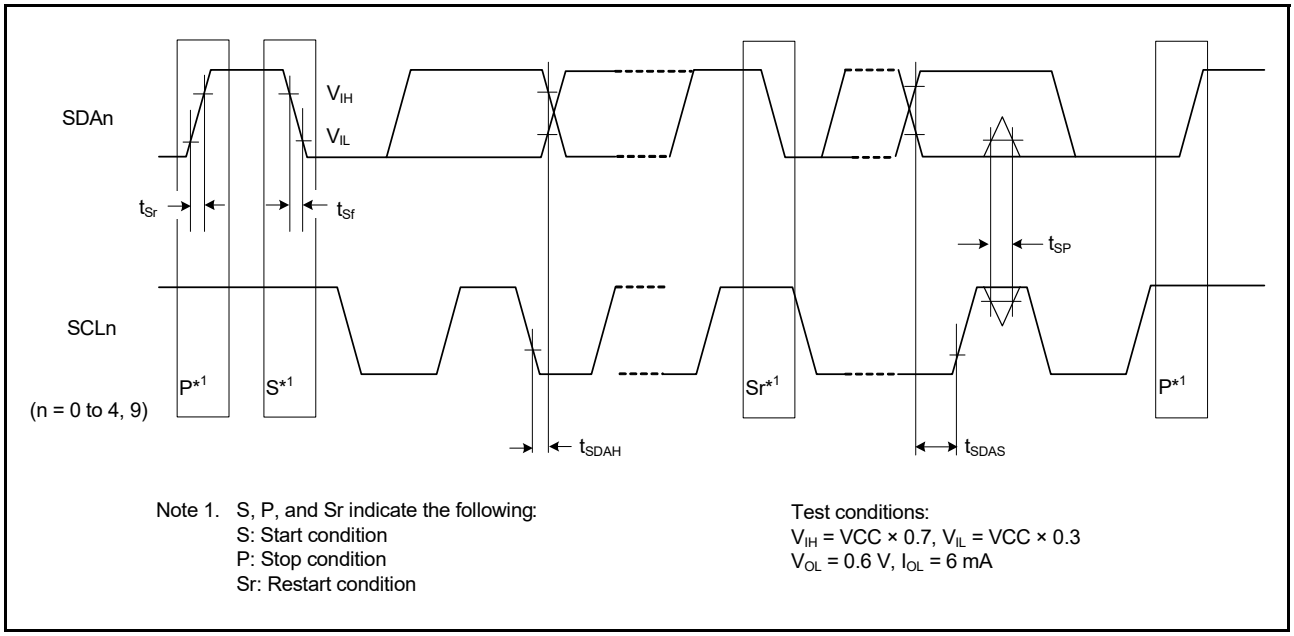


Figure 52.57 SCI simple IIC mode timing

52.3.10 SPI Timing

Table 52.40 SPI timing (1 of 2)
Conditions: Middle drive output is selected in the Port Drive Capability in the PmnPFS register

Parameter				Symbol	Min	Max	Unit*1	Test conditions
SPI	RSPCK clock cycle	Master		t_{SPcyc}	2*4	4096	t_{Pcyc}	Figure 52.58 C = 30pF
		Slave			6	4096		
	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 or above	t_{SPCKr}, t_{SPCKf}	-	10	ns	
			2.4 V or above		-	15		
			1.8 V or above		-	20		
			1.6 V or above		-	30		
		Input			-	1	µs	
	Data input setup time	Master		t_{SU}	10	-	ns	Figure 52.59 to Figure 52.64 C = 30pF
		Slave	2.4 V or above		10	-		
			1.8 V or above		15	-		
			1.6 V or above		20	-		
	Data input hold time	Master (RSPCK is PCLKA/2)		t_{HF}	0	-	ns	
		Master (RSPCK is other than above.)		t_H	t_{Pcyc}	-		
		Slave		t_H	20	-		
	SSL setup time	Master		t_{LEAD}	$-30 + N \times t_{Spcyc}$*2	-	ns	
		Slave			$6 \times t_{Pcyc}$	-	ns	
	SSL hold time	Master		t_{LAG}	$-30 + N \times t_{Spcyc}$*3	-	ns	
		Slave			$6 \times t_{Pcyc}$	-	ns	

Table 52.40 SPI timing (2 of 2)

Conditions: Middle drive output is selected in the Port Drive Capability in the PmnPFS register

Parameter				Symbol	Min	Max	Unit*1	Test conditions
SPI	Data output delay	Master	2.7 V or above	t_{OD}	-	14	ns	Figure 52.59 to Figure 52.64 C = 30pF
			2.4 V or above		-	20		
			1.8 V or above		-	25		
			1.6 V or above		-	30		
		Slave	2.7 V or above		-	50		
			2.4 V or above		-	60		
			1.8 V or above		-	85		
			1.6 V or above		-	110		
	Data output hold time	Master		t_{OH}	0	-	ns	
		Slave			0	-		
	Successive transmission delay	Master		t_{TD}	$t_{SPcyc} + 2 \times t_{Pcyc}$	$8 \times t_{SPcyc} + 2 \times t_{Pcyc}$	ns	
		Slave			$6 \times t_{Pcyc}$	-		
	MOSI and MISO rise and fall time	Output	2.7 V or above	t_{Dr}, t_{Df}	-	10	ns	
			2.4 V or above		-	15		
			1.8 V or above		-	20		
			1.6 V or above		-	30		
		Input			-	1	µs	
	SSL rise and fall time	Output	2.7 V or above	t_{SSLr}, t_{SSLf}	-	10	ns	
			2.4 V or above		-	15		
			1.8 V or above		-	20		
			1.6 V or above		-	30		
		Input			-	1	µs	
	Slave access time		2.4 V or above	t_{SA}	-	$2 \times t_{Pcyc} + 100$	ns	Figure 52.63 and Figure 52.64 C = 30pF
			1.8 V or above		-	$2 \times t_{Pcyc} + 140$		
			1.6 V or above		-	$2 \times t_{Pcyc} + 180$		
	Slave output release time		2.4 V or above	t_{REL}	-	$2 \times t_{Pcyc} + 100$	ns	
			1.8 V or above		-	$2 \times t_{Pcyc} + 140$		
			1.6 V or above		-	$2 \times t_{Pcyc} + 180$		

Note 1. t_{Pcyc}: PCLKA 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.
Note 4. The upper limit of RSPCK is 16 MHz.

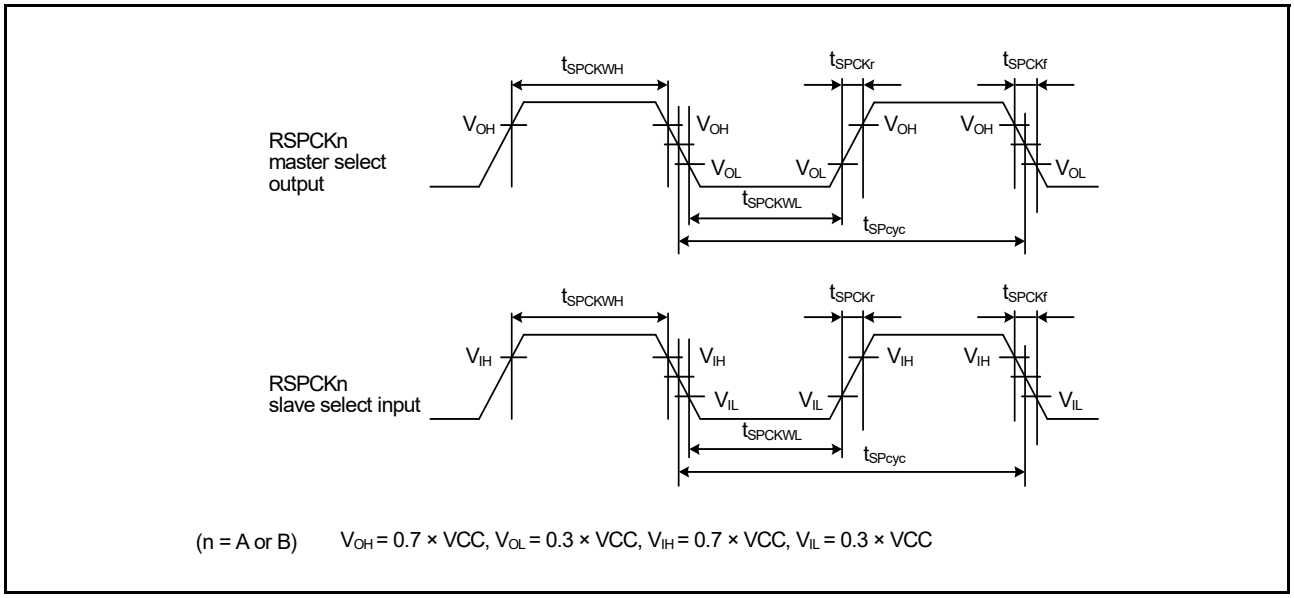


Figure 52.58 SPI clock timing

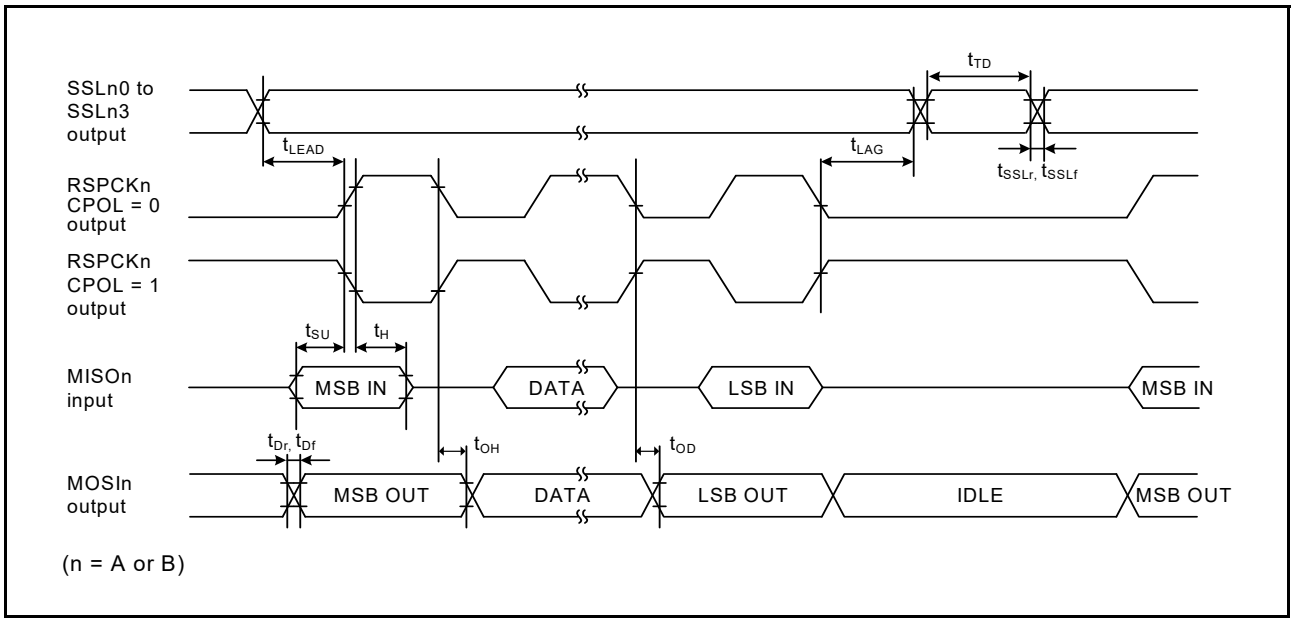


Figure 52.59 SPI timing (master, CPHA = 0) (bit rate: PCLKA division ratio is set to any value other than 1/2)

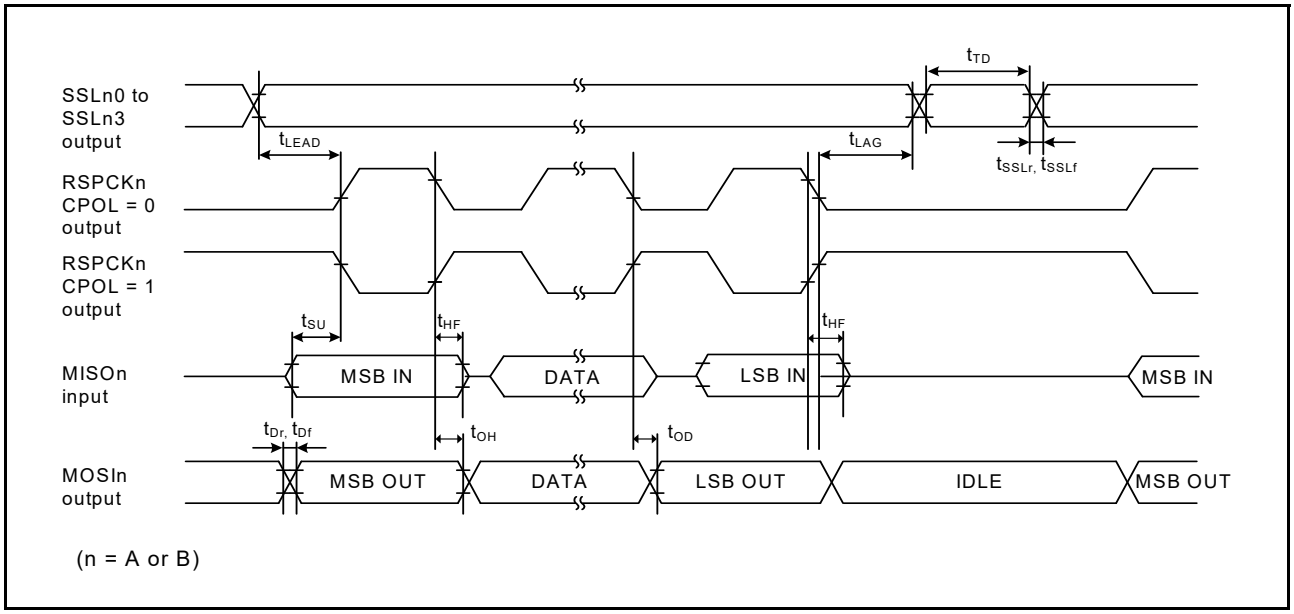


Figure 52.60 SPI timing (master, CPHA = 0) (bit rate: PCLKA division ratio is set to 1/2)

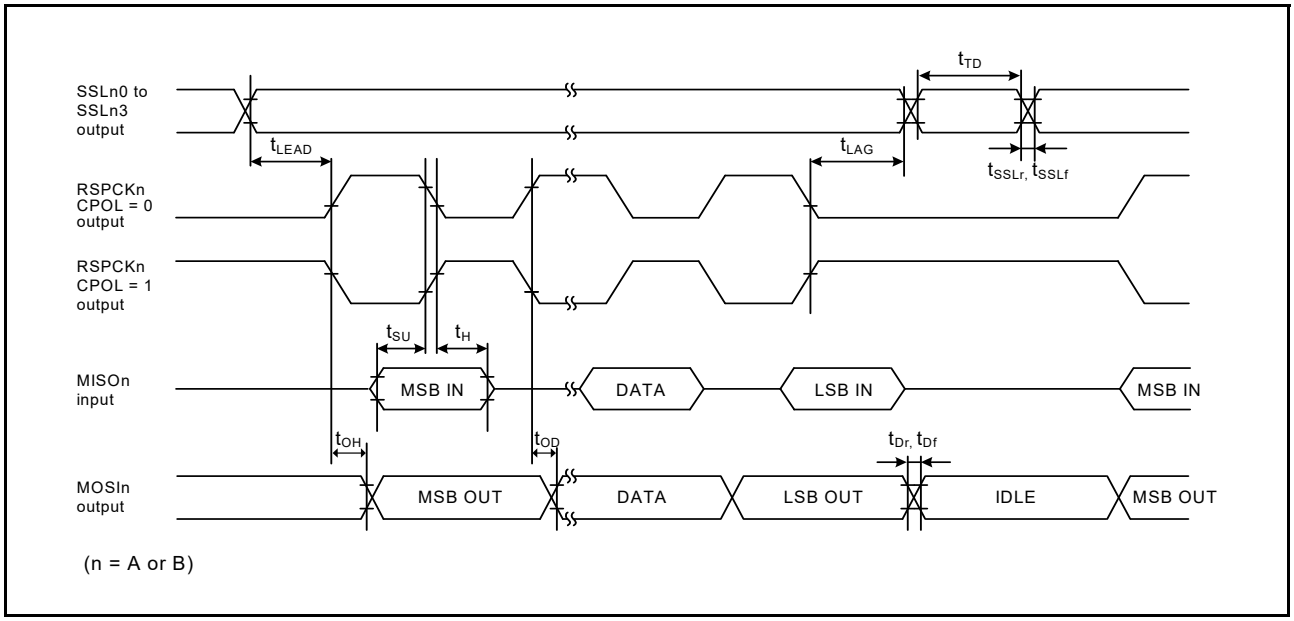


Figure 52.61 SPI timing (master, CPHA = 1) (bit rate: PCLKA division ratio is set to any value other than 1/2)

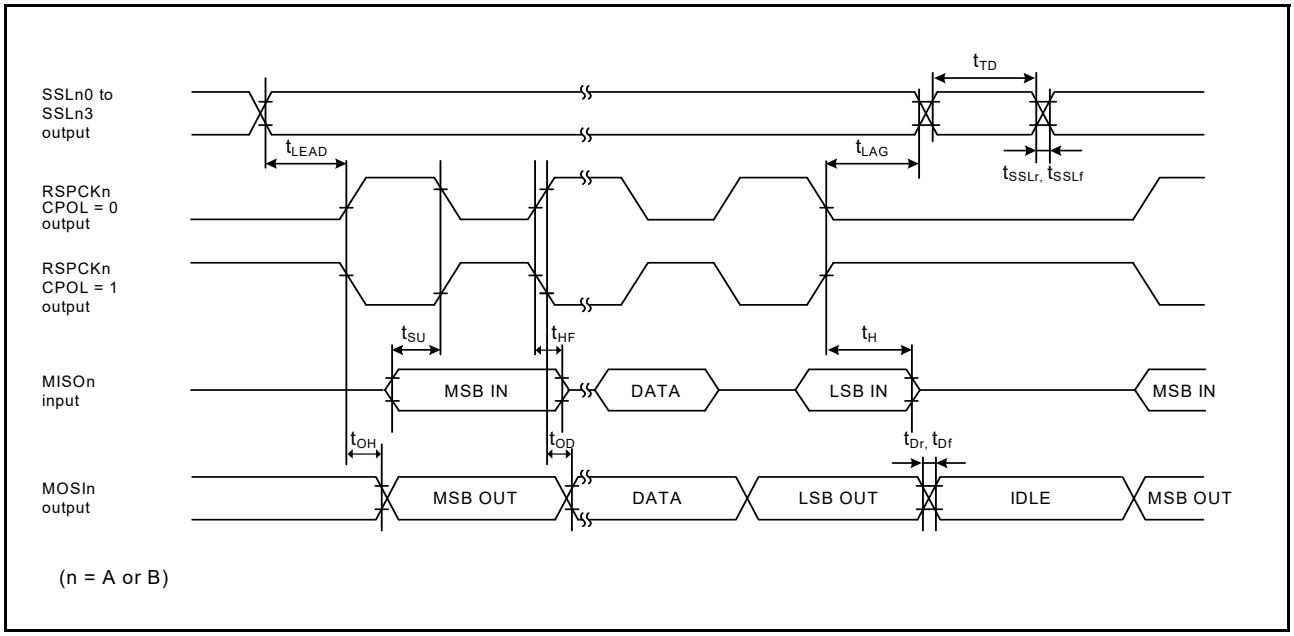


Figure 52.62 SPI timing (master, CPHA = 1) (bit rate: PCLKA division ratio is set to 1/2)

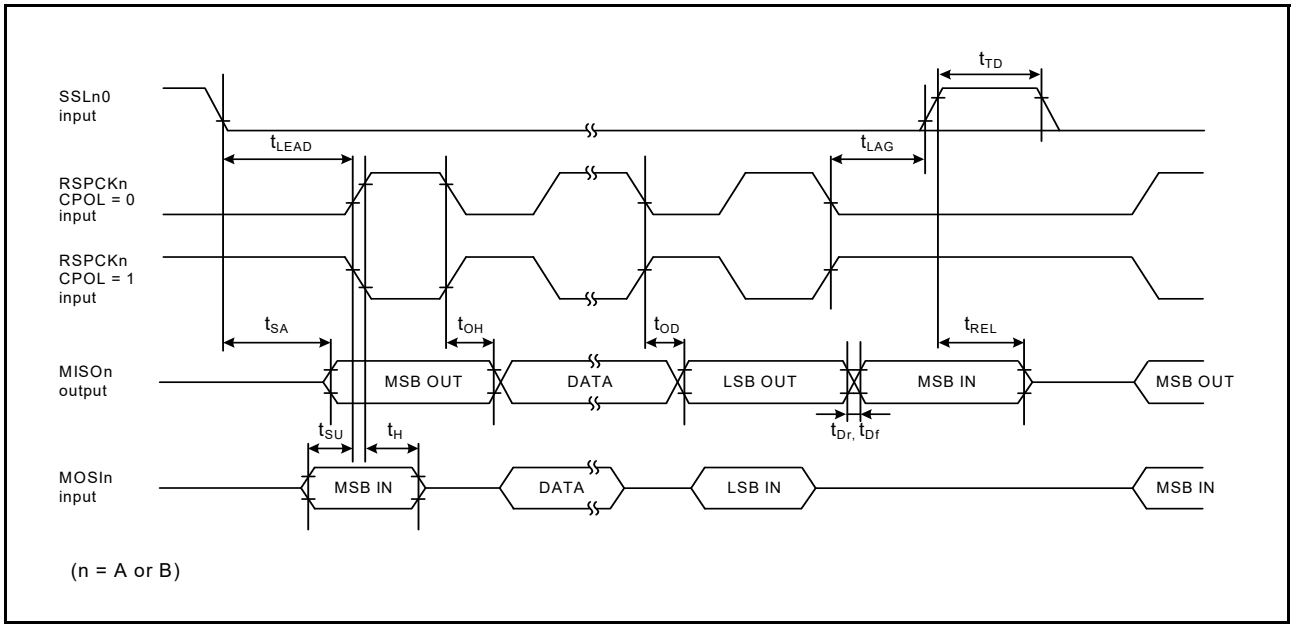


Figure 52.63 SPI timing (slave, CPHA = 0)

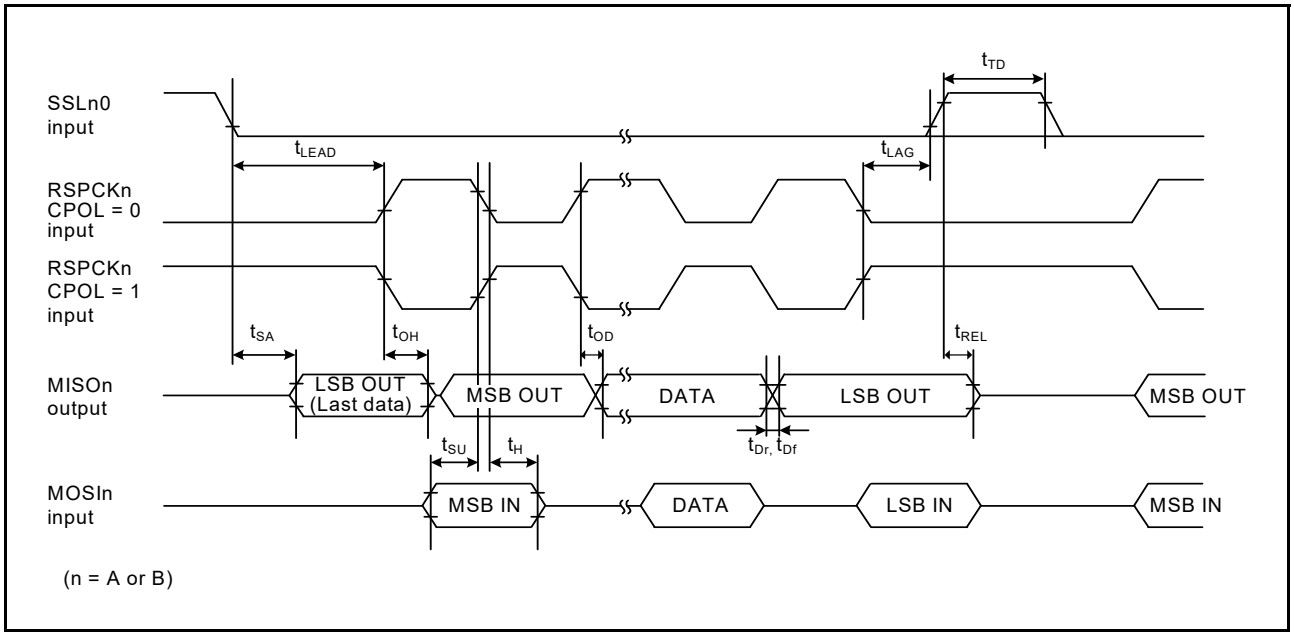


Figure 52.64 SPI timing (slave, CPHA = 1)

52.3.11 QSPI Timing

Table 52.41 QSPI timing

Conditions: VCC = AVCC0 = 1.8 to 5.5 V

Conditions: Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register

Parameter			Symbol	Min	Max	Unit*1	Test conditions
QSPI	QSPCLK clock cycle		t_{QScyc}	2*4	48	t_{Pcyc}	Figure 52.65
	QSPCLK clock high-level pulse width		t_{QSWH}	t_{QScyc} × 0.4	-	ns	
	QSPCLK clock low-level pulse width		t_{QSWL}	t_{QScyc} × 0.4	-	ns	
	Data input setup time	2.7 V or above	t_{SU}	40	-	ns	Figure 52.66
		2.4 V or above		40	-	ns	
		1.8 V or above		80	-	ns	
	Data input hold time		t_{IH}	0	-	ns	
	SSL setup time		t_{LEAD}	(N + 0.5) × t_{Qscyc} - 15*2	(N + 0.5) × t_{Qscyc} + 100*2	ns	
	SSL hold time		t_{LAG}	(N + 0.5) × t_{Qscyc} - 15*3	(N + 0.5) × t_{Qscyc} + 100*3	ns	
	Data output delay	2.7 V or above	t_{OD}	-	14	ns	
		2.4 V or above		-	20		
		1.8 V or above		-	30		
	Data output hold time	2.7 V or above	t_{OH}	–3.3	-	ns	
		1.8 V or above		–10	-		
	Successive transmission delay		t_{TD}	1	16	t_{Qscyc}	

Note 1. t_{Pcyc}: PCLKA cycle.
Note 2. N is set to 0 or 1 in SFMSLD.
Note 3. N is set to 0 or 1 in SFMSHD.
Note 4. The upper limit of QSPCLK is 16 MHz.

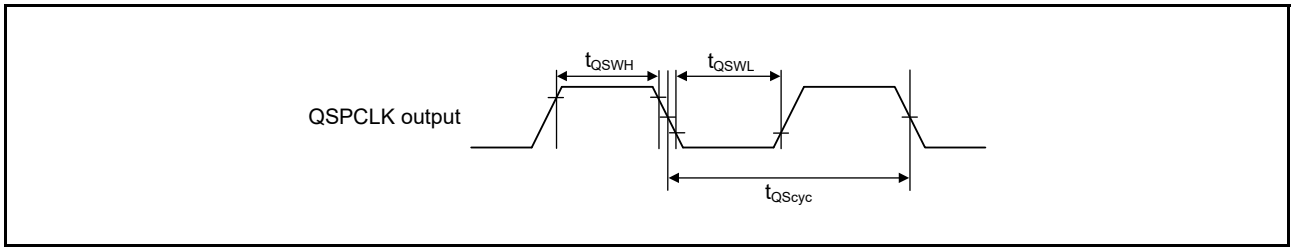


Figure 52.65 QSPI clock timing

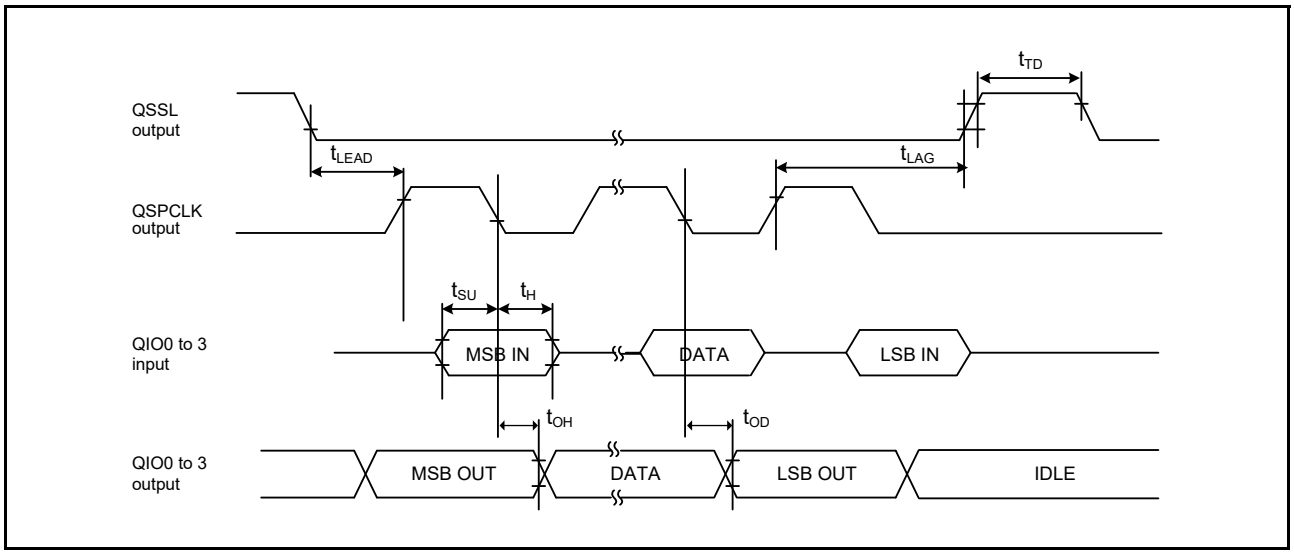


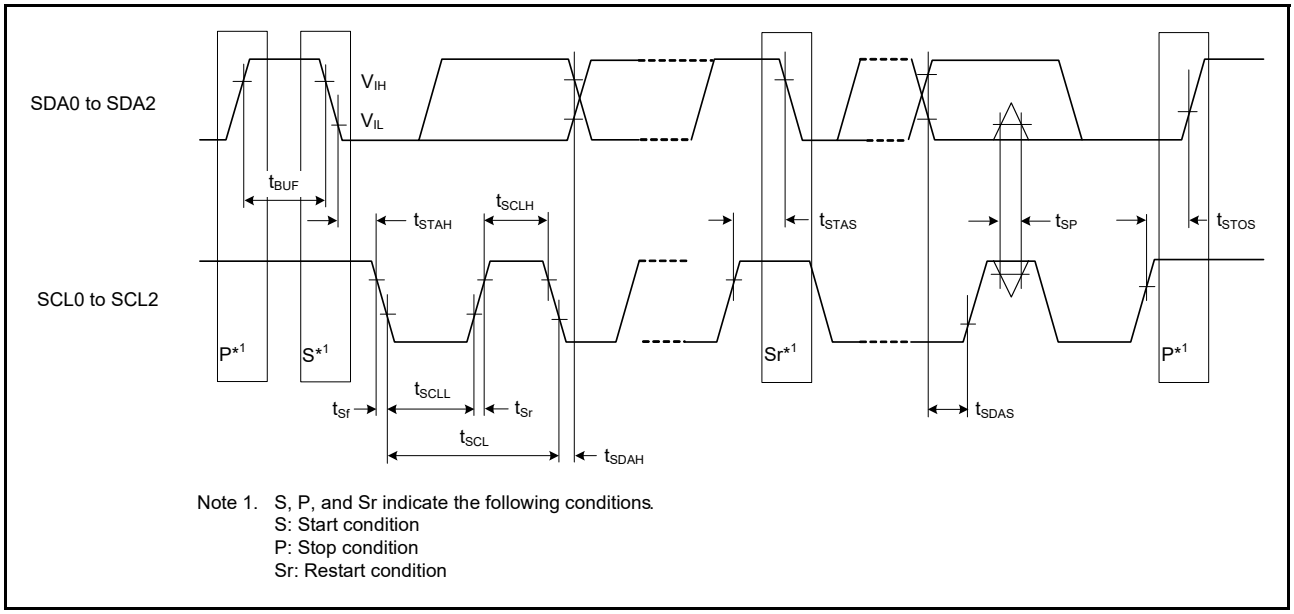
Figure 52.66 Transfer/receive timing

52.3.12 IIC Timing

Table 52.42 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 52.67
	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)*2	SCL input cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 600	-	ns	Figure 52.67
	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. The value 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 2. Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.



Note 1. S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 52.67 I²C bus interface input/output timing

52.3.13 SSI Timing

Table 52.43 SSI timing
Conditions: VCC = AVCC0 = 1.6 to 5.5 V

Parameter			Symbol	Min	Max	Unit	Test conditions
SSI	AUDIO_CLK input frequency	2.7 V or above	t_{AUDIO}	-	25	MHz	-
		1.6 V or above		-	4		
	Output clock period		t_O	250	-	ns	Figure 52.68
	Input clock period		t_I	250	-	ns	
	Clock high pulse width	1.8 V or above	t_{HC}	100	-	ns	
		1.6 V or above		200	-		
	Clock low pulse width	1.8 V or above	t_{LC}	100	-	ns	
		1.6 V or above		200	-		
	Clock rise time		t_{RC}	-	25	ns	
	Data delay	2.7 V or above	t_{DTR}	-	65	ns	Figure 52.69, Figure 52.70
		1.8 V or above		-	105		
		1.6 V or above		-	140		
	Set-up time	2.7 V or above	t_{SR}	65	-	ns	
		1.8 V or above		90	-		
		1.6 V or above		140	-		
	Hold time		t_{HTR}	40	-	ns	
	SSIDATA output delay from WS change time	1.8 V or above	T_{DTRW}	-	105	ns	Figure 52.71
		1.6 V or above		-	140		

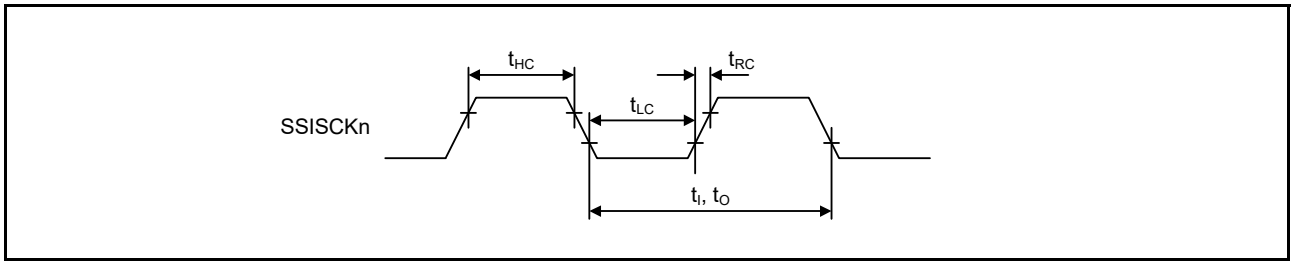


Figure 52.68 SSI clock input/output timing

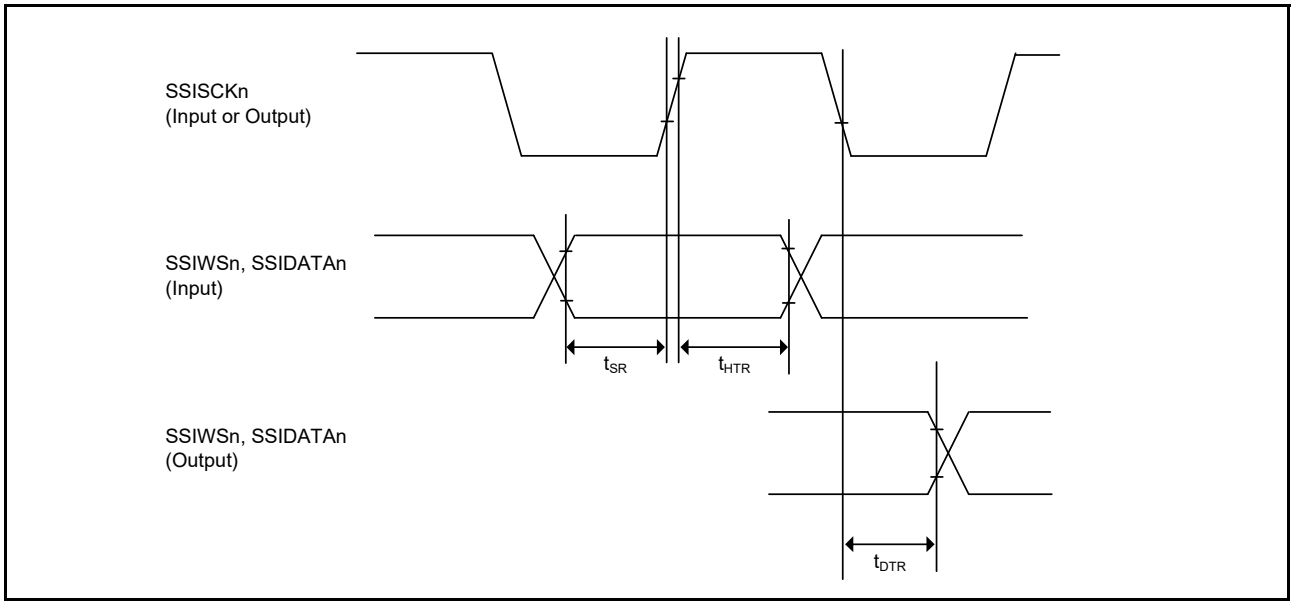


Figure 52.69 SSI data transmit/receive timing (SSICR.SCKP = 0)

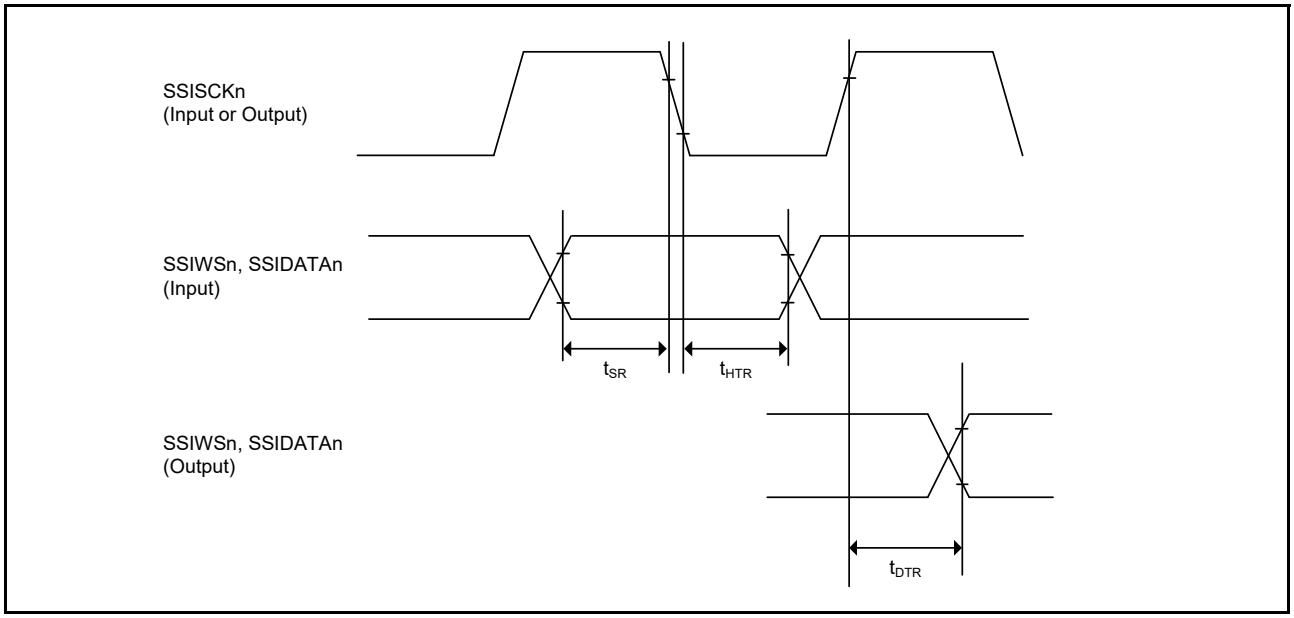


Figure 52.70 SSI data transmit/receive timing (SSICR.SCKP = 1)

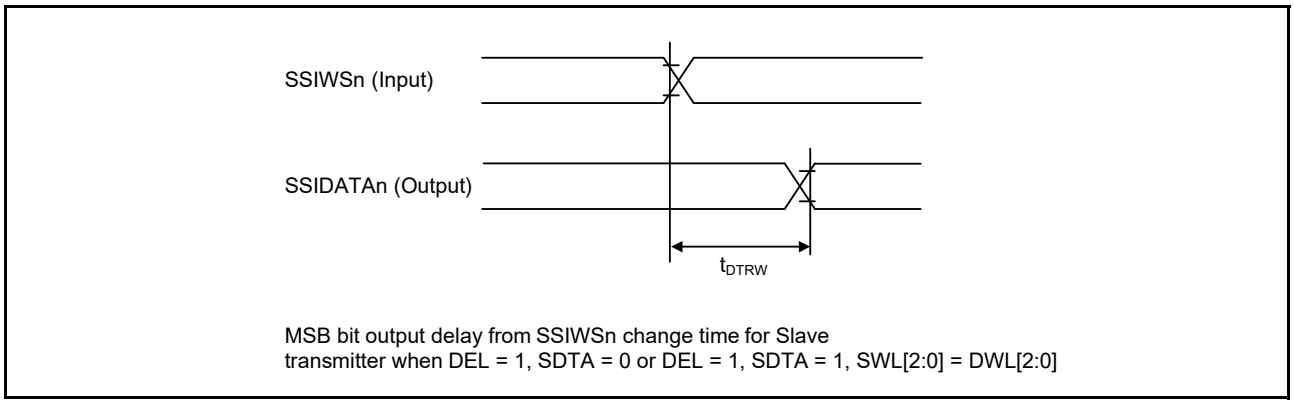


Figure 52.71 SSI data output delay from SSIWSn change time

52.3.14 SD/MMC Host Interface Timing

Table 52.44 SD/MMC host interface signal timing

Conditions: VCC = AVCC0 = 2.7 to 5.5 V

Conditions: Middle drive output is selected in the Port Drive Capability in PmnPFS register

Parameter	Symbol	Min	Max	Unit	Test conditions
SDCLK clock cycle	t_{SDCYC}	62.5	-	ns	Figure 52.72
SDCLK clock high-level pulse width	t_{SDWH}	18.25	-	ns	
SDCLK clock low-level pulse width	t_{SDWL}	18.25	-	ns	
SDCLK clock rising time	t_{SDLH}	-	10	ns	
SDCLK clock falling time	t_{SDHL}	-	10	ns	
SDCMD/SDDAT output data delay	t_{SDODLY}	–18.25	18.25	ns	
SDCMD/SDDAT input data setup	t_{SDIS}	9.25	-	ns	
SDCMD/SDDAT input data hold	t_{SDIH}	23.25	-	ns	

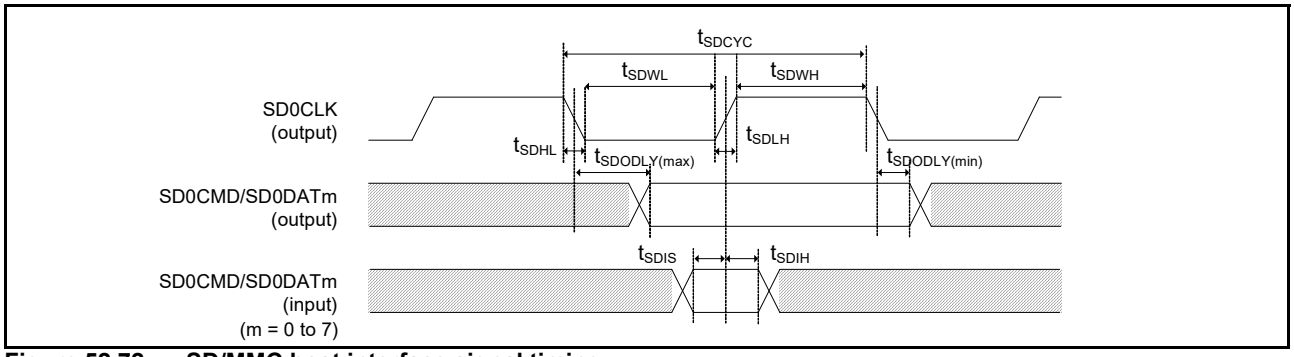


Figure 52.72 SD/MMC host interface signal timing

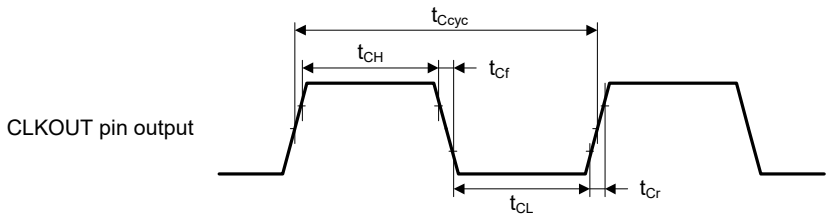
52.3.15 CLKOUT Timing

Table 52.45 CLKOUT timing

Parameter			Symbol	Min	Max	Unit*1	Test conditions
CLKOUT	CLKOUT pin output cycle*1	VCC = 2.7 V or above	t_{Ccyc}	62.5	-	ns	Figure 52.73
		VCC = 1.8 V or above		125	-		
		VCC = 1.6 V or above		250	-		
	CLKOUT pin high pulse width*2	VCC = 2.7 V or above	t_{CH}	15	-	ns	
		VCC = 1.8 V or above		30	-		
		VCC = 1.6 V or above		150	-		
	CLKOUT pin low pulse width*2	VCC = 2.7 V or above	t_{CL}	15	-	ns	
		VCC = 1.8 V or above		30	-		
		VCC = 1.6 V or above		150	-		
	CLKOUT pin output rise time	VCC = 2.7 V or above	t_{Cr}	-	12	ns	
		VCC = 1.8 V or above		-	25		
		VCC = 1.6 V or above		-	50		
	CLKOUT pin output fall time	VCC = 2.7 V or above	t_{Cf}	-	12	ns	
		VCC = 1.8 V or above		-	25		
		VCC = 1.6 V or above		-	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, the above should be satisfied with an input duty cycle of 45 to 55%.

Note 2. When the MOCO is selected as the clock output source (the CKOCR.CKOSEL[2:0] bits are 001b), set the clock output division ratio selection 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 52.73 CLKOUT output timing

52.4 USB Characteristics

52.4.1 USBFS Timing

Table 52.46 USB characteristics

Conditions: VCC = AVCC0 = VCC_USB = 3.0 to 3.6 V

Parameter			Symbol	Min	Max	Unit	Test conditions
Input characteristics	Input high level voltage		V_{IH}	2.0	-	V	-
	Input low level voltage		V_{IL}	-	0.8	V	-
	Differential input sensitivity		V_{DI}	0.2	-	V	| USB_DP - USB_DM |
	Differential common mode range		V_{CM}	0.8	2.5	V	-
Output characteristics	Output high level voltage		V_{OH}	2.8	VCC_USB	V	I_{OH} = –200 µA
	Output low level voltage		V_{OL}	0.0	0.3	V	I_{OL}= 2 mA
	Cross-over voltage		V_{CRS}	1.3	2.0	V	Figure 52.74, Figure 52.75, Figure 52.76
	Rise time	FS	t_r	4	20	ns	
		LS		75	300		
	Fall time	FS	t_f	4	20	ns	
		LS		75	300		
	Rise/fall time ratio	FS	t_r/t_f	90	111.11	%	
		LS		80	125		
	Output resistance		Z_{DRV}	28	44	Ω	(Adjusting the resistance of external elements is not necessary.)
VBUS characteristics	VBUS input voltage		V_{IH}	VCC × 0.8	-	V	-
			V_{IL}	-	VCC × 0.2	V	-
Pull-up, pull-down	Pull-down resistor		R_{PD}	14.25	24.80	kΩ	-
	Pull-up resistor		R_{PUI}	0.9	1.575	kΩ	During idle state
			R_{PUA}	1.425	3.09	kΩ	During reception
Battery Charging Specification version 1.2	D + sink current		I_{DP_SINK}	25	175	µA	-
	D – sink current		I_{DM_SINK}	25	175	µA	-
	DCD source current		I_{DP_SRC}	7	13	µA	-
	Data detection voltage		V_{DAT_REF}	0.25	0.4	V	-
	D + source voltage		V_{DP_SRC}	0.5	0.7	V	Output current = 250 µA
	D – source voltage		V_{DM_SRC}	0.5	0.7	V	Output current = 250 µA

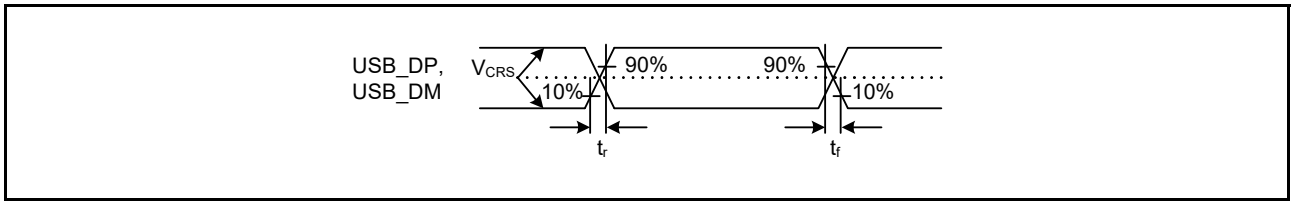


Figure 52.74 USB_DP and USB_DM output timing

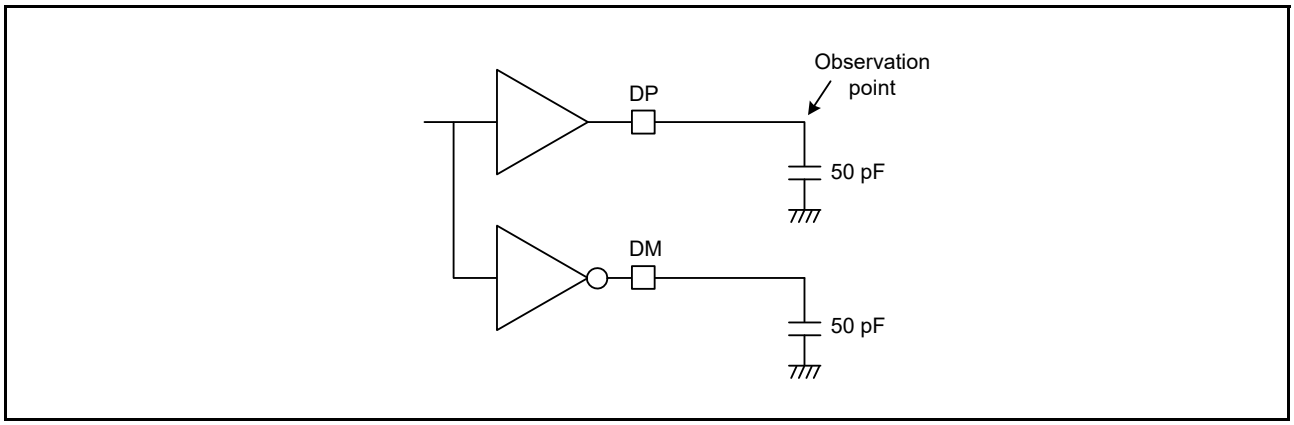


Figure 52.75 Test circuit for Full-Speed (FS) connection

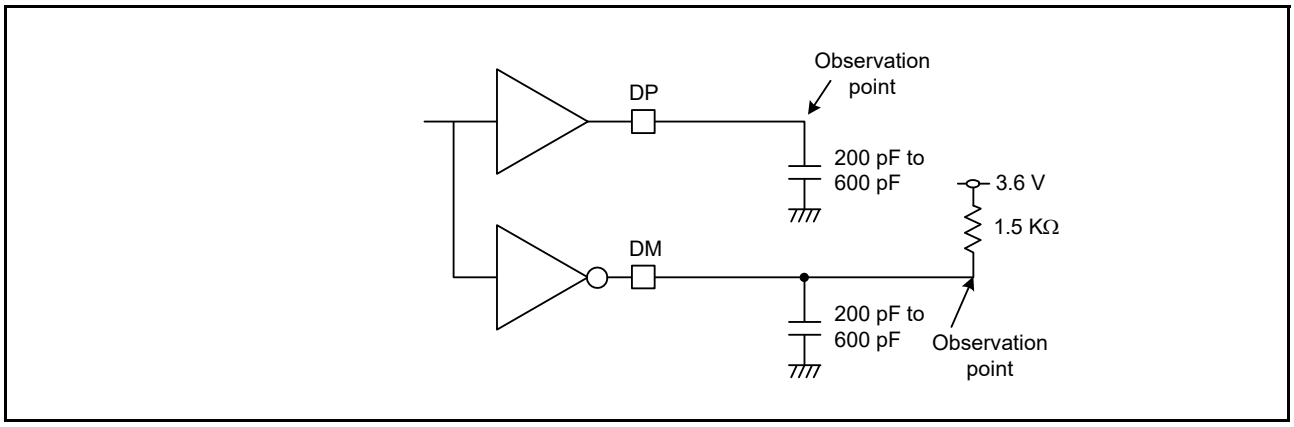


Figure 52.76 Test circuit for Low-Speed (LS) connection

52.4.2 USB External Supply

Table 52.47 USB regulator

Parameter		Min	Typ	Max	Unit	Test conditions
VCC_USB supply current	VCC_USB_LDO ≥ 3.8V	-	-	50	mA	-
	VCC_USB_LDO ≥ 4.5V	-	-	100	mA	-
VCC_USB supply voltage		3.0	-	3.6	V	-

52.5 ADC14 Characteristics

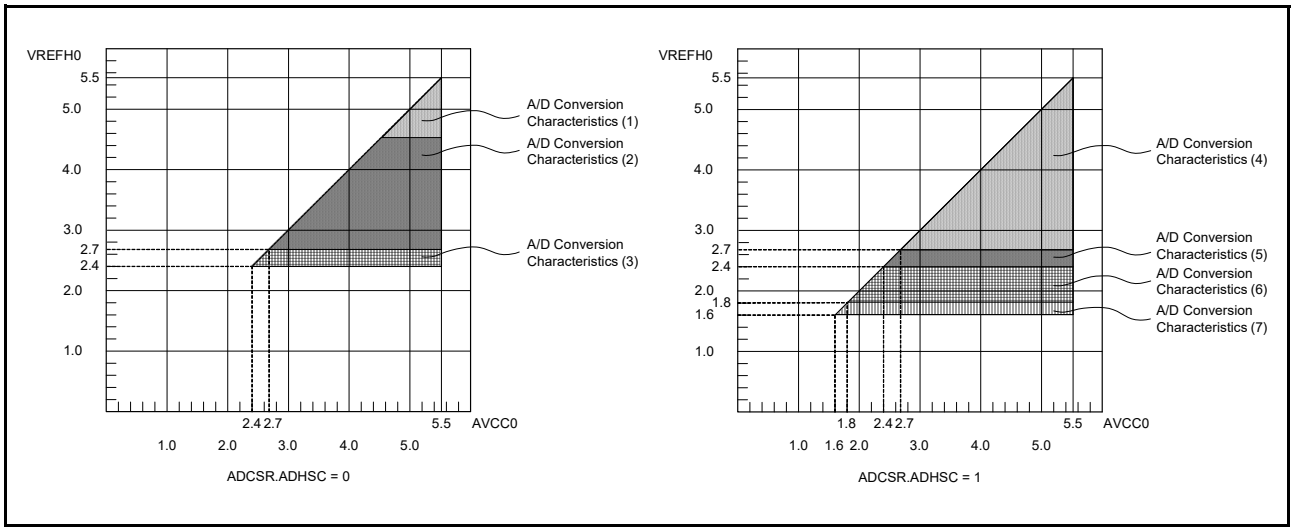


Figure 52.77 AVCC0 to VREFH0 voltage range

Table 52.48 A/D conversion characteristics (1) in High-speed A/D conversion mode (1 of 2)
Conditions: VCC = AVCC0 = 4.5 to 5.5 V, VREFH0 = 4.5 to 5.5 V, VSS = AVSS0 = VREFL0 = 0V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
Frequency		1	-	64	MHz	-
Analog input capacitance*2	Cs	-	-	8 (reference data)	pF	High-precision channel
		-	-	9 (reference data)	pF	Normal-precision channel
Analog input resistance	Rs	-	-	2.5 (reference data)	kΩ	High-precision channel
		-	-	6.7 (reference data)	kΩ	Normal-precision channel
Analog input voltage range	Ain	0	-	VREFH0	V	-
12-bit mode						
Resolution		-	-	12	Bit	-
Conversion time*1 (Operation at PCLKC = 64 MHz)	Permissible signal source impedance Max. = 0.3 kΩ	0.70	-	-	µs	High-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0Dh
		1.13	-	-	µs	Normal-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 28h
Offset error		-	±0.5	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Full-scale error		-	±0.75	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Quantization error		-	±0.5	-	LSB	-
Absolute accuracy		-	±1.25	±5.0	LSB	High-precision channel
				±8.0	LSB	Other than above
DNL differential nonlinearity error		-	±1.0	-	LSB	-
INL integral nonlinearity error		-	±1.0	±3.0	LSB	-
14-bit mode						
Resolution		-	-	14	Bit	-

Table 52.48 A/D conversion characteristics (1) in High-speed A/D conversion mode (2 of 2)
Conditions: VCC = AVCC0 = 4.5 to 5.5 V, VREFH0 = 4.5 to 5.5 V, VSS = AVSS0 = VREFL0 = 0V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
Conversion time*1 (Operation at PCLKC = 64 MHz)	Permissible signal source impedance Max. = 0.3 kΩ	0.80	-	-	µs	High-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0Dh
		1.22	-	-	µs	Normal-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 28h
Offset error		-	±2.0	±18	LSB	High-precision channel
				±24.0	LSB	Other than above
Full-scale error		-	±3.0	±18	LSB	High-precision channel
				±24.0	LSB	Other than above
Quantization error		-	±0.5	-	LSB	-
Absolute accuracy		-	±5.0	±20	LSB	High-precision channel
				±32.0	LSB	Other than above
DNL differential nonlinearity error		-	±4.0	-	LSB	-
INL integral nonlinearity error		-	±4.0	±12.0	LSB	-

Note: The characteristics apply when no pin functions other than 14-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 (C_{in}), see section 52.2.4, I/O V_{OH}, V_{OL}, and Other Characteristics.

Table 52.49 A/D conversion characteristics (2) in High-speed A/D conversion mode (1 of 2)
Conditions: VCC = AVCC0 = 2.7 to 5.5 V, VREFH0 = 2.7 to 5.5 V, VSS = AVSS0 = VREFL0 = 0V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
Frequency		1	-	48	MHz	-
Analog input capacitance*2	Cs	-	-	8 (reference data)	pF	High-precision channel
		-	-	9 (reference data)	pF	Normal-precision channel
Analog input resistance	Rs	-	-	2.5 (reference data)	kΩ	High-precision channel
		-	-	6.7 (reference data)	kΩ	Normal-precision channel
Analog input voltage range	Ain	0	-	VREFH0	V	-
12-bit mode						
Resolution		-	-	12	Bit	-
Conversion time*1 (Operation at PCLKC = 48 MHz)	Permissible signal source impedance Max. = 0.3 kΩ	0.94	-	-	µs	High-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0Dh
		1.50	-	-	µs	Normal-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 28h
Offset error		-	±0.5	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Full-scale error		-	±0.75	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Quantization error		-	±0.5	-	LSB	-
Absolute accuracy		-	±1.25	±5.0	LSB	High-precision channel
				±8.0	LSB	Other than above
DNL differential nonlinearity error		-	±1.0	-	LSB	-

Table 52.49 A/D conversion characteristics (2) in High-speed A/D conversion mode (2 of 2)
Conditions: VCC = AVCC0 = 2.7 to 5.5 V, VREFH0 = 2.7 to 5.5 V, VSS = AVSS0 = VREFL0 = 0V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
INL integral nonlinearity error		-	±1.0	±3.0	LSB	-
14-bit mode						
Resolution		-	-	14	Bit	-
Conversion time*1 (Operation at PCLKC = 48 MHz)	Permissible signal source impedance Max. = 0.3 kΩ	1.06	-	-	μs	High-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0Dh
		1.63	-	-	μs	Normal-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 28h
Offset error		-	±2.0	±18	LSB	High-precision channel
				±24.0	LSB	Other than above
Full-scale error		-	±3.0	±18	LSB	High-precision channel
				±24.0	LSB	Other than above
Quantization error		-	±0.5	-	LSB	-
Absolute accuracy		-	±5.0	±20	LSB	High-precision channel
				±32.0	LSB	Other than above
DNL differential nonlinearity error		-	±4.0	-	LSB	-
INL integral nonlinearity error		-	±4.0	±12.0	LSB	-

Note: The characteristics apply when no pin functions other than 14-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 (C_{in}), see section 52.2.4, I/O V_{OH}, V_{OL}, and Other Characteristics.

Table 52.50 A/D conversion characteristics (3) in High-speed A/D conversion mode (1 of 2)
Conditions: VCC = AVCC0 = 2.4 to 5.5 V, VREFH0 = 2.4 to 5.5 V, VSS = AVSS0 = VREFL0 = 0V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
Frequency		1	-	32	MHz	-
Analog input capacitance*2	Cs	-	-	8 (reference data)	pF	High-precision channel
		-	-	9 (reference data)	pF	Normal-precision channel
Analog input resistance	Rs	-	-	2.5 (reference data)	kΩ	High-precision channel
		-	-	6.7 (reference data)	kΩ	Normal-precision channel
Analog input voltage range	Ain	0	-	VREFH0	V	-
12-bit mode						
Resolution		-	-	12	Bit	-
Conversion time*1 (Operation at PCLKC = 32 MHz)	Permissible signal source impedance Max. = 1.3 kΩ	1.41	-	-	μs	High-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0Dh
		2.25	-	-	μs	Normal-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 28h
Offset error		-	±0.5	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Full-scale error		-	±0.75	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Quantization error		-	±0.5	-	LSB	-
Absolute accuracy		-	±1.25	±5.0	LSB	High-precision channel
				±8.0	LSB	Other than above

Table 52.50 A/D conversion characteristics (3) in High-speed A/D conversion mode (2 of 2)
Conditions: VCC = AVCC0 = 2.4 to 5.5 V, VREFH0 = 2.4 to 5.5 V, VSS = AVSS0 = VREFL0 = 0V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
DNL differential nonlinearity error		-	±1.0	-	LSB	-
INL integral nonlinearity error		-	±1.0	±3.0	LSB	-
14-bit mode						
Resolution		-	-	14	Bit	-
Conversion time*1 (Operation at PCLKC = 32 MHz)	Permissible signal source impedance Max. = 1.3 kΩ	1.59	-	-	µs	High-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 0Dh
		2.44	-	-	µs	Normal-precision channel ADCSR.ADHSC = 0 ADSSTRn.SST[7:0] = 28h
Offset error		-	±2.0	±18	LSB	High-precision channel
				±24.0	LSB	Other than above
Full-scale error		-	±3.0	±18	LSB	High-precision channel
				±24.0	LSB	Other than above
Quantization error		-	±0.5	-	LSB	-
Absolute accuracy		-	±5.0	±20	LSB	High-precision channel
				±32.0	LSB	Other than above
DNL differential nonlinearity error		-	±4.0	-	LSB	-
INL integral nonlinearity error		-	±4.0	±12.0	LSB	-

Note: The characteristics apply when no pin functions other than 14-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 (C_{in}), see section 52.2.4, I/O V_{OH}, V_{OL}, and Other Characteristics.

Table 52.51 A/D conversion characteristics (4) in Low-power A/D conversion mode (1 of 2)
Conditions: VCC = AVCC0 = 2.7 to 5.5 V, VREFH0 = 2.7 to 5.5 V, VSS = AVSS0 = VREFL0 = 0V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
Frequency		1	-	24	MHz	-
Analog input capacitance*2	Cs	-	-	8 (reference data)	pF	High-precision channel
		-	-	9 (reference data)	pF	Normal-precision channel
Analog input resistance	Rs	-	-	2.5 (reference data)	kΩ	High-precision channel
		-	-	6.7 (reference data)	kΩ	Normal-precision channel
Analog input voltage range	Ain	0	-	VREFH0	V	-
12-bit mode						
Resolution		-	-	12	Bit	-
Conversion time*1 (Operation at PCLKC = 24 MHz)	Permissible signal source impedance Max. = 1.1 kΩ	2.25	-	-	µs	High-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 0Dh
		3.38	-	-	µs	Normal-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 28h
Offset error		-	±0.5	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Full-scale error		-	±0.75	±4.5	LSB	High-precision channel
				±6.0	LSB	Other than above
Quantization error		-	±0.5	-	LSB	-

Table 52.51 A/D conversion characteristics (4) in Low-power A/D conversion mode (2 of 2)
Conditions: VCC = AVCC0 = 2.7 to 5.5 V, VREFH0 = 2.7 to 5.5 V, VSS = AVSS0 = VREFL0 = 0V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
Absolute accuracy		-	±1.25	±5.0	LSB	High-precision channel
				±8.0	LSB	Other than above
DNL differential nonlinearity error		-	±1.0	-	LSB	-
INL integral nonlinearity error		-	±1.0	±3.0	LSB	-
14-bit mode						
Resolution		-	-	14	Bit	-
Conversion time*1 (Operation at PCLKC = 24 MHz)	Permissible signal source impedance Max. = 1.1 kΩ	2.50	-	-	µs	High-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 0Dh
		3.63	-	-	µs	Normal-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 28h
Offset error		-	±2.0	±18	LSB	High-precision channel
				±24.0	LSB	Other than above
Full-scale error		-	±3.0	±18	LSB	High-precision channel
				±24.0	LSB	Other than above
Quantization error		-	±0.5	-	LSB	-
Absolute accuracy		-	±5.0	±20	LSB	High-precision channel
				±32.0	LSB	Other than above
DNL differential nonlinearity error		-	±4.0	-	LSB	-
INL integral nonlinearity error		-	±4.0	±12.0	LSB	-

Note: The characteristics apply when no pin functions other than 14-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 (C_{in}), see section 52.2.4, I/O V_{OH}, V_{OL}, and Other Characteristics.

Table 52.52 A/D conversion characteristics (5) in Low-power A/D conversion mode (1 of 2)
Conditions: VCC = AVCC0 = 2.4 to 5.5 V, VREFH0 = 2.4 to 5.5 V, VSS = AVSS0 = VREFL0 = 0V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter			Min	Typ	Max	Unit	Test conditions
Frequency			1	-	16	MHz	-
Analog input capacitance*2		Cs	-	-	8 (reference)	pF	High-precision channel
			-	-	9 (reference)	pF	Normal-precision channel
Analog input resistance		Rs	-	-	2.5 (reference)	kΩ	High-precision channel
			-	-	6.7 (reference)	kΩ	Normal-precision channel
Analog input voltage range		Ain	0	-	VREFH0	V	-
12-bit mode							
Resolution			-	-	12	Bit	-
Conversion time*1 (Operation at PCLKC = 16 MHz)	Permissible signal source impedance Max. = 2.2 kΩ		3.38	-	-	µs	High-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 0Dh
			5.06	-	-	µs	Normal-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 28h
Offset error			-	±0.5	±4.5	LSB	High-precision channel
					±6.0	LSB	Other than above
Full-scale error			-	±0.75	±4.5	LSB	High-precision channel
					±6.0	LSB	Other than above

Table 52.52 A/D conversion characteristics (5) in Low-power A/D conversion mode (2 of 2)
Conditions: VCC = AVCC0 = 2.4 to 5.5 V, VREFH0 = 2.4 to 5.5 V, VSS = AVSS0 = VREFL0 = 0V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
Quantization error		-	±0.5	-	LSB	-
Absolute accuracy		-	±1.25	±5.0	LSB	High-precision channel
				±8.0	LSB	Other than above
DNL differential nonlinearity error		-	±1.0	-	LSB	-
INL integral nonlinearity error		-	±1.0	±3.0	LSB	-
14-bit mode						
Resolution		-	-	14	Bit	-
Conversion time*1 (Operation at PCLKC = 16 MHz)	Permissible signal source impedance Max. = 2.2 kΩ	3.75	-	-	µs	High-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 0Dh
		5.44	-	-	µs	Normal-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 28h
Offset error		-	±2.0	±18	LSB	High-precision channel
				±24.0	LSB	Other than above
Full-scale error		-	±3.0	±18	LSB	High-precision channel
				±24.0	LSB	Other than above
Quantization error		-	±0.5	-	LSB	-
Absolute accuracy		-	±5.0	±20	LSB	High-precision channel
				±32.0	LSB	Other than above
DNL differential nonlinearity error		-	±4.0	-	LSB	-
INL integral nonlinearity error		-	±4.0	±12.0	LSB	-

Note: The characteristics apply when no pin functions other than 14-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 (C_{in}), see section 52.2.4, I/O V_{OH}, V_{OL}, and Other Characteristics.

Table 52.53 A/D conversion characteristics (6) in Low-power A/D conversion mode (1 of 2)
Conditions: VCC = AVCC0 = 1.8 to 5.5 V (AVCC0 = VCC when VCC < 2.0 V), VREFH0 = 1.8 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter			Min	Typ	Max	Unit	Test conditions
Frequency			1	-	8	MHz	-
Analog input capacitance*2		Cs	-	-	8 (reference data)	pF	High-precision channel
			-	-	9 (reference data)	pF	Normal-precision channel
Analog input resistance		Rs	-	-	3.8 (reference data)	kΩ	High-precision channel
			-	-	8.2 (reference data)	kΩ	Normal-precision channel
Analog input voltage range		Ain	0	-	VREFH0	V	-
12-bit mode							
Resolution			-	-	12	Bit	-
Conversion time*1 (Operation at PCLKC = 8 MHz)	Permissible signal source impedance Max. = 5 kΩ		6.75	-	-	µs	High-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 0Dh
			10.13	-	-	µs	Normal-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 28h
Offset error			-	±1.0	±7.5	LSB	High-precision channel
					±10.0	LSB	Other than above

Table 52.53 A/D conversion characteristics (6) in Low-power A/D conversion mode (2 of 2)

Conditions: VCC = AVCC0 = 1.8 to 5.5 V (AVCC0 = VCC when VCC < 2.0 V), VREFH0 = 1.8 to 5.5 V, VSS = AVSS0 = VREFL0 = 0 V
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
Full-scale error		-	±1.5	±7.5	LSB	High-precision channel
				±10.0	LSB	Other than above
Quantization error		-	±0.5	-	LSB	-
Absolute accuracy		-	±3.0	±8.0	LSB	High-precision channel
				±12.0	LSB	Other than above
DNL differential nonlinearity error		-	±1.0	-	LSB	-
INL integral nonlinearity error		-	±1.0	±3.0	LSB	-
14-bit mode						
Resolution		-	-	14	Bit	-
Conversion time*1 (Operation at PCLKC = 8 MHz)	Permissible signal source impedance Max. = 5 kΩ	7.50	-	-	µs	High-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 0Dh
		10.88	-	-	µs	Normal-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 28h
Offset error		-	±4.0	±30.0	LSB	High-precision channel
				±40.0	LSB	Other than above
Full-scale error		-	±6.0	±30.0	LSB	High-precision channel
				±40.0	LSB	Other than above
Quantization error		-	±0.5	-	LSB	-
Absolute accuracy		-	±12.0	±32.0	LSB	High-precision channel
				±48.0	LSB	Other than above
DNL differential nonlinearity error		-	±4.0	-	LSB	-
INL integral nonlinearity error		-	±4.0	±12.0	LSB	-

Note: The characteristics apply when no pin functions other than 14-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 (C_{in}), see section 52.2.4, I/O V_{OH}, V_{OL}, and Other Characteristics.

Table 52.54 A/D conversion characteristics (7) in Low-power A/D conversion mode (1 of 2)

Conditions: VCC = AVCC0 = 1.6 to 5.5 V (AVCC0 = VCC when VCC < 2.0 V), VREFH0 = 1.6 to 5.5 V, VSS = AVSS0 = VREFL0 = 0
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter			Min	Typ	Max	Unit	Test conditions
Frequency			1	-	4	MHz	-
Analog input capacitance*2		Cs	-	-	8 (reference data)	pF	High-precision channel
			-	-	9 (reference data)	pF	Normal-precision channel
Analog input resistance		Rs	-	-	13.1 (reference data)	kΩ	High-precision channel
			-	-	14.3 (reference data)	kΩ	Normal-precision channel
Analog input voltage range		Ain	0	-	VREFH0	V	-
12-bit mode							
Resolution			-	-	12	Bit	-
Conversion time*1 (Operation at PCLKC = 4 MHz)	Permissible signal source impedance Max. = 9.9 kΩ		13.5	-	-	µs	High-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 0Dh
			20.25	-	-	µs	Normal-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 28h

Table 52.54 A/D conversion characteristics (7) in Low-power A/D conversion mode (2 of 2)

Conditions: VCC = AVCC0 = 1.6 to 5.5 V (AVCC0 = VCC when VCC < 2.0 V), VREFH0 = 1.6 to 5.5 V, VSS = AVSS0 = VREFL0 = 0
Reference voltage range applied to the VREFH0 and VREFL0.

Parameter		Min	Typ	Max	Unit	Test conditions
Offset error		-	±1.0	±7.5	LSB	High-precision channel
				±10.0	LSB	Other than above
Full-scale error		-	±1.5	±7.5	LSB	High-precision channel
				±10.0	LSB	Other than above
Quantization error		-	±0.5	-	LSB	-
Absolute accuracy		-	±3.0	±8.0	LSB	High-precision channel
				±12.0	LSB	Other than above
DNL differential nonlinearity error		-	±1.0	-	LSB	-
INL integral nonlinearity error		-	±1.0	±3.0	LSB	-
14-bit mode						
Resolution		-	-	14	Bit	-
Conversion time*1 (Operation at PCLKC = 4 MHz)	Permissible signal source impedance Max. = 9.9 kΩ	15.0	-	-	µs	High-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 0Dh
		21.75	-	-	µs	Normal-precision channel ADCSR.ADHSC = 1 ADSSTRn.SST[7:0] = 28h
Offset error		-	±4.0	±30.0	LSB	High-precision channel
				±40.0	LSB	Other than above
Full-scale error		-	±6.0	±30.0	LSB	High-precision channel
				±40.0	LSB	Other than above
Quantization error		-	±0.5	-	LSB	-
Absolute accuracy		-	±12.0	±32.0	LSB	High-precision channel
				±48.0	LSB	Other than above
DNL differential nonlinearity error		-	±4.0	-	LSB	-
INL integral nonlinearity error		-	±4.0	±12.0	LSB	-

Note: The characteristics apply when no pin functions other than 14-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 (C_{in}), see section 52.2.4, I/O V_{OH}, V_{OL}, and Other Characteristics.

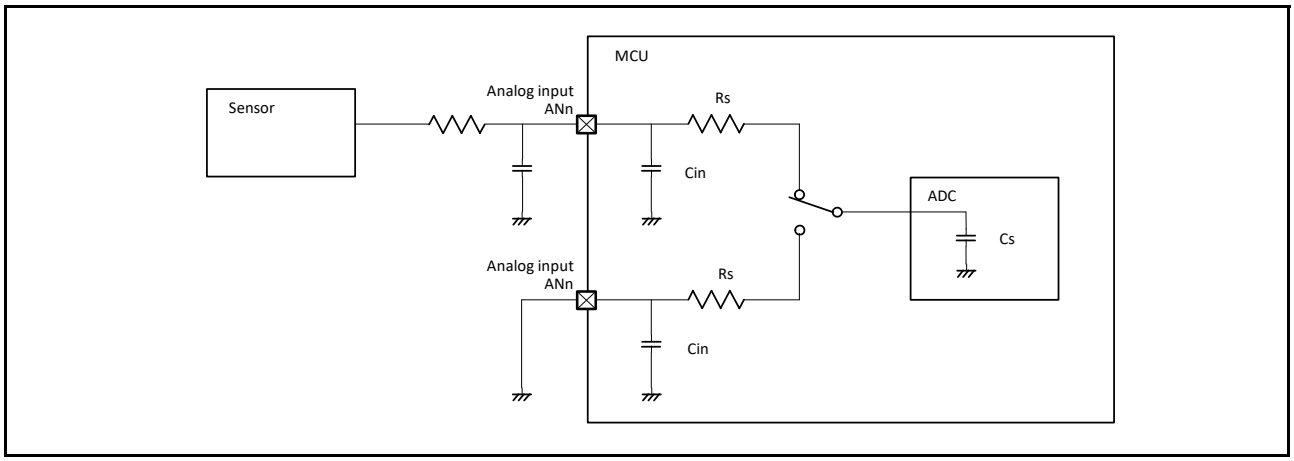


Figure 52.78 Equivalent circuit for analog input

Table 52.55 14-Bit A/D converter channel classification

Classification	Channel	Conditions	Remarks
High-precision channel	AN000 to AN015	AVCC0 = 1.6 to 5.5 V	Pins AN000 to AN015 cannot be used as general I/O, IRQ8, IRQ9 inputs, and TS transmission, when the A/D converter is in use
Normal-precision channel	AN016 to AN027		
Internal reference voltage input channel	Internal reference voltage	AVCC0 = 2.0 to 5.5 V	-
Temperature sensor input channel	Temperature sensor output	AVCC0 = 2.0 to 5.5 V	-

Table 52.56 A/D internal reference voltage characteristics

Conditions: VCC = AVCC0 = VREFH0 = 2.0 to 5.5 V*1

Parameter	Min	Typ	Max	Unit	Test conditions
Internal reference voltage input channel*2	1.36	1.43	1.50	V	-
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 < 2.0 V.
Note 2. The 14-bit A/D internal reference voltage indicates the voltage when the internal reference voltage is input to the 14-bit A/D converter.
Note 3. This is a parameter for ADC14 when the internal reference voltage is used as the high-potential reference voltage.
Note 4. This is a parameter for ADC14 when the internal reference voltage is selected for an analog input channel in ADC14.

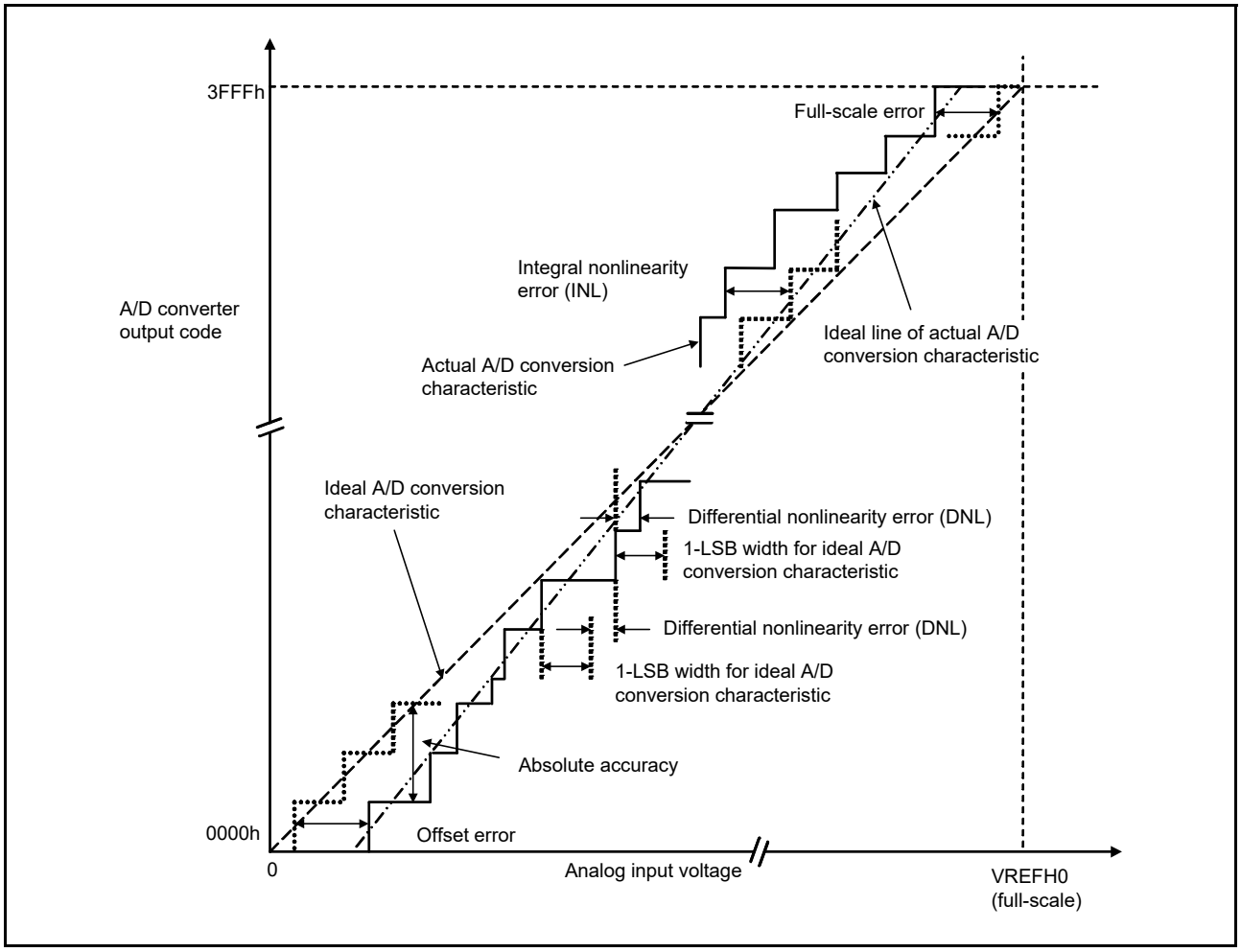


Figure 52.79 Illustration of 14-bit A/D converter characteristic terms

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 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, 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 003h to 00Dh, though an output code of 008h 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 actually 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.

52.6 DAC12 Characteristics

Table 52.57 D/A conversion characteristics (1)
Conditions: VCC = AVCC0 = 1.8 to 5.5 V
Reference voltage = VREFH or VREFL selected

Parameter	Min	Typ	Max	Unit	Test conditions
Resolution	-	-	12	bit	-
Resistive load	30	-	-	kΩ	-
Capacitive load	-	-	50	pF	-
Output voltage range	0.35	-	AVCC0 – 0.47	V	
DNL differential nonlinearity error	-	±0.5	±1.0	LSB	-
INL integral nonlinearity error	-	±2.0	±8.0	LSB	-
Offset error	-	-	±20	mV	-
Full-scale error	-	-	±20	mV	-
Output impedance	-	5	-	Ω	-
Conversion time	-	-	30	μs	-

Table 52.58 D/A conversion characteristics (2)
Conditions: VCC = AVCC0 = 1.8 to 5.5 V
Reference voltage = AVCC0 or AVSS0 selected

Parameter	Min	Typ	Max	Unit	Test conditions
Resolution	-	-	12	bit	-
Resistive load	30	-	-	kΩ	-
Capacitive load	-	-	50	pF	-
Output voltage range	0.35	-	AVCC0 – 0.47	V	-
DNL differential nonlinearity error	-	±0.5	±2.0	LSB	-
INL integral nonlinearity error	-	±2.0	±8.0	LSB	-
Offset error	-	-	±30	mV	-
Full-scale error	-	-	±30	mV	-
Output impedance	-	5	-	Ω	-
Conversion time	-	-	30	μs	-

Table 52.59 D/A conversioncharacteristics (3)
Conditions: VCC = AVCC0 = 1.8 to 5.5 V
Reference voltage = internal reference voltage selected

Parameter	Min	Typ	Max	Unit	Test conditions
Resolution	-	-	12	bit	-
Internal reference voltage (Vbgr)	1.36	1.43	1.50	V	-
Resistive load	30	-	-	kΩ	-
Capacitive load	-	-	50	pF	-
Output voltage range	0.35	-	Vbgr	V	-
DNL differential nonlinearity error	-	±2.0	±16.0	LSB	-
INL integral nonlinearity error	-	±8.0	±16.0	LSB	-
Offset error	-	-	±30	mV	-
Output impedance	-	5	-	Ω	-
Conversion time	-	-	30	μs	-

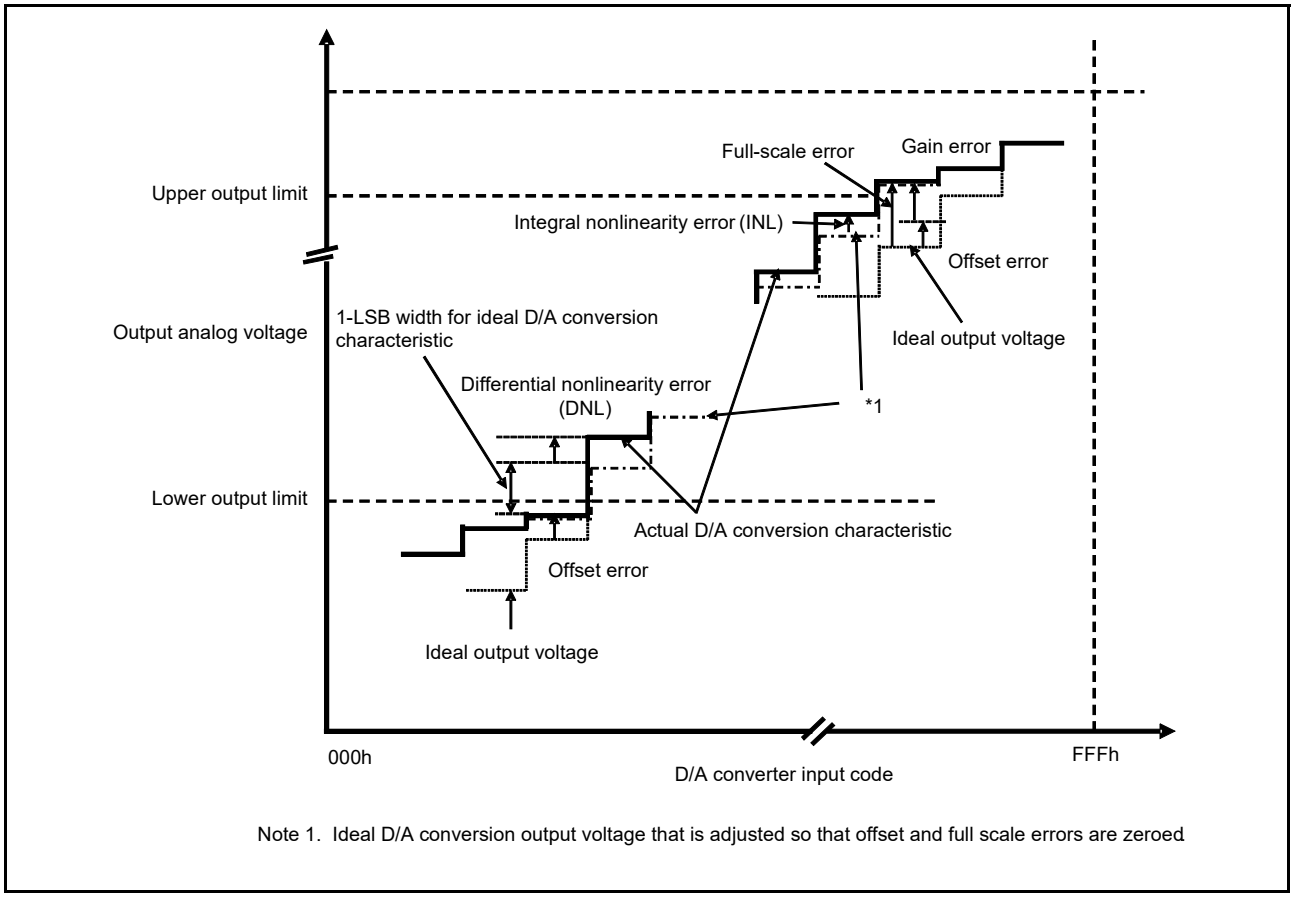


Figure 52.80 Illustration of D/A converter characteristic terms

Integral nonlinearity error (INL)

Integral nonlinearity error is the maximum deviation between the ideal output voltage based on the ideal conversion characteristic when the measured offset and full-scale errors are zeroed, and the actual output voltage.

Differential nonlinearity error (DNL)

Differential nonlinearity error is the difference between 1-LSB voltage width based on the ideal D/A conversion characteristics and the width of the actual output voltage.

Offset error

Offset error is the difference between the highest actual output voltage that falls below the lower output limit and the ideal output voltage based on the input code.

Full-scale error

Full-scale error is the difference between the lowest actual output voltage that exceeds the upper output limit and the ideal output voltage based on the input code.

52.7 TSN Characteristics

Table 52.60 TSN characteristics
Conditions: VCC = AVCC0 = 2.0 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	-	-	–3.65	-	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	-

52.8 OSC Stop Detect Characteristics

Table 52.61 Oscillation stop detection circuit characteristics

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Detection time	t_{dr}	-	-	1	ms	Figure 52.81

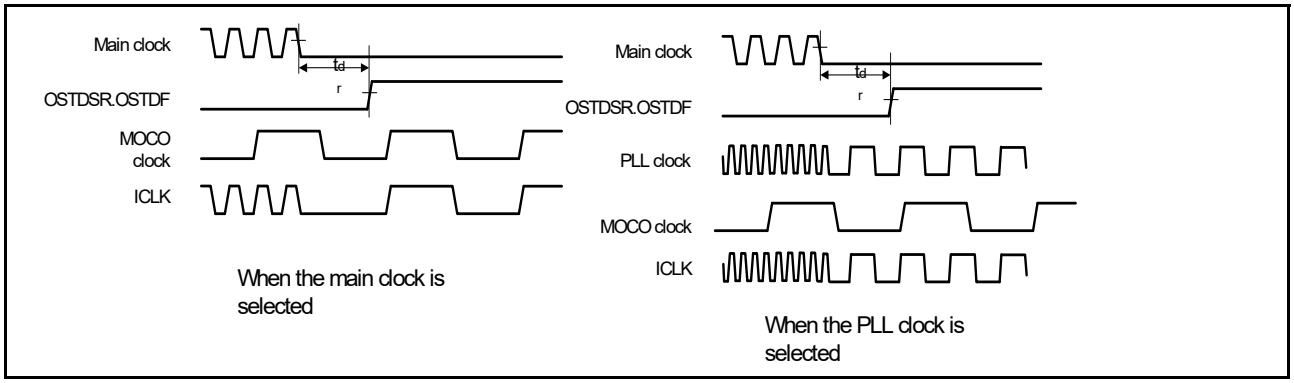


Figure 52.81 Oscillation stop detection timing

52.9 POR and LVD Characteristics

Table 52.62 Power-on reset circuit and voltage detection circuit characteristics (1)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Voltage detection level*1	Power-on reset (POR)	V_{POR}	1.27	1.42	1.57	V	Figure 52.82, Figure 52.83
	Voltage detection circuit (LVD0)*2	V_{det0_0}	3.68	3.85	4.00	V	Figure 52.84 At falling edge VCC
		V_{det0_1}	2.68	2.85	2.96		
		V_{det0_2}	2.38	2.53	2.64		
		V_{det0_3}	1.78	1.90	2.02		
		V_{det0_4}	1.60	1.69	1.82		
	Voltage detection circuit (LVD1)*3	V_{det1_0}	4.13	4.29	4.45	V	Figure 52.85 At falling edge VCC
		V_{det1_1}	3.98	4.16	4.30		
		V_{det1_2}	3.86	4.03	4.18		
		V_{det1_3}	3.68	3.86	4.00		
		V_{det1_4}	2.98	3.10	3.22		
		V_{det1_5}	2.89	3.00	3.11		
		V_{det1_6}	2.79	2.90	3.01		
		V_{det1_7}	2.68	2.79	2.90		
		V_{det1_8}	2.58	2.68	2.78		
		V_{det1_9}	2.48	2.58	2.68		
		V_{det1_A}	2.38	2.48	2.58		
		V_{det1_B}	2.10	2.20	2.30		
		V_{det1_C}	1.84	1.96	2.05		
		V_{det1_D}	1.74	1.86	1.95		
		V_{det1_E}	1.63	1.75	1.84		
		V_{det1_F}	1.60	1.65	1.73		
	Voltage detection circuit (LVD2)*4	V_{det2_0}	4.11	4.31	4.48	V	Figure 52.86 At falling edge VCC
		V_{det2_1}	3.97	4.17	4.34		
		V_{det2_2}	3.83	4.03	4.20		
		V_{det2_3}	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.VDSEL1[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 52.63 Power-on reset circuit and voltage detection circuit characteristics (2)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Wait time after power-on reset cancellation	LVD0:enable	t_{POR}	-	1.7	-	ms	-
	LVD0:disable	t_{POR}	-	1.3	-	ms	-
Wait time after voltage monitor 0,1,2 reset cancellation	LVD0:enable*1	$t_{LVD0,1,2}$	-	0.6	-	ms	-
	LVD0:disable*2	$t_{LVD1,2}$	-	0.2	-	ms	-
Response delay*3		t_{det}	-	-	350	µs	Figure 52.82, Figure 52.83
Minimum VCC down time		t_{VOFF}	450	-	-	µs	Figure 52.82, VCC = 1.0 V or above
Power-on reset enable time		$t_{W (POR)}$	1	-	-	ms	Figure 52.83, VCC = below 1.0 V
LVD operation stabilization time (after LVD is enabled)		$T_{d (E-A)}$	-	-	300	µs	Figure 52.85, Figure 52.86
Hysteresis width (POR)		V_{PORH}	-	110	-	mV	-
Hysteresis width (LVD0, LVD1 and LVD2)		V_{LVH}	-	60	-	mV	LVD0 selected
			-	100	-	mV	V_{det1_0} to V_{det1_2} selected.
			-	60	-		V_{det1_3} to V_{det1_9} selected.
			-	50	-		V_{det1_A} or V_{det1_B} selected.
			-	40	-		V_{det1_C} or V_{det1_F} selected.
			-	60	-		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.

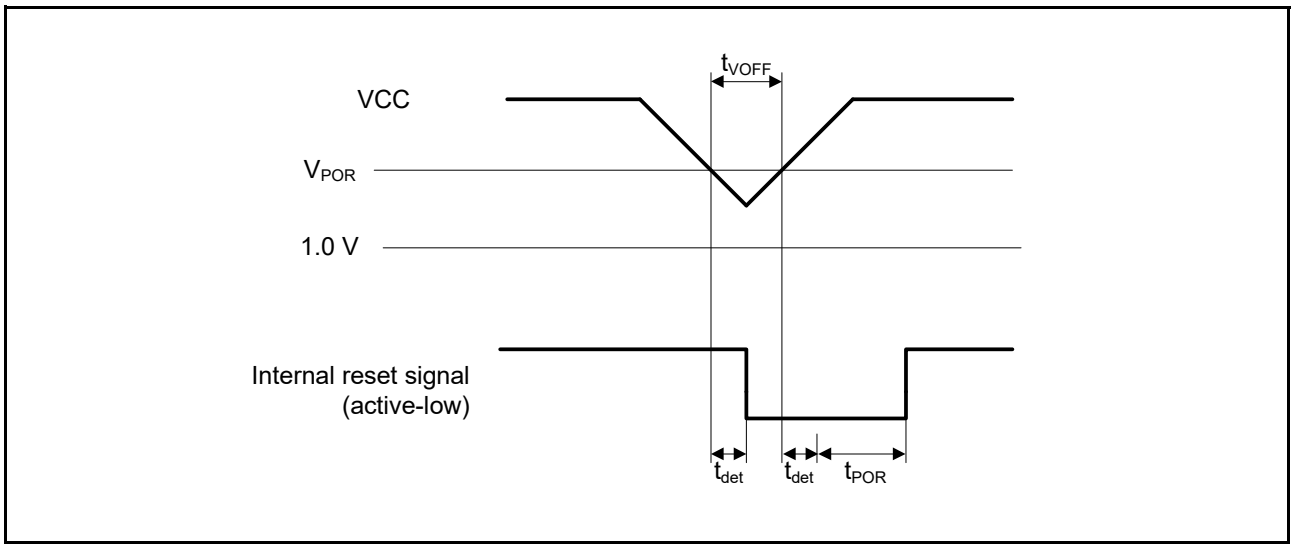
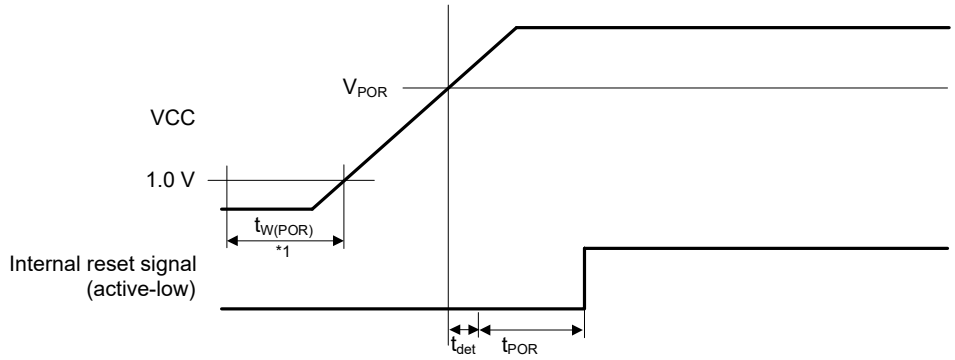


Figure 52.82 Voltage detection reset timing



Note: $t_{W(POR)}$ 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 52.83 Power-on reset timing

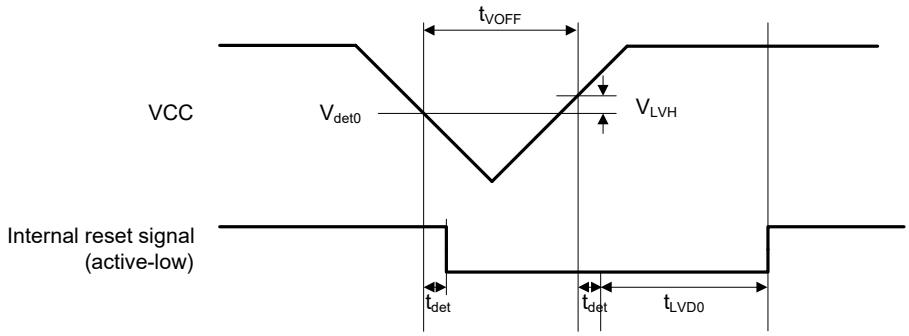


Figure 52.84 Voltage detection circuit timing (V_{det0})

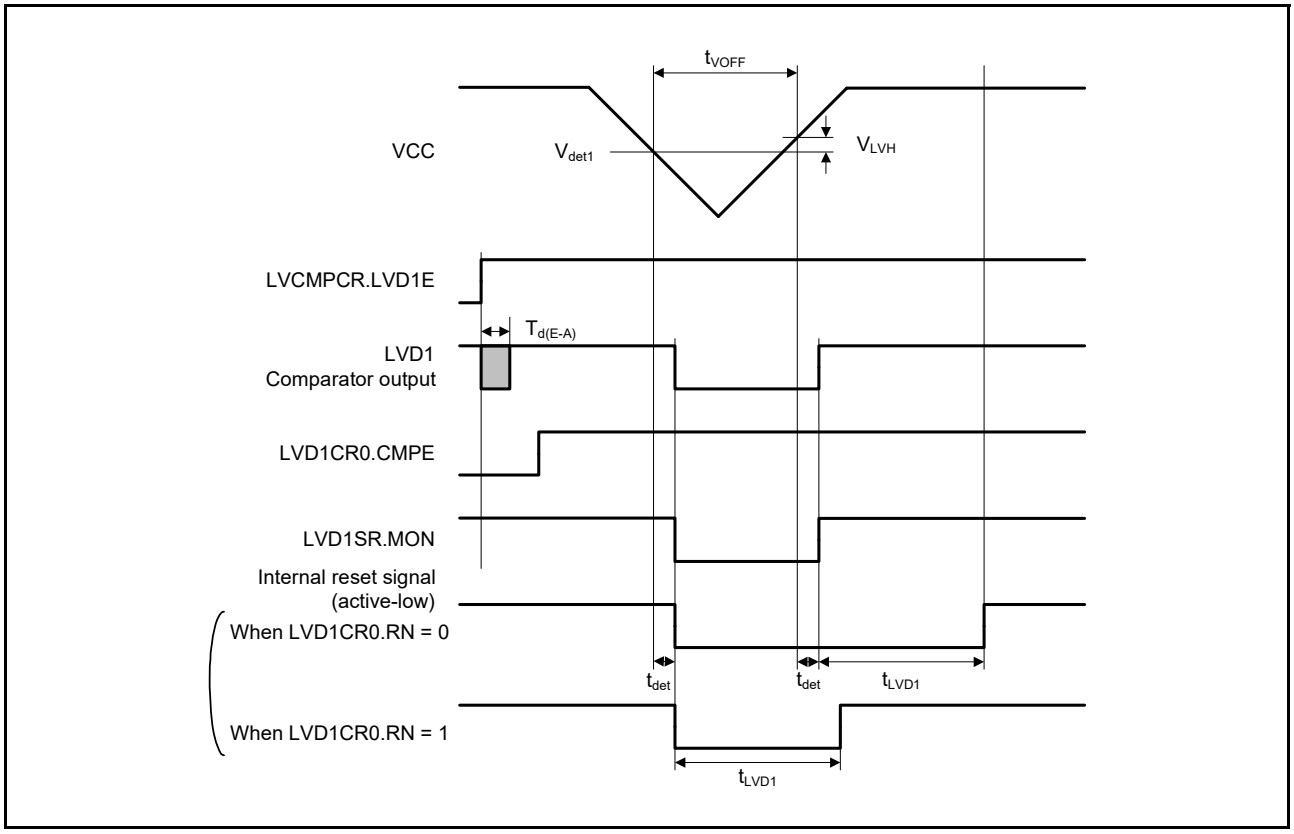


Figure 52.85 Voltage detection circuit timing (V_{det1})

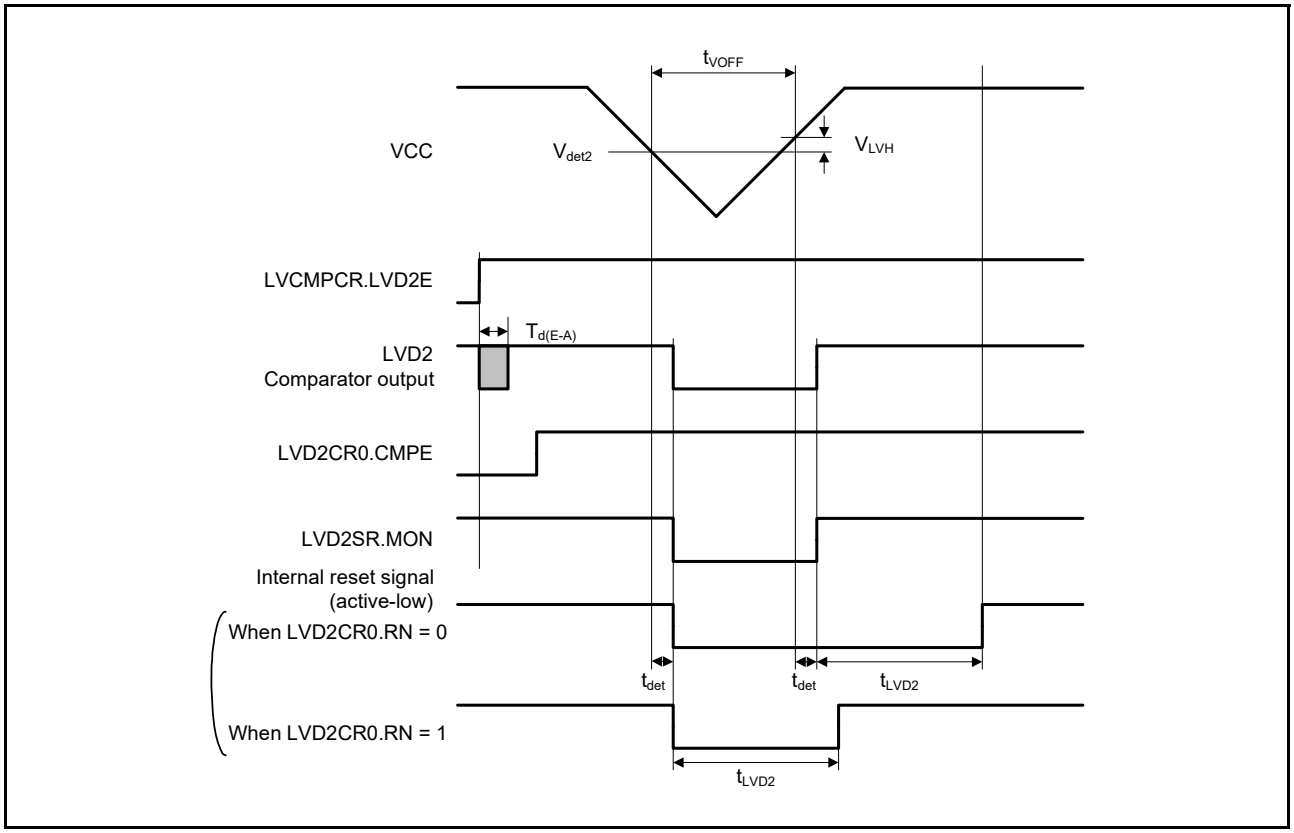


Figure 52.86 Voltage detection circuit timing (V_{det2})

52.10 Battery Backup Function Characteristics

Table 52.64 Battery Backup Function Characteristics
Conditions: VCC = AVCC0 = 1.6V to 5.5V, VBATT = 1.6 to 3.6 V, VSS = AVSS0 = 0V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Voltage level for switching to battery backup (falling)		$V_{DETBATT}$	1.99	2.09	2.19	V	Figure 52.87, Figure 52.88
Hysteresis width for switching to battery back up		V_{VBATTH}	-	100	-	mV	
VCC-off period for starting power supply switching		$t_{VOFFBATT}$	300	-	-	µs	-
Voltage detection level VBATT_Power-on reset (VBATT_POR)		$V_{VBATPOR}$	1.30	1.40	1.50	V	Figure 52.87, Figure 52.88
Wait time after VBATT_POR reset time cancellation		$t_{VBATPOR}$	-	-	3	mS	-
Level for detection of voltage drop on the VBATT pin (falling)	VBTLVDLVL[1:0] = 10b	$V_{DETBATLVD}$	2.11	2.2	2.29	V	Figure 52.89
	VBTLVDLVL[1:0] = 11b		1.92	2	2.08	V	
Hysteresis width for VBATT pin LVD		$V_{VBATLVDTH}$	-	50	-	mV	
VBATT pin LVD operation stabilization time		t_{d_vbat}	-	-	300	µs	Figure 52.89
VBATT pin LVD response delay time		t_{det_vbat}	-	-	350	µs	
Allowable voltage change rising/falling gradient		dt/dVCC	1.0	-	-	ms/V	-
VCC voltage level for access to the VBATT backup registers		$V_{_BKBATT}$	1.8	-	-	V	-

Note: The VCC-off period for starting power supply switching indicates the period in which VCC is below the minimum value of the voltage level for switching to battery backup ($V_{DETBATT}$).

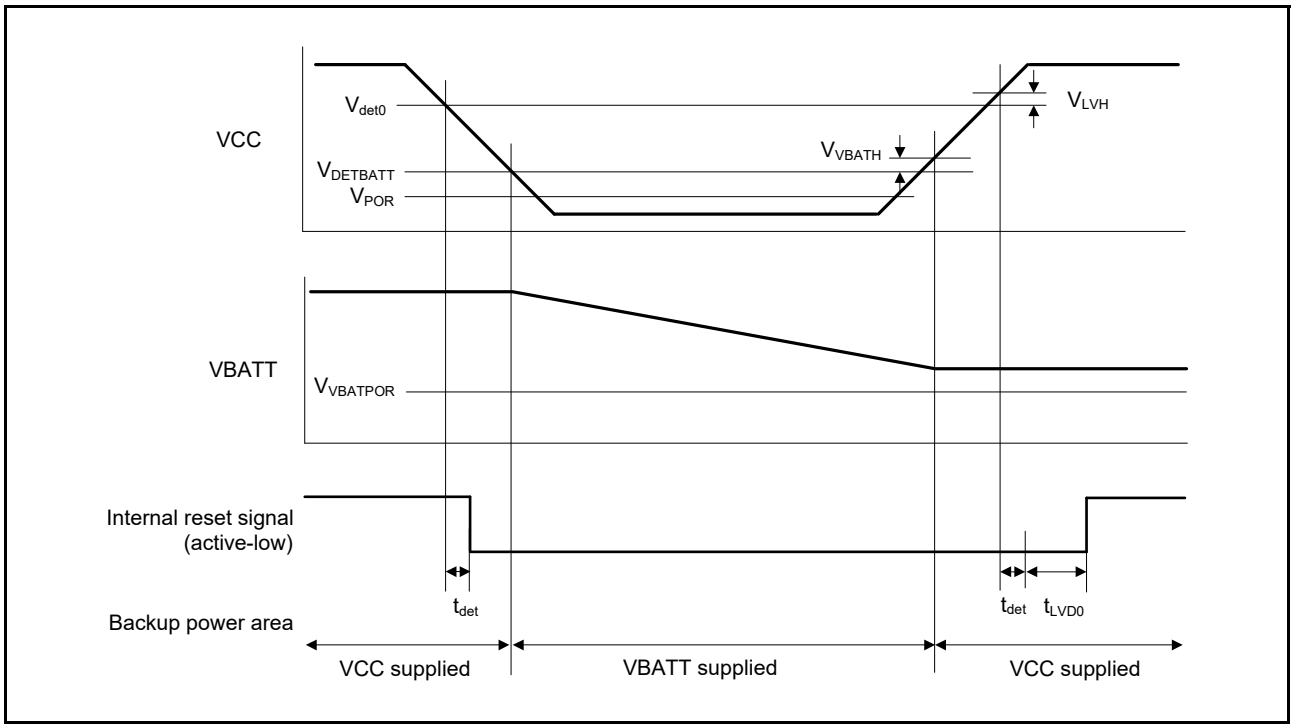


Figure 52.87 Power supply switching and LVD0 reset Timing

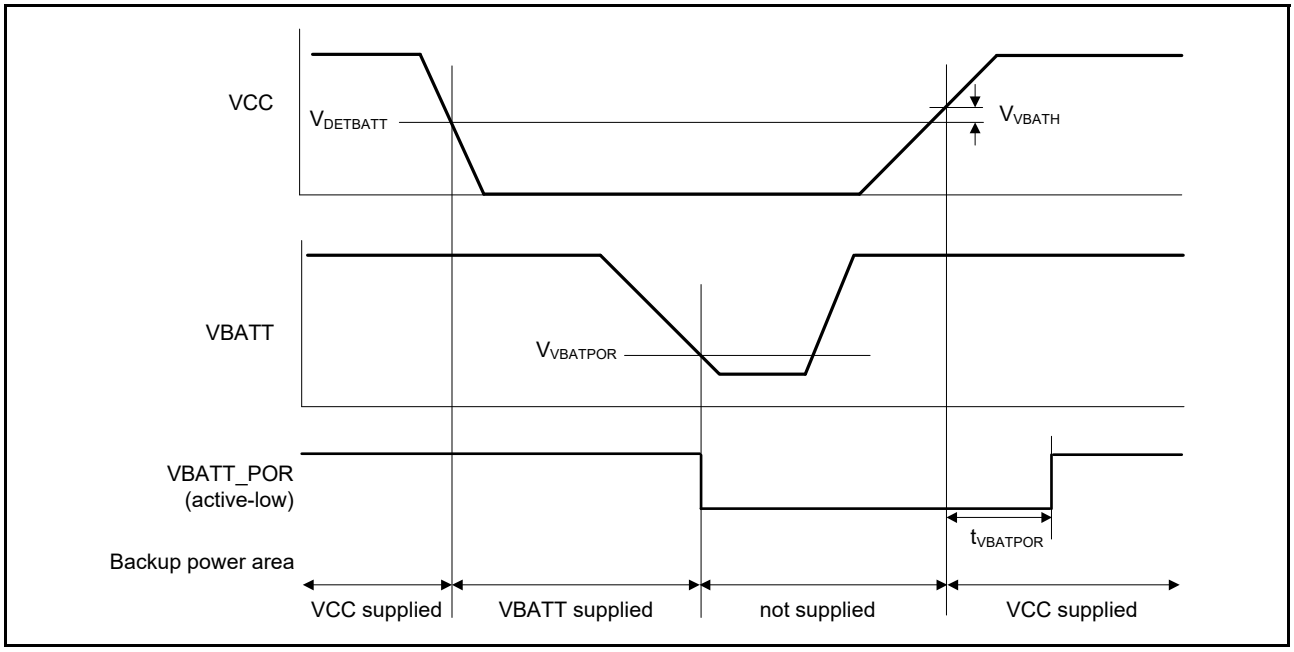


Figure 52.88 VBATT_POR reset timing

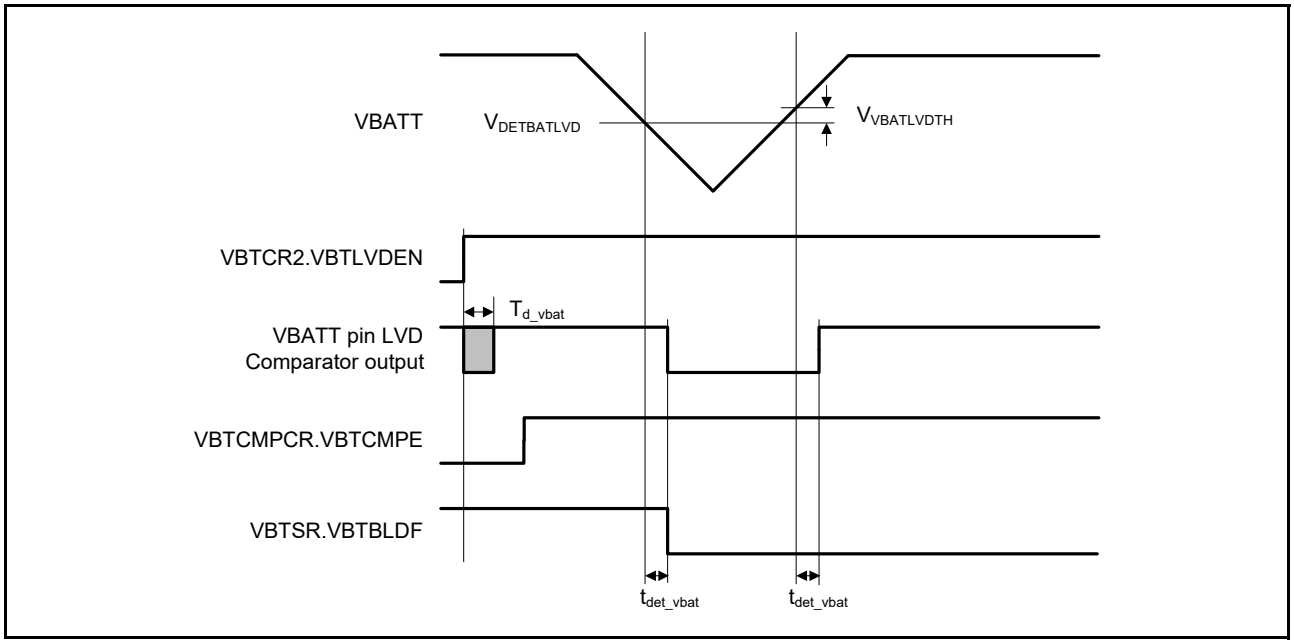


Figure 52.89 VBATT pin voltage detection circuit timing

Table 52.65 VBATT-I/O characteristics

Parameter			Symbol	Min	Typ	Max	Unit	Test conditions
VBATWIOn I/O output characteristics (n = 0 to 2)	VCC > $V_{DETBATT}$	VCC = 4.0 to 5.5 V	V_{OH}	VCC - 0.8	-	-	V	I_{OH} = -200 µA
			V_{OL}	-	-	0.8		I_{OL} = 200 µA
		VCC = 2.7 to 4.0 V	V_{OH}	VCC - 0.5	-	-		I_{OH} = -100 µA
			V_{OL}	-	-	0.5		I_{OL} = 100 µA
		VCC = $V_{DETBATT}$ to 2.7 V	V_{OH}	VCC - 0.3	-	-		I_{OH} = -50 µA
			V_{OL}	-	-	0.3		I_{OL} = 50 µA
	VCC < $V_{DETBATT}$	VBATT = 2.7 to 3.6 V	V_{OH}	V_{BATT} - 0.5	-	-		I_{OH} = -100 µA
			V_{OL}	-	-	0.5		I_{OL} = 100 µA
		VBATT = 1.6 to 2.7 V	V_{OH}	V_{BATT} - 0.3	-	-		I_{OH} = -50 µA
			V_{OL}	-	-	0.3		I_{OL} = 50 µA

52.11 CTSU Characteristics

Table 52.66 CTSU characteristics
Conditions: VCC = AVCC0 = 1.8 to 5.5 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
External capacitance connected to TSCAP pin	C_{tscap}	9	10	11	nF	-
TS pin capacitive load	C_{base}	-	-	50	pF	-
Permissible output high current	ΣIoH	-	-	-24	mA	When the mutual capacitance method is applied

52.12 Segment LCD Controller/Driver Characteristics

52.12.1 Resistance Division Method

[Static Display Mode]

Table 52.67 Resistance division method LCD characteristics (1)
Conditions: VL4 ≤ VCC ≤ 5.5 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
LCD drive voltage	V_{L4}	2.0	-	VCC	V	-

[1/2 Bias Method, 1/4 Bias Method]

Table 52.68 Resistance division method LCD characteristics (2)
Conditions: VL4 ≤ VCC ≤ 5.5 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
LCD drive voltage	V_{L4}	2.7	-	VCC	V	-

[1/3 Bias Method]

Table 52.69 Resistance division method LCD characteristics (3)
Conditions: VL4 ≤ VCC ≤ 5.5 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
LCD drive voltage	V_{L4}	2.5	-	VCC	V	-

52.12.2 Internal Voltage Boosting Method

[1/3 Bias Method]

Table 52.70 Internal voltage boosting method LCD characteristics

Conditions: VCC = AVCC0 = 1.8 V to 5.5 V

Parameter	Symbol	Conditions		Min	Typ	Max	Unit	Test conditions
LCD output voltage variation range	V_{L1}	C1 to C4*1 = 0.47 µF	VLCD = 04h	0.90	1.0	1.08	V	-
			VLCD = 05h	0.95	1.05	1.13	V	-
			VLCD = 06h	1.00	1.10	1.18	V	-
			VLCD = 07h	1.05	1.15	1.23	V	-
			VLCD = 08h	1.10	1.20	1.28	V	-
			VLCD = 09h	1.15	1.25	1.33	V	-
			VLCD = 0Ah	1.20	1.30	1.38	V	-
			VLCD = 0Bh	1.25	1.35	1.43	V	-
			VLCD = 0Ch	1.30	1.40	1.48	V	-
			VLCD = 0Dh	1.35	1.45	1.53	V	-
			VLCD = 0Eh	1.40	1.50	1.58	V	-
			VLCD = 0Fh	1.45	1.55	1.63	V	-
			VLCD = 10h	1.50	1.60	1.68	V	-
			VLCD = 11h	1.55	1.65	1.73	V	-
			VLCD = 12h	1.60	1.70	1.78	V	-
			VLCD = 13h	1.65	1.75	1.83	V	-
Doubler output voltage	V_{L2}	C1 to C4*1 = 0.47 µF		$2 \times V_{L1}$ - 0.1	$2 \times V_{L1}$	$2 \times V_{L1}$	V	-
Tripler output voltage	V_{L4}	C1 to C4*1 = 0.47 µF		$3 \times V_{L1}$ - 0.15	$3 \times V_{L1}$	$3 \times V_{L1}$	V	-
Reference voltage setup time*2	t_{VL1S}			5	-	-	ms	Figure 52.90
LCD output voltage variation range*3	t_{VLWT}	C1 to C4*1 = 0.47 µF		500	-	-	ms	

Note 1. This is a capacitor that is connected between voltage pins used to drive the LCD.
C1: A capacitor connected between CAPH and CAPL
C2: A capacitor connected between VL1 and GND
C3: A capacitor connected between VL2 and GND
C4: A capacitor connected between VL4 and GND
C1 = C2 = C3 = C4 = 0.47 µF ±30%

Note 2. This is the time required to wait from when the reference voltage is specified using the VLCD register (or when the internal voltage boosting method is selected (by setting the MDSET[1:0] bits in the LCDM0 register to 01b) if the default value reference voltage is used) until voltage boosting starts (VLCON = 1).

Note 3. This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

[1/4 Bias Method]

Table 52.71 Internal voltage boosting method LCD characteristics

Conditions: VCC = AVCC0 = 1.8 V to 5.5 V

Parameter	Symbol	Conditions		Min	Typ	Max	Unit	Test conditions
LCD output voltage variation range	V_{L1}	C1 to C5*1 = 0.47 µF	VLCD = 04h	0.90	1.0	1.08	V	-
			VLCD = 05h	0.95	1.05	1.13	V	-
			VLCD = 06h	1.00	1.10	1.18	V	-
			VLCD = 07h	1.05	1.15	1.23	V	-
			VLCD = 08h	1.10	1.20	1.28	V	-
			VLCD = 09h	1.15	1.25	1.33	V	-
			VLCD = 0Ah	1.20	1.30	1.38	V	-
			VLCD = 0Bh	1.25	1.35	1.43	V	-
			VLCD = 0Ch	1.30	1.40	1.48	V	-
Doubler output voltage	V_{L2}	C1 to C5*1 = 0.47 µF		$2V_{L1}$ - 0.08	$2V_{L1}$	$2V_{L1}$	V	-
Tripler output voltage	V_{L3}	C1 to C5*1 = 0.47 µF		$3V_{L1}$ - 0.12	$3V_{L1}$	$3V_{L1}$	V	-
Quadruply output voltage	V_{L4}*4	C1 to C5*1 = 0.47 µF		$4V_{L1}$ - 0.16	$4V_{L1}$	$4V_{L1}$	V	-
Reference voltage setup time*2	t_{VL1S}			5	-	-	ms	Figure 52.90
LCD output voltage variation range*3	t_{VLWT}	C1 to C5*1 = 0.47 µF		500	-	-	ms	

Note 1. This is a capacitor that is connected between voltage pins used to drive the LCD.
C1: A capacitor connected between CAPH and CAPL
C2: A capacitor connected between VL1 and GND
C3: A capacitor connected between VL2 and GND
C4: A capacitor connected between VL3 and GND
C5: A capacitor connected between VL4 and GND
C1 = C2 = C3 = C4 = C5 = 0.47 µF ± 30%

Note 2. This is the time required to wait from when the reference voltage is specified by using the VLCD register (or when the internal voltage boosting method is selected (by setting the MDSET1 and MDSET0 bits in the LCDM0 register to 01b) if the default value reference voltage is used) until voltage boosting starts (VLCON = 1).

Note 3. This is the wait time from when voltage boosting is started (VLCON = 1) until display is enabled (LCDON = 1).

Note 4. V_{L4} must be 5.5 V or lower.

52.12.3 Capacitor Split Method

[1/3 Bias Method]

Table 52.72 Internal voltage boostingmethod LCD characteristics

Conditions: VCC = AVCC0 = 2.2 V to 5.5 V

Parameter	Symbol	Conditions	Min	Typ	Max	Unit	Test conditions
VL4 voltage*1	V_{L4}	C1 to C4 = 0.47 µF*2	-	VCC	-	V	-
VL2 voltage*1	V_{L2}	C1 to C4 = 0.47 µF*2	2/3 × V_{L4} - 0.07	2/3 × V_{L4}	2/3 × V_{L4} + 0.07	V	-
VL1 voltage*1	V_{L1}	C1 to C4 = 0.47 µF*2	1/3 × V_{L4} - 0.08	1/3 × V_{L4}	1/3 × V_{L4} + 0.08	V	-
Capacitor split wait time*1	t_{WAIT}		100	-	-	ms	Figure 52.90

Note 1. This is the wait time from when voltage bucking is started (VLCON = 1) until display is enabled (LCDON = 1).
Note 2. This is a capacitor that is connected between voltage pins used to drive the LCD.
C1: A capacitor connected between CAPH and CAPL
C2: A capacitor connected between VL1 and GND
C3: A capacitor connected between VL2 and GND
C4: A capacitor connected between VL4 and GND
C1 = C2 = C3 = C4 = 0.47 µF ± 30%

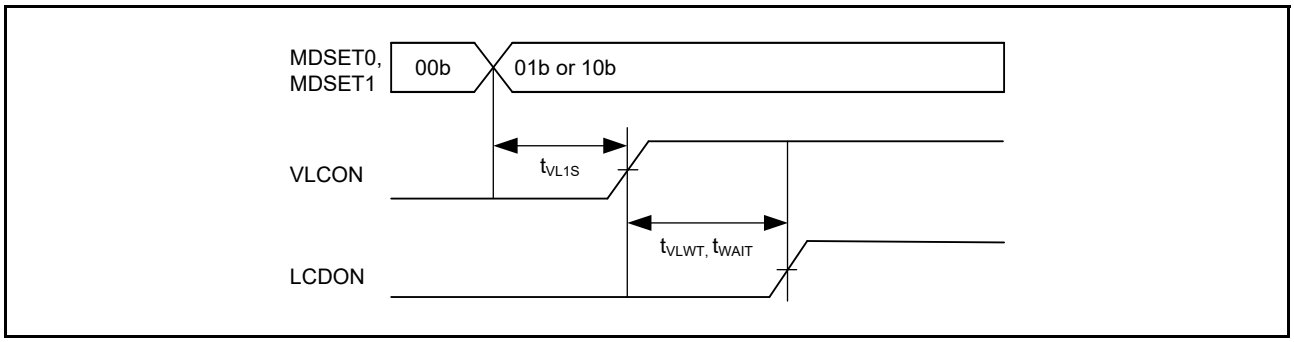


Figure 52.90 LCD reference voltage setup time, voltage boosting wait time, and capacitor split wait time

52.13 Comparator Characteristics

Table 52.73 ACMPHS characteristics
Conditions: VCC = AVCC0 = 2.7 to 5.5 V, VSS = AVSS0 = 0 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Input offset voltage	V_{IOCMP}	-	±5	±40	mV	-
Input voltage range	V_{ICMP}	0	-	AVCC0	V	-
Internal reference voltage	-	1.36	1.44	1.50	V	-
Input signal cycle	t_{PCMP}	10	-	-	µs	-
Output delay time	t_d	-	50	100	ns	Input amplitude ± 100 mV
Stabilization wait time during input channel switching*1	t_{WAIT}	300	-	-	ns	Input amplitude ± 100 mV
Operation stabilization wait time*2	t_{CMP}	1	-	-	µs	3.3 V ≤ AVCC0 ≤ 5.5 V
		3	-	-	µs	2.7 V ≤ AVCC0 < 3.3 V

Note 1. Period of time from when the comparator input channel is switched until the comparator is switched to output.
Note 2. Period of time from when the comparator operation is enabled (CMPCTL.HCMPON = 1) until the comparator satisfies the DC/AC characteristics.

Table 52.74 ACMPLP characteristics
Conditions: VCC = AVCC0 = 1.8 to 5.5 V, VSS = AVSS0 = 0 V

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Reference voltage range		VREF	0	-	VCC –1.4	V	-
Input voltage range		VI	0	-	VCC	V	-
Internal reference voltage		-	1.36	1.44	1.50	V	-
Output delay	High-speed mode	Td	-	-	1.2	µs	VCC = 3.0 Slew rate of input signal > 50 mV/µs
	Low-speed mode		-	-	5	µs	
	Window mode		-	-	2	µs	
Offset voltage	High-speed mode	-	-	-	50	mV	-
	Low-speed mode	-	-	-	40	mV	-
	Window mode	-	-	-	60	mV	-
Internal reference voltage for window mode		VRFH	-	0.76 × VCC	-	V	-
		VRFL	-	0.24 × VCC	-	V	-
Operation stabilization wait time		T_{cmp}	100	-	-	µs	-

52.14 OPAMP Characteristics

Table 52.75 OPAMP characteristics

Conditions: 1.8 V ≤ AVCC0 = VCC ≤ 5.5 V, VSS = AVSS0 = 0 V

Parameter	Symbol	Conditions		Min	Typ	Max	Unit
Common mode input range	Vicm1	Low power mode		0.2	-	AVCC0 – 0.5	V
	Vicm2	High-speed mode		0.3	-	AVCC0 – 0.6	V
Output voltage range	Vo1	Low power mode		0.1	-	AVCC0 – 0.1	V
	Vo2	High-speed mode		0.1	-	AVCC0 – 0.1	V
Input offset voltage	Vioff	3σ		–10	-	10	mV
Open gain	Av			60	120	-	dB
Gain-bandwidth (GB) product	GBW1	Low power mode		-	0.04	-	MHz
	GBW2	High-speed mode		-	1.7	-	MHz
Phase margin	PM	CL = 20 pF		50	-	-	deg
Gain margin	GM	CL = 20 pF		10	-	-	dB
Equivalent input noise	Vnoise1	f = 1 kHz	Low power mode	-	230	-	nV/√Hz
	Vnoise2	f = 10 kHz		-	200	-	nV/√Hz
	Vnoise3	f = 1 kHz	High-speed mode	-	90	-	nV/√Hz
	Vnoise4	f = 2 kHz		-	70	-	nV/√Hz
Power supply reduction ratio	PSRR			-	90	-	dB
Common mode signal reduction ratio	CMRR			-	90	-	dB
Stabilization wait time	Tstd1	CL = 20 pF Only operational amplifier is activated *1	Low power mode	650	-	-	µs
	Tstd2		High-speed mode	13	-	-	µs
	Tstd3	CL = 20 pF Operational amplifier and reference current circuit are activated simultaneously	Low power mode	650	-	-	µs
	Tstd4		High-speed mode	13	-	-	µs
Settling time	Tset1	CL = 20 pF	Low power mode	-	-	750	µs
	Tset2		High-speed mode	-	-	13	µs
Slew rate	Tslew1	CL = 20 pF	Low power mode	-	0.02	-	V/µs
	Tslew2		High-speed mode	-	1.1	-	V/µs
Load current	Iload1	Low power mode		–100	-	100	µA
	Iload2	High-speed mode		–100	-	100	µA
Load capacitance	CL			-	-	20	pF

Note 1. When the operational amplifier reference current circuit is activated in advance.

52.15 Flash Memory Characteristics

52.15.1 Code Flash Memory Characteristics

Table 52.76 Code flash characteristics (1)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Reprogramming/erasure cycle*1		N_{PEC}	1000	-	-	Times	-
Data hold time	After 1000 times of N_{PEC}	t_{DRP}	20*2, *3	-	-	Year	T_a = +85°C

Note 1. The reprogram/erase cycle is the number of erasure for each block. When the reprogram/erase cycle is n times (n = 1,000), erasing can be performed n times for each block. For instance, when 8-byte programming is performed 256 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. Characteristic when using the flash memory programmer and the self-programming library provided by Renesas Electronics.
Note 3. This result is obtained from reliability testing.

Table 52.77 Code flash characteristics (2)
High-speed operating mode
Conditions: VCC = AVCC0 = 2.7 to 5.5 V

Parameter		Symbol	FCLK = 1 MHz			FCLK = 32 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	8-byte	t_{P8}	-	116	998	-	54	506	µs
Erasure time	2-KB	t_{E2K}	-	9.03	287	-	5.67	222	ms
Blank check time	8-byte	t_{BC8}	-	-	56.8	-	-	16.6	µs
	2-KB	t_{BC2K}	-	-	1899	-	-	140	µs
Erase suspended time		t_{SED}	-	-	22.5	-	-	10.7	µs
Startup area switching setting time		t_{SAS}	-	21.7	585	-	12.1	447	ms
Access window time		t_{AWS}	-	21.7	585	-	12.1	447	ms
OCD/serial programmer ID setting time		t_{OSIS}	-	21.7	585	-	12.1	447	ms
Flash memory mode transition wait time 1		t_{DIS}	2	-	-	2	-	-	µs
Flash memory mode transition wait time 2		t_{MS}	5	-	-	5	-	-	µs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by the software.
Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing the flash memory. When using FCLK 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 FCLK must be ±3.5%. Confirm the frequency accuracy of the clock source.

Table 52.78 Code flash characteristics (3)
Middle-speed operating mode
Conditions: VCC = AVCC0 = 1.8 to 5.5 V, Ta = –40 to +85°C

Parameter		Symbol	FCLK = 1 MHz			FCLK = 8 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	8-byte	t_{P8}	-	157	1411	-	101	966	µs
Erasure time	2-KB	t_{E2K}	-	9.10	289	-	6.10	228	ms
Blank check time	8-byte	t_{BC8}	-	-	87.7	-	-	52.5	µs
	2-KB	t_{BC2K}	-	-	1930	-	-	414	µs
Erase suspended time		t_{SED}	-	-	32.7	-	-	21.6	µs
Startup area switching setting time		t_{SAS}	-	22.5	592	-	14.0	464	ms
Access window time		t_{AWS}	-	22.5	592	-	14.0	464	ms
OCD/serial programmer ID setting time		t_{OSIS}	-	22.5	592	-	14.0	464	ms
Flash memory mode transition wait time 1		t_{DIS}	2	-	-	2	-	-	µs
Flash memory mode transition wait time 2		t_{MS}	720	-	-	720	-	-	ns

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by the software.
Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing the flash memory. When using FCLK 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 FCLK must be ±3.5%. Confirm the frequency accuracy of the clock source.

52.15.2 Data Flash Memory Characteristics

Table 52.79 Data flash characteristics (1)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions
Reprogramming/erasure cycle*1		N_{DPEC}	100,000	1,000,000	-	Times	-
Data hold time	After 10000 times of N_{DPEC}	t_{DDRP}	20*2, *3	-	-	Year	Ta = +85°C
	After 100000 times of N_{DPEC}		5*2, *3	-	-	Year	
	After 1000000 times of N_{DPEC}		-	1*2, *3	-	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 1,000 times for different addresses in 1-byte 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. Characteristics when using the flash memory programmer and the self-programming library provided by Renesas Electronics.
Note 3. These results are obtained from reliability testing.

Table 52.80 Data flash characteristics (2)
High-speed operating mode
Conditions: VCC = AVCC0 = 2.7 to 5.5 V

Parameter		Symbol	FCLK = 4 MHz			FCLK = 32 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	1-byte	t_{DP1}	-	52.4	463	-	42.1	387	µs
Erasure time	1-KB	t_{DE1K}	-	8.98	286	-	6.42	237	ms
Blank check time	1-byte	t_{DBC1}	-	-	24.3	-	-	16.6	µs
	1-KB	t_{DBC1K}	-	-	1872	-	-	512	µs
Suspended time during erasing		t_{DSED}	-	-	13.0	-	-	10.7	µs
Data flash STOP recovery time		t_{DSTOP}	5	-	-	5	-	-	µs

Note 1. Does not include the time until each operation of the flash memory is started after instructions are executed by the software.
Note 2. The lower-limit frequency of FCLK is 1 MHz during programming or erasing the flash memory. When using FCLK 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 3. The frequency accuracy of FCLK must be ±3.5%. Confirm the frequency accuracy of the clock source.

Table 52.81 Data flash characteristics (3)
Middle-speed operating mode
Conditions: VCC = AVCC0 = 1.8 to 5.5 V, Ta = –40 to +85°C

Parameter		Symbol	FCLK = 4 MHz			FCLK = 8 MHz			Unit
			Min	Typ	Max	Min	Typ	Max	
Programming time	1-byte	t_{DP1}	-	94.7	886	-	89.3	849	µs
Erasure time	1-KB	t_{DE1K}	-	9.59	299	-	8.29	273	ms
Blank check time	1-byte	t_{DBC1}	-	-	56.2	-	-	52.5	µs
	1-KB	t_{DBC1K}	-	-	2.17	-	-	1.51	ms
Suspended time during erasing		t_{DSED}	-	-	23.0	-	-	21.7	µs
Data flash STOP recovery time		t_{DSTOP}	720	-	-	720	-	-	ns

Note 1. Does not include the time until each operation of the flash memory is started after instructions are executed by the software.
Note 2. The lower-limit frequency of FCLK is 1 MHz during programming or erasing the flash memory. When using FCLK 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 3. The frequency accuracy of FCLK must be ±3.5%. Confirm the frequency accuracy of the clock source.

52.16 Boundary Scan

Table 52.82 Boundary scan
Conditions: VCC = AVCC0 = 2.4 to 5.5 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
TCK clock cycle time	t_{TCKcyc}	100	-	-	ns	Figure 52.91
TCK clock high pulse width	t_{TCKH}	45	-	-	ns	
TCK clock low pulse width	t_{TCKL}	45	-	-	ns	
TCK clock rise time	t_{TCKr}	-	-	5	ns	
TCK clock fall time	t_{TCKf}	-	-	5	ns	
TMS setup time	t_{TMSS}	20	-	-	ns	Figure 52.92
TMS hold time	t_{TMSH}	20	-	-	ns	
TDI setup time	t_{TDIS}	20	-	-	ns	
TDI hold time	t_{TDIH}	20	-	-	ns	
TDO data delay	t_{TDOD}	-	-	70	ns	
Boundary Scan circuit start up time*1	t_{BSSTUP}	t_{RESWP}	-	-	-	Figure 52.93

Note 1. Boundary scan does not function until Power-On-Reset becomes negative.

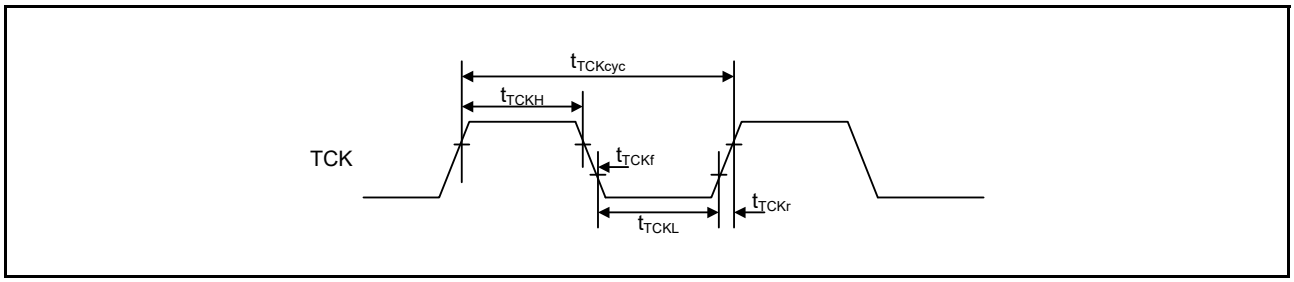


Figure 52.91 Boundary scan TCK timing

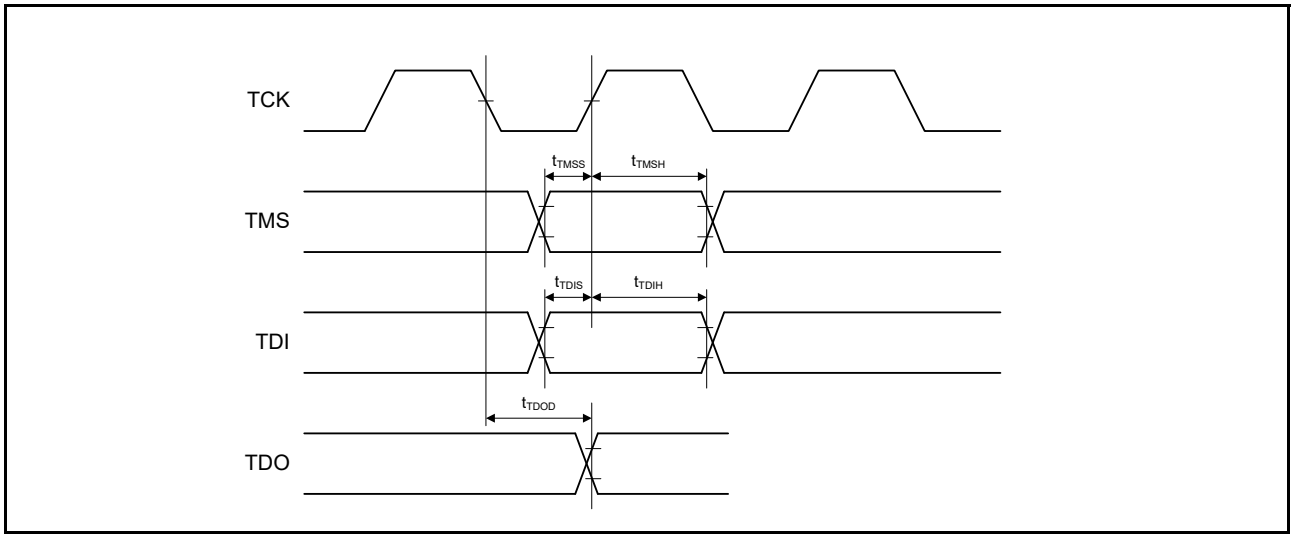


Figure 52.92 Boundary scan input/output timing

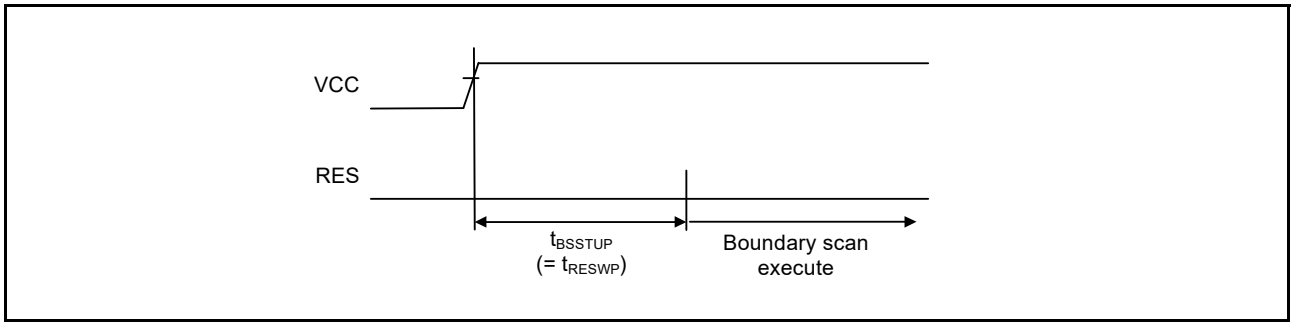


Figure 52.93 Boundary scan circuit start up timing

52.17 Joint Test Action Group (JTAG)

Table 52.83 JTAG (Debug) characteristics (1)
Conditions: VCC = AVCC0 = 2.4 to 5.5 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
TCK clock cycle time	t_{TCKcyc}	80	-	-	ns	Figure 52.94
TCK clock high pulse width	t_{TCKH}	35	-	-	ns	
TCK clock low pulse width	t_{TCKL}	35	-	-	ns	
TCK clock rise time	t_{TCKr}	-	-	5	ns	
TCK clock fall time	t_{TCKf}	-	-	5	ns	
TMS setup time	t_{TMSS}	16	-	-	ns	Figure 52.95
TMS hold time	t_{TMSH}	16	-	-	ns	
TDI setup time	t_{TDIS}	16	-	-	ns	
TDI hold time	t_{TDIH}	16	-	-	ns	
TDO data delay time	t_{TDOD}	-	-	70	ns	

Table 52.84 JTAG (Debug) characteristics (2)
Conditions: VCC = AVCC0 = 1.6 to 2.4 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
TCK clock cycle time	t_{TCKcyc}	250	-	-	ns	Figure 52.94
TCK clock high pulse width	t_{TCKH}	120	-	-	ns	
TCK clock low pulse width	t_{TCKL}	120	-	-	ns	
TCK clock rise time	t_{TCKr}	-	-	5	ns	
TCK clock fall time	t_{TCKf}	-	-	5	ns	
TMS setup time	t_{TMSS}	50	-	-	ns	Figure 52.95
TMS hold time	t_{TMSH}	50	-	-	ns	
TDI setup time	t_{TDIS}	50	-	-	ns	
TDI hold time	t_{TDIH}	50	-	-	ns	
TDO data delay time	t_{TDOD}	-	-	150	ns	

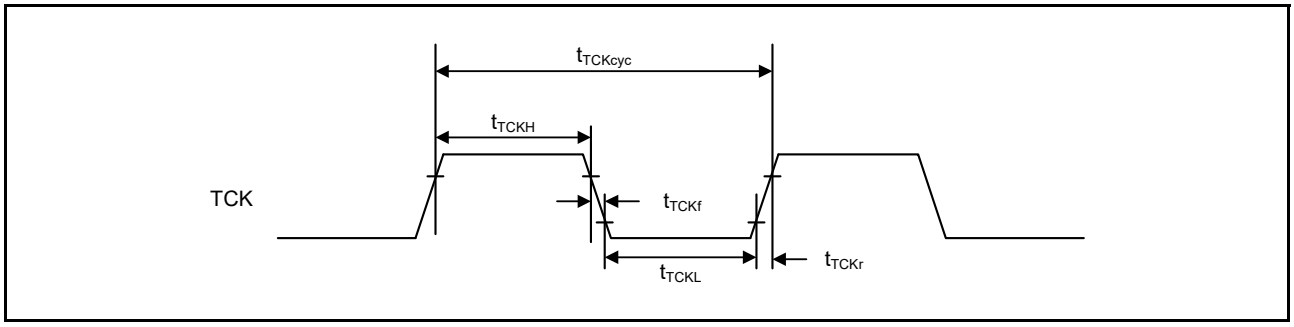


Figure 52.94 JTAG TCK timing

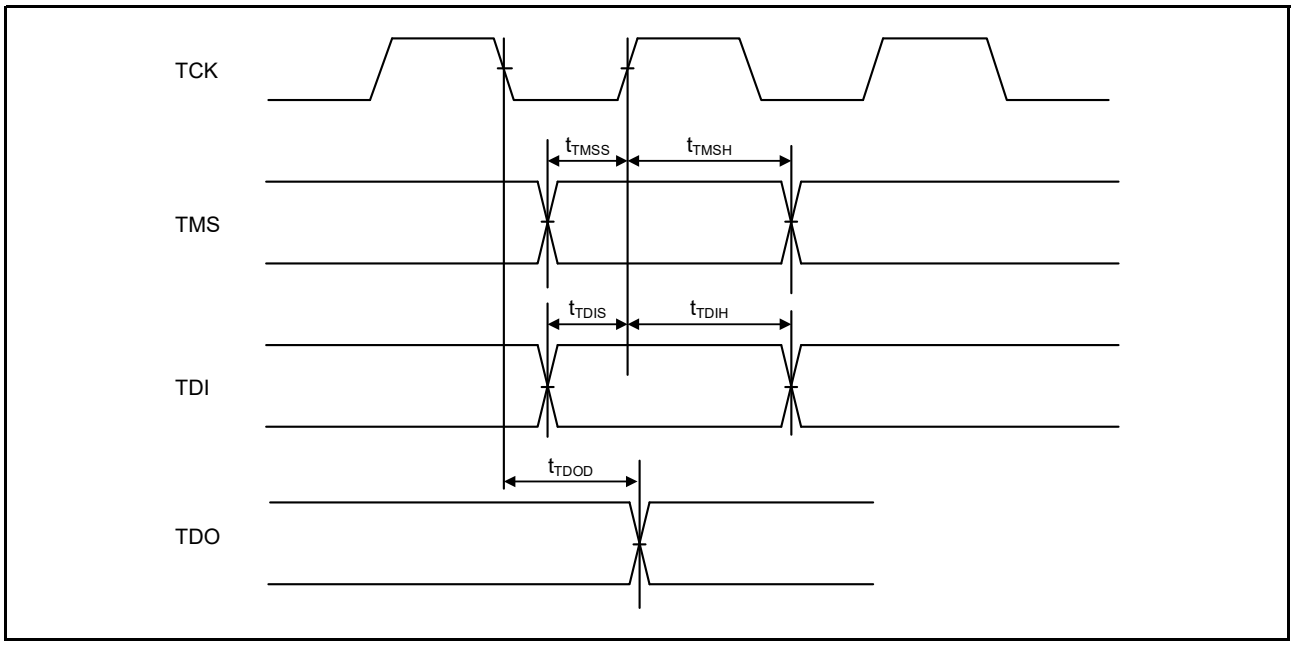


Figure 52.95 JTAG input/output timing

52.17.1 Serial Wire Debug (SWD)

Table 52.85 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 52.96
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 52.97
SWDIO hold time	t_{SWDH}	16	-	-	ns	
SWDIO data delay time	t_{SWDD}	2	-	70	ns	

Table 52.86 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 52.96
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 52.97
SWDIO hold time	t_{SWDH}	50	-	-	ns	
SWDIO data delay time	t_{SWDD}	2	-	150	ns	

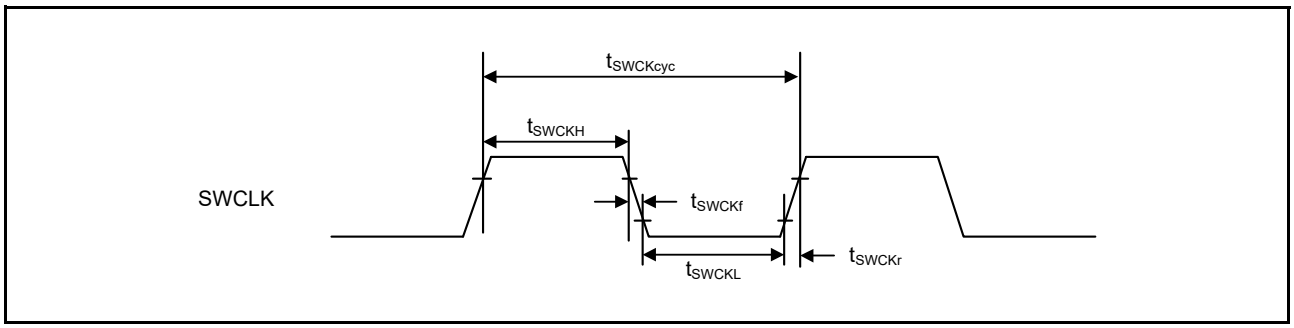


Figure 52.96 SWD SWCLK timing

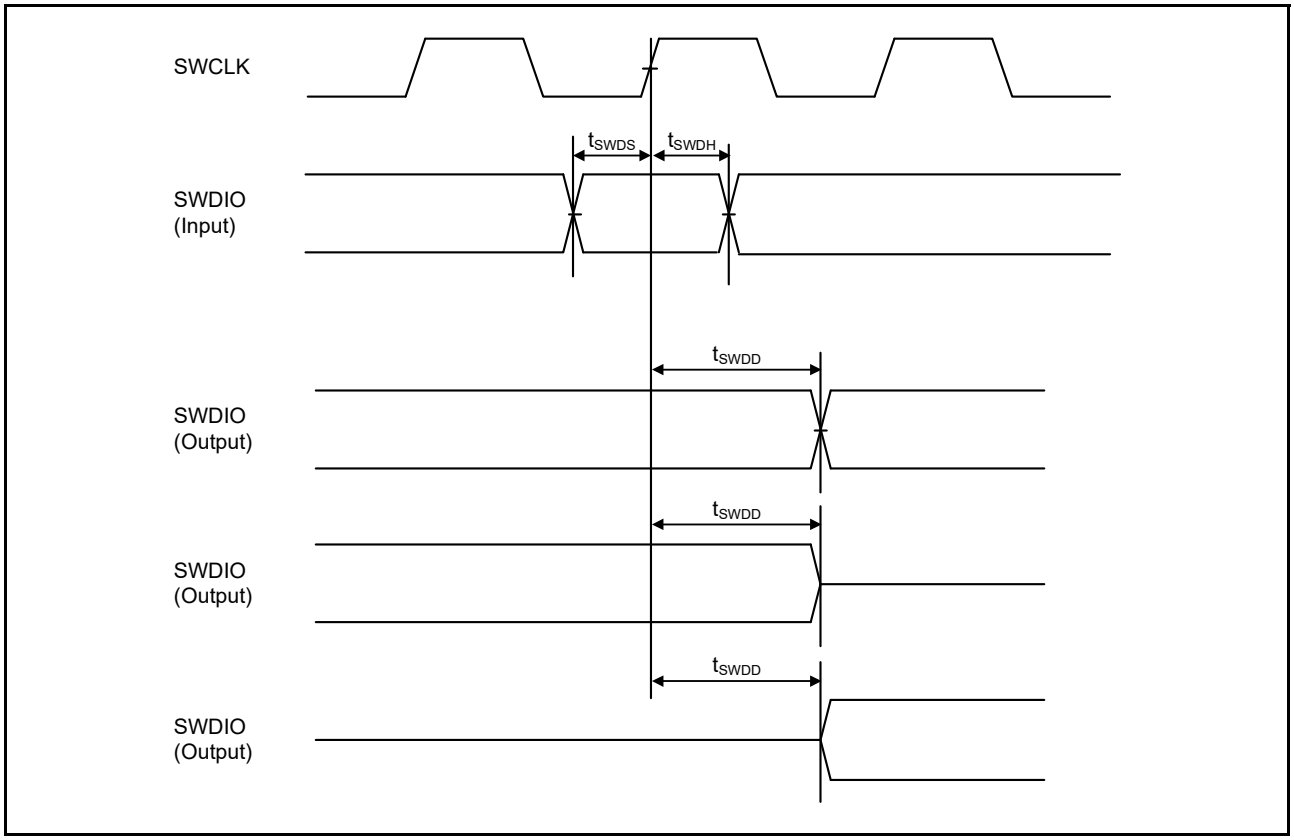


Figure 52.97 SWD input/output timing

Appendix 1. Port States in Each Processing Mode

Table 1.1 Port states in each processing state (1 of 5)

Port Name	Reset	Software Standby Mode	
		OPE = 0	OPE = 1
P000/IRQ6,	Hi-Z	Keep-O*1	
P001/IRQ7,	Hi-Z	Keep-O*1	
P002/IRQ8	Hi-Z	Keep-O*1	
P003	Hi-Z	Keep-O	
P004/IRQ9	Hi-Z	Keep-O*1	
P005/IRQ10	Hi-Z	Keep-O*1	
P006/IRQ11	Hi-Z	Keep-O*1	
P007	Hi-Z	Keep-O	
P008/IRQ12	Hi-Z	Keep-O*1	
P009/IRQ13	Hi-Z	Keep-O*1	
P010/IRQ14	Hi-Z	Keep-O*1	
P011/IRQ15	Hi-Z	Keep-O*1	
P012, P013	Hi-Z	Keep-O	
P014/DA0	Hi-Z	[DA0 output (DAOE0=1)] DA output retained [Other than the above (DAOE0=0)] Keep-O	
P015/IRQ13/DA1	Hi-Z	[DA1 output (DAOE1=1)] DA output retained [Other than the above (DAOE1=0)] Keep-O*1	
P100/D00/RXD0_A/CMPIN0/KR00/IRQ2/ AGTIO0_A	Hi-Z	[D00 output] Hi-Z [AGTIO0 selected] AGTIO0 output*2 [Other than the above] Keep-O*1	
P101/D01/CMPREF0/KR01/IRQ1	Hi-Z	[D01 output] Hi-Z [Other than the above] Keep-O*1	
P102/D02/CMPIN1/KR02/AGTO0	Hi-Z	[D02 output] Hi-Z [AGTO0 selected] AGTO0 output*2 [Other than the above] Keep-O*1	
P103/D03/CMPREF1/KR03	Hi-Z	[D03 output] Hi-Z [Other than the above] Keep-O*1	
P104/D04/KR04/IRQ1	Hi-Z	[D04 output] Hi-Z [Other than the above] Keep-O*1	
P105/D05/KR05/IRQ0	Hi-Z	[D05 output] Hi-Z [Other than the above] Keep-O*1	
P106/D06/KR06	Hi-Z	[D06 output] Hi-Z [Other than the above] Keep-O*1	
P107/D07/KR07	Hi-Z	[D07 output] Hi-Z [Other than the above] Keep-O*1	
P108/TMS	Pull-up	Keep-O	

Table 1.1 Port states in each processing state (2 of 5)

Port Name	Reset	Software Standby Mode	
		OPE = 0	OPE = 1
P109/TDO/CLKOUT_B	TDO output	[CLKOUT selected] CLKOUT output [Other than the above] Keep-O	
P110/IRQ3/TDI/VCOUT	Pull-up	[ACMPLP selected] VCOUT output [Other than the above] Keep-O*1	
P111/A05/IRQ4	Hi-Z	[A05 output] Hi-Z [Other than the above] Keep-O*1	[A05 output] Address output retained [Other than the above] Keep-O*1
P112/A04	Hi-Z	[A04 output] Hi-Z [Other than the above] Keep-O	[A04 output] Address output retained [Other than the above] Keep-O
P113/A03	Hi-Z	[A03 output] Hi-Z [Other than the above] Keep-O	[A03 output] Address output retained [Other than the above] Keep-O
P114/A02	Hi-Z	[A02 output] Hi-Z [Other than the above] Keep-O	[A02 output] Address output retained [Other than the above] Keep-O
P115/A01	Hi-Z	[A01 output] Hi-Z [Other than the above] Keep-O	[A01 output] Address output retained [Other than the above] Keep-O
P200/NMI	Hi-Z	Hi-Z	
P201	Pull-up	Keep-O	
P202/WR1/BC1/IRQ3	Hi-Z	[WR1/BC1 output] Hi-Z [Other than the above] Keep-O*1	[WR1/BC1 output] H [Other than the above] Keep-O*1
P203/IRQ2	Hi-Z	Keep-O*1	
P204/SCL0_B/USB_OVRCURB/AGTIO1_A	Hi-Z	[AGTIO1 selected] AGTIO1 output*2 [Other than the above] Keep-O*1	
P205/A16/USB_OVRCURA/IRQ1/CLKOUT_A/AGTO1	Hi-Z	[A16 output] Hi-Z [CLKOUT selected] CLKOUT output [AGTO1 selected] AGTO1 output*2 [Other than the above] Keep-O*1	[A16 output] Address output retained [CLKOUT selected] CLKOUT output [AGTO1 selected] AGTO1 output*2 [Other than the above] Keep-O*1
P206/WAIT/IRQ0	Hi-Z	Keep-O*1	
P212/IRQ3/EXTAL	Hi-Z	Keep-O*1	
P213/IRQ2/XTAL	Hi-Z	Keep-O*1	
P214/XCOUT	Hi-Z	[Sub-clock Oscillator selected] Sub-clock Oscillator is operating [Other than the above] Hi-Z	
P215/XCIN	Hi-Z	[Sub-clock Oscillator selected] Sub-clock Oscillator is operating [Other than the above] Hi-Z	
P300/TCK	Pull-up	Keep-O	
P301/A06/IRQ6	Hi-Z	[A06 output] Hi-Z [Other than the above] Keep-O*1	[A06 output] Address output retained [Other than the above] Keep-O*1
P302/A07/IRQ5	Hi-Z	[A07 output] Hi-Z [Other than the above] Keep-O*1	[A07 output] Address output retained [Other than the above] Keep-O*1

Table 1.1 Port states in each processing state (3 of 5)

Port Name	Reset	Software Standby Mode	
		OPE = 0	OPE = 1
P303/A08	Hi-Z	[A08 output] Hi-Z [Other than the above] Keep-O	[A08 output] Address output retained [Other than the above] Keep-O
P304/A09/IRQ9	Hi-Z	[A09 output] Hi-Z [Other than the above] Keep-O*1	[A09 output] Address output retained [Other than the above] Keep-O*1
P305/A10/IRQ8	Hi-Z	[A10 output] Hi-Z [Other than the above] Keep-O*1	[A10 output] Address output retained [Other than the above] Keep-O*1
P306/A11	Hi-Z	[A11 output] Hi-Z [Other than the above] Keep-O	[A11 output] Address output retained [Other than the above] Keep-O
P307/A12	Hi-Z	[A12 output] Hi-Z [Other than the above] Keep-O	[A12 output] Address output retained [Other than the above] Keep-O
P308/A13	Hi-Z	[A13 output] Hi-Z [Other than the above] Keep-O	[A13 output] Address output retained [Other than the above] Keep-O
P309/A14	Hi-Z	[A14 output] Hi-Z [Other than the above] Keep-O	[A14 output] Address output retained [Other than the above] Keep-O
P310/A15	Hi-Z	[A15 output] Hi-Z [Other than the above] Keep-O	[A15 output] Address output retained [Other than the above] Keep-O
P311/CS2	Hi-Z	[CS2 output] Hi-Z [Other than the above] Keep-O	[CS2 output] H [Other than the above] Keep-O
P312/CS3	Hi-Z	[CS3 output] Hi-Z [Other than the above] Keep-O	[CS3 output] H [Other than the above] Keep-O
P313 to P315	Hi-Z	Keep-O	
P400/SCL0_A/IRQ0	Hi-Z	Keep-O*1	
P401/SDA0_A/IRQ5	Hi-Z	Keep-O*1	
P402/IRQ4/RTCIC0/AGTIO0_B/AGTIO1_B	Hi-Z	Keep-O*1	
P403/RTCIC1/AGTIO0_C/AGTIO1_C	Hi-Z	Keep-O*1	
P404/RTCIC2	Hi-Z	Keep-O*1	
P405, P406	Hi-Z	Keep-O	
P407/SDA0_B/USB_VBUS/RTCOUT	Hi-Z	[RTCOUT selected] RTCOUT output [Other than the above] Keep-O*1	
P408/IRQ7	Hi-Z	[AGTOB1 selected] AGTOB1 output*2 [Other then the above] Keep-O*1	
P409/IRQ6	Hi-Z	[AGTOB1 selected] AGTOB1 output*2 [Other then the above] Keep-O*1	
P410/RXD0_B/IRQ5/AGTOB1	Hi-Z	[AGTOB1 selected] AGTOB1 output*2 [Other then the above] Keep-O*1	

Table 1.1 Port states in each processing state (4 of 5)

Port Name	Reset	Software Standby Mode	
		OPE = 0	OPE = 1
P411/IRQ4	Hi-Z	[AGTOB1 selected] AGTOB1 output*2 [Other then the above] Keep-O*1	
P412 to P415	Hi-Z	Keep-O	
P500/AGTOA0	Hi-Z	[AGTOA0 selected] AGTOA0 output*2 [Other than the above] Keep-O*1	
P501/USB_OVRCURA/IRQ11/AGTOB0	Hi-Z	[AGTOB0 selected] AGTOB0 output*2 [Other then the above] Keep-O*1	
P502/USB_OVRCURB/IRQ12	Hi-Z	[AGTOB0 selected] AGTOB0 output*2 [Other then the above] Keep-O*1	
P503, P504	Hi-Z	Keep-O	
P505/IRQ14	Hi-Z	Keep-O*1	
P506/IRQ15	Hi-Z	Keep-O*1	
P507	Hi-Z	Keep-O	
P511/IRQ15	Hi-Z	Keep-O*1	
P512/IRQ14	Hi-Z	Keep-O*1	
P600/RD	Hi-Z	[RD output] Hi-Z [Other than the above] Keep-O	[RD output] H [Other than the above] Keep-O
P601/WR0/WR	Hi-Z	[WR0/WR output] Hi-Z [Other than the above] Keep-O	[WR0/WR output] H [Other than the above] Keep-O
P602/EBCLK	Hi-Z	[EBCLK output] H [Other than the above] Keep-O	
P603/D13	Hi-Z	[D13 output] Hi-Z [Other than the above] Keep-O	
P604/D12	Hi-Z	[D12 output] Hi-Z [Other than the above] Keep-O	
P605/D11	Hi-Z	[D11 output] Hi-Z [Other than the above] Keep-O	
P606	Hi-Z	Keep-O	
P608/A00/BC0	Hi-Z	[A00 output] Hi-Z [BC0 output] Hi-Z [Other than the above] Keep-O	[A00 output] Address output retained [BC0 output] H [Other than the above] Keep-O
P609/CS1	Hi-Z	[CS1 output] Hi-Z [Other than the above] Keep-O	[CS1 output] H [Other than the above] Keep-O
P610/CS0	Hi-Z	[CS0 output] Hi-Z [Other than the above] Keep-O	[CS0 output] H [Other than the above] Keep-O
P611	Hi-Z	Keep-O	

Table 1.1 Port states in each processing state (5 of 5)

Port Name	Reset	Software Standby Mode	
		OPE = 0	OPE = 1
P612/D8	Hi-Z	[D8 output] Hi-Z [Other than the above] Keep-O	
P613/D9	Hi-Z	[D9 output] Hi-Z [Other than the above] Keep-O	
P614/D10	Hi-Z	[D10 output] Hi-Z [Other than the above] Keep-O	
P700 to P705	Hi-Z	Keep-O	
P708/IRQ11	Hi-Z	Keep-O*1	
P709/IRQ10	Hi-Z	Keep-O*1	
P710 to P713	Hi-Z	Keep-O	
P800/D14	Hi-Z	[D14 output] Hi-Z [Other than the above] Keep-O	
P801/D15	Hi-Z	[D15 output] Hi-Z [Other than the above] Keep-O	
P802 to P809	Hi-Z	Keep-O	
P900 to P902	Hi-Z	Keep-O	
USB_DP	Hi-Z	Keep-O	
USB_DM	Hi-Z	Keep-O	

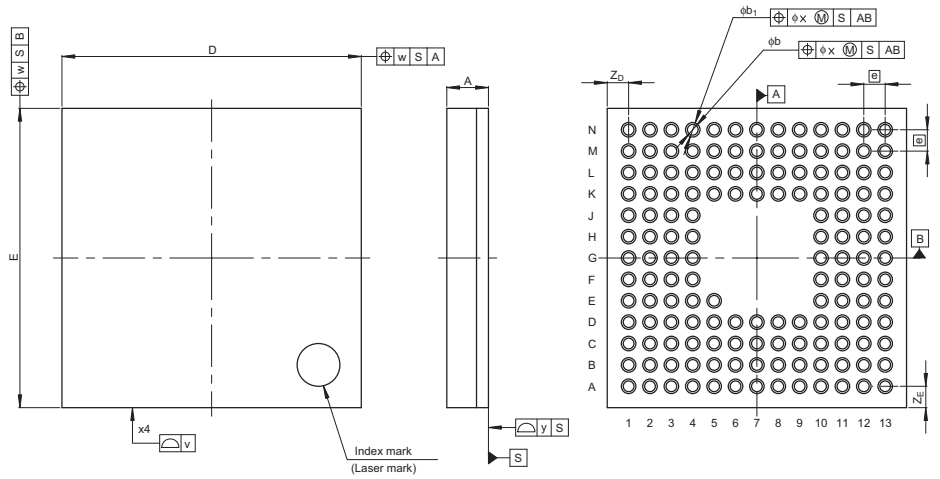
H: High-level
Hi-Z: High-impedance
Keep-O: Output pins retain their previous values. Input pins become high-impedance.

Note: The LCD output is retained when the LCD controller/driver pin functions (COM0 to COM7 and SEG00 to SEG51) are set and LOCO or SOSC is selected in the SLCDSCKCR.LCDSCKSEL[2:0] bits.
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 or SOSC 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.]
P-TFLGA145-7x7-0.50	PTLG0145KA-A	145F0G	0.1g

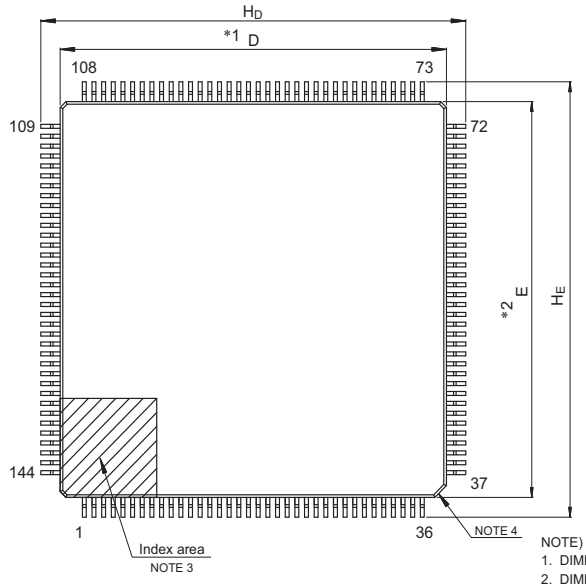


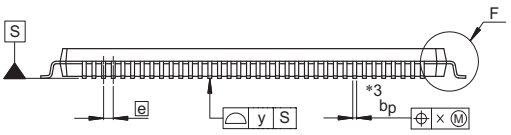
Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	7.0	—
E	—	7.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.05
e	—	0.5	—
b	0.21	0.25	0.29
b_1	0.29	0.34	0.39
x	—	—	0.08
y	—	—	0.08
Z_D	—	0.5	—
Z_E	—	0.5	—

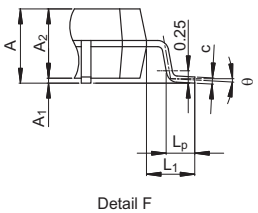
Figure 2.1 LGA 145-pin

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP144-20x20-0.50	PLQP0144KA-B	—	1.2

Unit: mm







Detail F

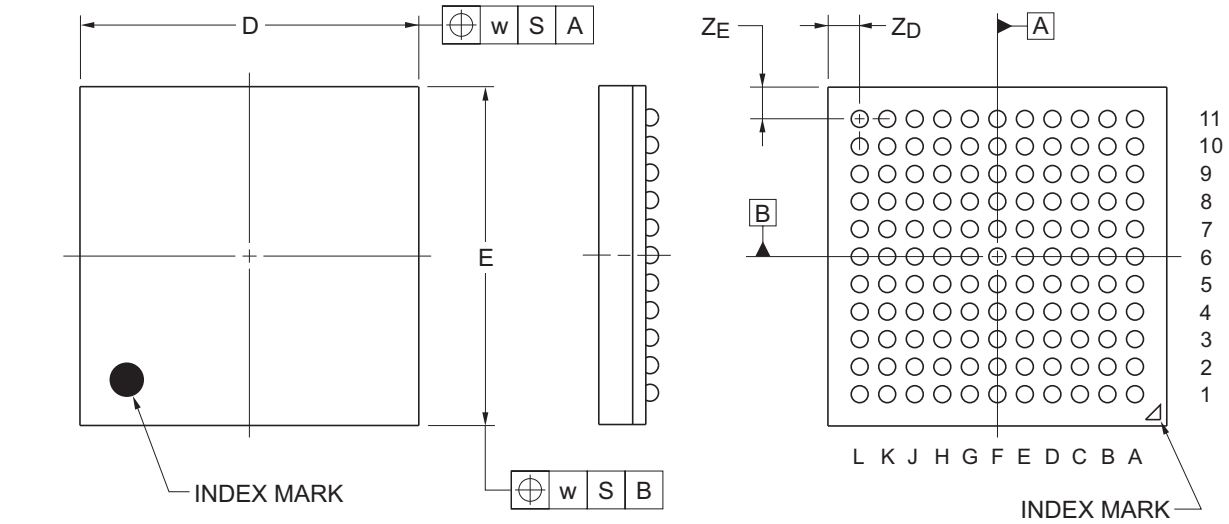
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

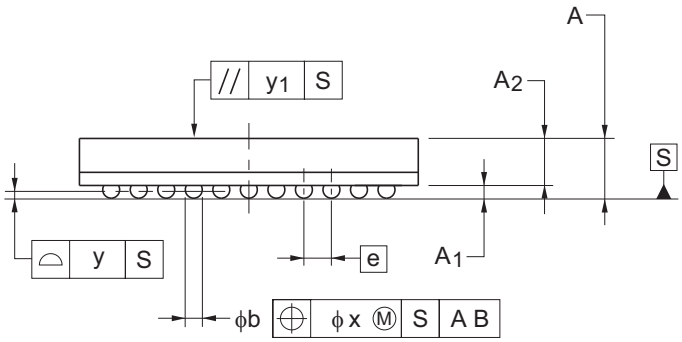
Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	19.9	20.0	20.1
E	19.9	20.0	20.1
A_2	—	1.4	—
H_D	21.8	22.0	22.2
H_E	21.8	22.0	22.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.17	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
L_p	0.45	0.6	0.75
L_1	—	1.0	—

Figure 2.2 LQFP 144-pin

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFBGA121-8x8-0.65	PLBG0121JA-A	—	0.15

Unit: m

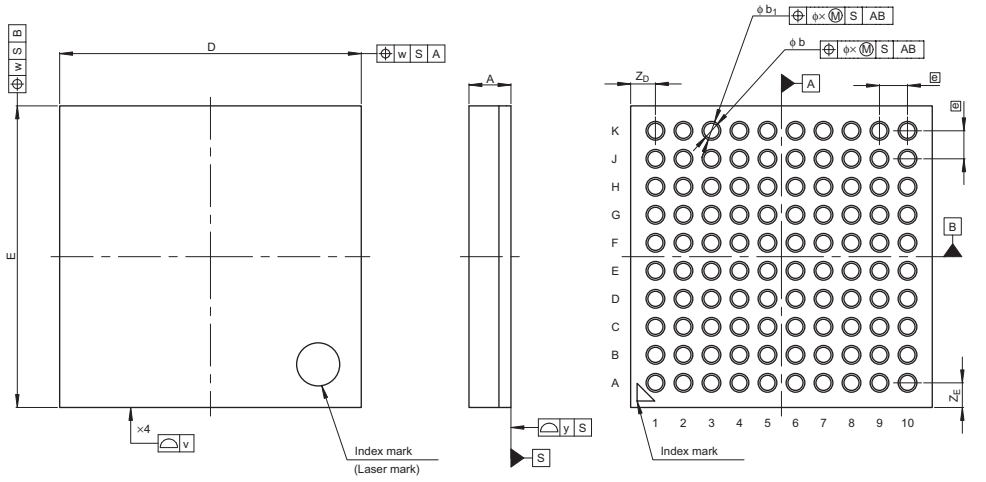




Reference Symbol	Dimensions in millime		
	Min	Nom	M
D	7.90	8.00	8.
E	7.90	8.00	8.
w	—	0.20	–
A	1.11	1.21	1.
A_1	0.25	0.30	0.
A_2	—	0.91	–

Figure 2.3 BGA 121-pin

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-TFLGA100-7x7-0.65	PTLG0100JA-A	100F0G	0.1g

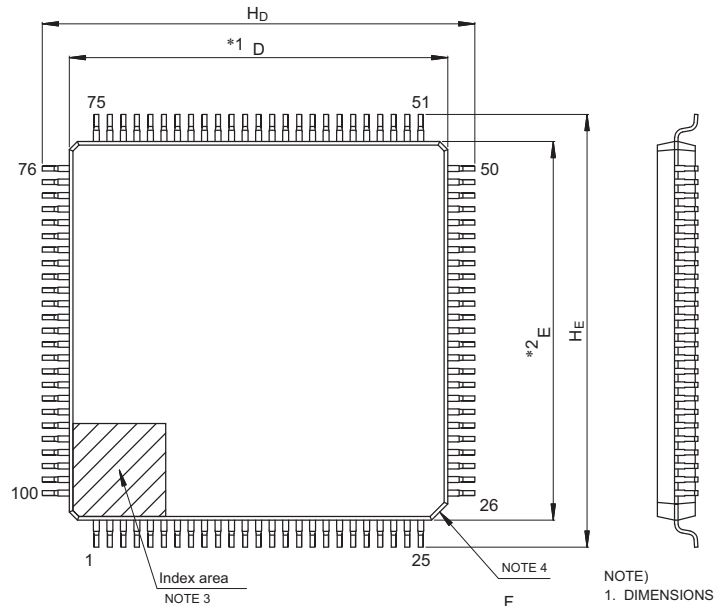


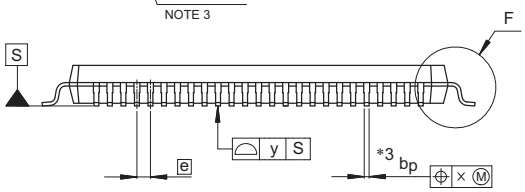
Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	7.0	—
E	—	7.0	—
v	—	—	0.15
w	—	—	0.20
A	—	—	1.05
e	—	0.65	—
b	0.31	0.35	0.39
b_1	0.385	0.435	0.485
x	—	—	0.08
y	—	—	0.10
Z_D	—	0.575	—
Z_E	—	0.575	—

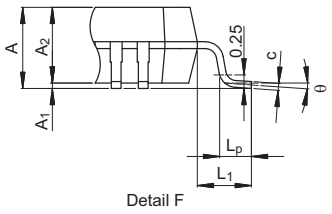
Figure 2.4 LGA 100-pin

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP100-14x14-0.50	PLQP0100KB-B	—	0.6

Unit: mm







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.

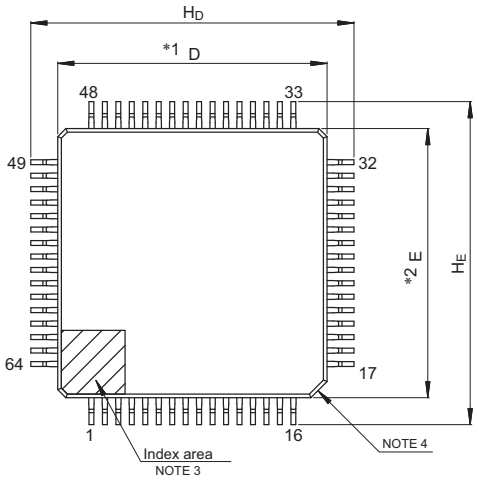
Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

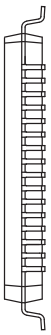
© 2015 Renesas Electronics Corporation. All rights reserved.

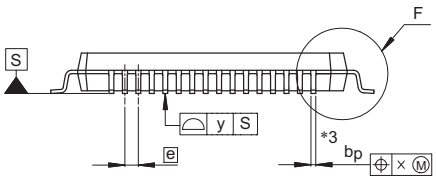
Figure 2.5 LQFP 100-pin

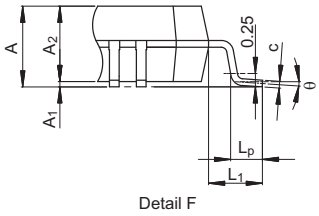
JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP64-10x10-0.50	PLQP0064KB-C	—	0.3

Unit: mm









Detail F

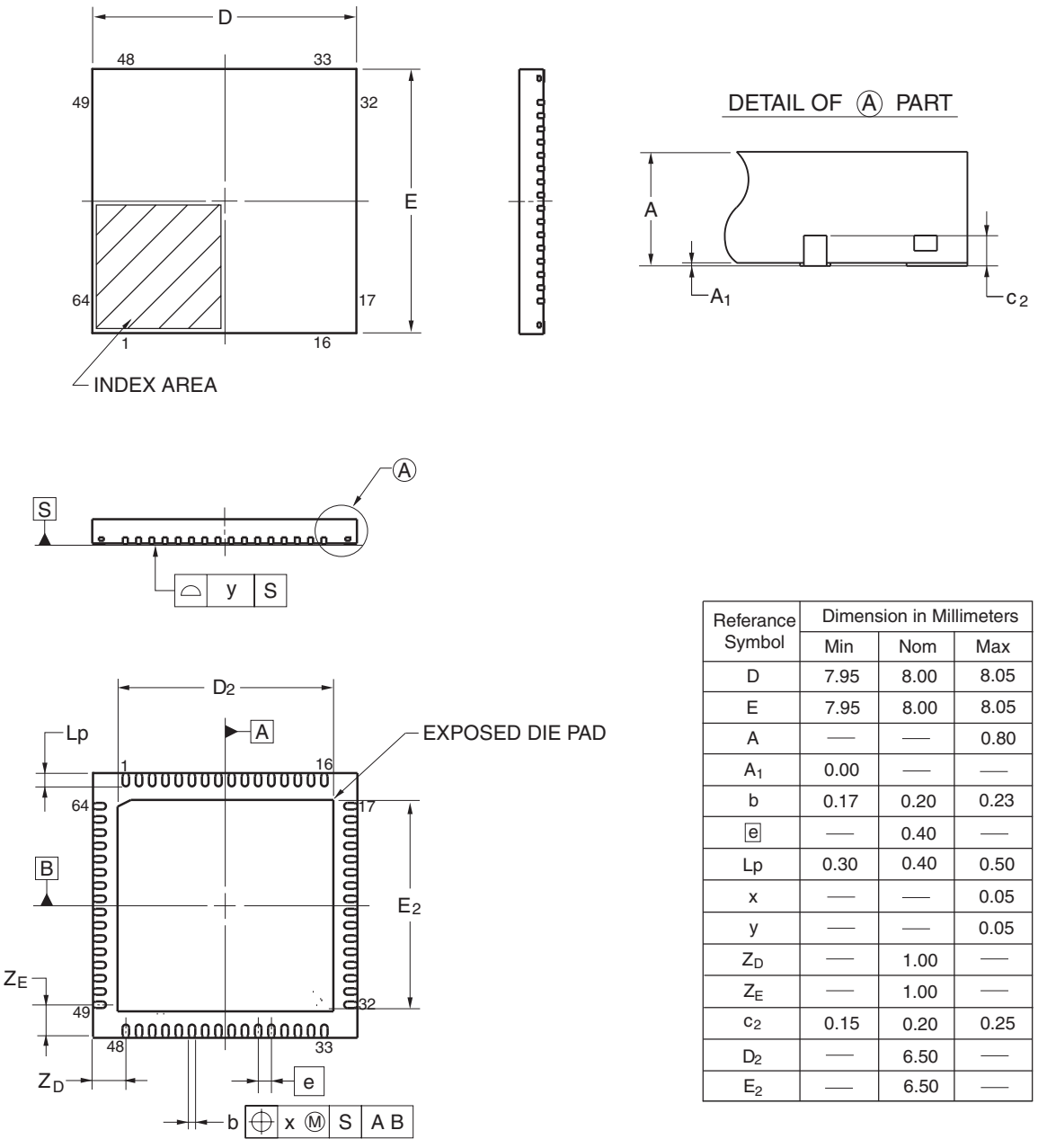
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A_2	—	1.4	—
H_D	11.8	12.0	12.2
H_E	11.8	12.0	12.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	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 2.6 LQFP 64-pin

JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-HWQFN64-8x8-0.40	PWQN0064LA-A	P64K8-40-9B5-3	0.16



Referance Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	7.95	8.00	8.05
E	7.95	8.00	8.05
A	—	—	0.80
A_1	0.00	—	—
b	0.17	0.20	0.23
e	—	0.40	—
Lp	0.30	0.40	0.50
x	—	—	0.05
y	—	—	0.05
Z_D	—	1.00	—
Z_E	—	1.00	—
c_2	0.15	0.20	0.25
D_2	—	6.50	—
E_2	—	6.50	—

©2013 Renesas Electronics Corporation. All rights reserved.

Figure 2.7 QFN 64-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 3.1 shows the name, description, and base address of each peripheral.

Table 3.1 Peripheral base address (1 of 2)

Name	Description	Base address
MMPU	Bus Master MPU	0x40000000
SMPU	Bus Slave MPU	0x40000C00
SPMON	CPU Stack Pointer Monitor	0x40000D00
MMF	Memory Mirror Function	0x40001000
SRAM	SRAM Control	0x40002000
BUS	BUS Control	0x40003000
DMAC0	Direct Memory Access Controller 0	0x40005000
DMAC1	Direct Memory Access Controller 1	0x40005040
DMAC2	Direct Memory Access Controller 2	0x40005080
DMAC3	Direct Memory Access Controller 3	0x400050C0
DMA	DMAC Module Activation	0x40005200
DTC	Data Transfer Controller	0x40005400
ICU	Interrupt Controller	0x40006000
DBG	Debug Function	0x4001B000
FCACHE	Flash Cache	0x4001C000
SYSTEM	System Control	0x4001E000
PORT0	Port 0 Control Registers	0x40040000
PORT1	Port 1 Control Registers	0x40040020
PORT2	Port 2 Control Registers	0x40040040
PORT3	Port 3 Control Registers	0x40040060
PORT4	Port 4 Control Registers	0x40040080
PORT5	Port 5 Control Registers	0x400400A0
PORT6	Port 6 Control Registers	0x400400C0
PORT7	Port 7 Control Registers	0x400400E0
PORT8	Port 8 Control Registers	0x40040100
PORT9	Port 9 Control Registers	0x40040120
PFS	Pmn Pin Function Control Register	0x40040800
PMISC	Miscellaneous Port Control Register	0x40040D00
ELC	Event Link Controller	0x40041000
POEG	Port Output Enable Module for GPT	0x40042000
RTC	Realtime Clock	0x40044000
WDT	Watchdog Timer	0x40044200
IWDT	Independent Watchdog Timer	0x40044400
CAC	Clock Frequency Accuracy Measurement Circuit	0x40044600
MSTP	Module Stop Control B,C,D	0x40047000
SSI0	Serial Sound Interface 0	0x4004E000
SSI1	Serial Sound Interface 1	0x4004E100
CAN0	CAN0 Module	0x40050000

Table 3.1 Peripheral base address (2 of 2)

Name	Description	Base address
IIC0	Inter-Integrated Circuit 0	0x40053000
IIC1	Inter-Integrated Circuit 1	0x40053100
IIC2	Inter-Integrated Circuit 2	0x40053200
DOC	Data Operation Circuit	0x40054100
ADC140	14-bit A/D Converter	0x4005C000
DAC12	12-bit D/A converter	0x4005E000
SDHI0	SD Host Interface 0	0x40062000
SCI0	Serial Communication Interface 0	0x40070000
SCI1	Serial Communication Interface 1	0x40070020
SCI2	Serial Communication Interface 2	0x40070040
SCI3	Serial Communication Interface 3	0x40070060
SCI4	Serial Communication Interface 4	0x40070080
SCI9	Serial Communication Interface 9	0x40070120
IRDA	Infrared Data Association	0x40070F00
SPI0	Serial Peripheral Interface 0	0x40072000
SPI1	Serial Peripheral Interface 1	0x40072100
CRC	CRC Calculator	0x40074000
GPT320	General PWM Timer 0 (32-bit)	0x40078000
GPT321	General PWM Timer 1 (32-bit)	0x40078100
GPT322	General PWM Timer 2 (32-bit)	0x40078200
GPT323	General PWM Timer 3 (32-bit)	0x40078300
GPT324	General PWM Timer 4 (32-bit)	0x40078400
GPT325	General PWM Timer 5 (32-bit)	0x40078500
GPT326	General PWM Timer 6 (32-bit)	0x40078600
GPT327	General PWM Timer 7 (32-bit)	0x40078700
GPT328	General PWM Timer 8 (32-bit)	0x40078800
GPT329	General PWM Timer 9 (32-bit)	0x40078900
GPT_OPS	Output Phase Switching Controller	0x40078FF0
KINT	Key Interrupt Function	0x40080000
CTSU	Capacitive Touch Sensing Unit	0x40081000
SLCDC	Segment LCD Controller/Driver	0x40082000
AGT0	Asynchronous General purpose Timer 0	0x40084000
AGT1	Asynchronous General purpose Timer 1	0x40084100
ACMPHS0	High-Speed Analog Comparator 0	0x40085000
ACMPHS1	High-Speed Analog Comparator 1	0x40085100
ACMPLP	Low-Power Analog Comparator	0x40085E00
OPAMP	Operational Amplifier	0x40086000
USBFS	USB 2.0 Full-Speed Module	0x40090000
TSN	Temperature Sensor	0x407EC000
QSPI	Quad-SPI	0x64000000

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 3.2 and Table 3.3:

- 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 masters such as DTC or DMAC.

Table 3.2 shows the register access cycles for non-GPT modules.

Table 3.2 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		
MMPU, SMPU, SPMON, MMF, SRAM, BUS, DMACn, DMA, DTC, ICU, DBG, FCACHE	4000 0000h	4001 CFFFh	2				ICLK	Memory Protection Unit, Memory Mirror Function, SRAM, Buses, DMA Controller, Data Transfer Controller, Interrupt Controller, CPU, Flash Memory
SYSTEM	4001 E000h	4001 E3FFh	3				ICLK	Low Power Modes, Resets, Low Voltage Detection, Clock Generation Circuit, Register Write Protection
SYSTEM	4001 E400h	4001 E6FFh	7		5 to 7		PCLKB	Low Power Modes, Resets, Low Voltage Detection, Battery Backup Function
PORTn, PFS, PMISC, ELC, POEG, RTC, WDT, IWDT, CAC, MSTP	4004 0000h	4004 7FFFh	3		2 to 3		PCLKB	I/O Ports, Event Link Controller, Port Output Enable for GPT, Realtime Clock, Watchdog Timer, Independent Watchdog Timer, Clock Frequency Accuracy Measurement Circuit, Module Stop Control
SSIn, CAN0, IICn, DOC, ADC140, DAC12	4004 E000h	4005 EFFFh	3		2 to 3		PCLKB	Serial Sound Interface, Controller Area Network Module, I^2C Bus Interface, Data Operation Circuit, 14-Bit A/D Converter, 12-Bit D/A Converter
SDHI0	4006 2000h	4006 2FFFh	3		2 to 3		PCLKA	SD/MMC Host Interface
SCIn	4007 0000h	4007 0EFFh	5*2		2 to 3*2		PCLKA	Serial Communications Interface
IRDA	4007 0F00h	4007 0FFFh	5		2 to 3		PCLKA	IrDA Interface
SPIn	4007 2000h	4007 2FFFh	5*3		2 to 3*3		PCLKA	Serial Peripheral Interface
CRC	4007 4000h	4007 4FFFh	3		2 to 3		PCLKA	CRC Calculator

Table 3.2 Access cycles for non-GPT modules (2 of 2)

Peripherals	Address		Number of access cycles					
			ICLK = PCLK		ICLK > PCLK*1		Cycle unit	Related function
	From	To	Read	Write	Read	Write		
GPT32n, GPT_OPS	4007 8000h	4007 8FFFh	Refer to Table 3*4				PCLKA	General PWM Timer
TSN	407E C000h	407E CFFFh	7		7		ICLK	Temperature Sensor
KINT, CTSU, SLCDC	4008 0000h	4008 1FFFh	2		1 to 2		PCLKB	Key interrupt Function, Capacitive Touch Sensing Unit, Segment LCD Controller
AGTn	4008 4000h	4008 4FFFh	3		2 to 3		PCLKB	Asynchronous General Purpose Timer
ACMPHSn, ACMPLP, OPAMP	4008 5000h	4008 6FFFh	2		1 to 2		PCLKB	High-Speed Analog Comparator, Low-Power Analog Comparator, Operational Amplifier
USBFS	4009 0000h	4009 03FFh	4		3 to 4		PCLKB	USB 2.0 Full-Speed Module
USBFS	4009 0400h	4009 04FFh	3		2 to 3		PCLKB	USB 2.0 Full-Speed Module
QSPI	6400 0000h	6400 000Fh	4	13 to *5	2 to 3	12 to *5	PCLKA	Quad Serial Peripheral Interface
QSPI	6400 0010h	6400 0013h	24 to *5	5 to *5	23 to *5	4 to *5	PCLKA	Quad Serial Peripheral Interface
QSPI	6400 0014h	6400 0037h	3	13 to *5	2 to 3	12 to *5	PCLKA	Quad Serial Peripheral Interface
QSPI	6400 0804h	6400 0807h	2	2	2 to 3	2 to 3	PCLKA	Quad Serial Peripheral Interface

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. When accessing a 16-bit register (FTDRHL, FRDRHL, FCR, FDR, LSR, and CDR), access is 2 cycles more than the value shown in Table 3.2. When accessing an 8-bit register (FTDRH, FTDRL, FRDRH, and FRDRL), the access cycles are as shown in Table 3.2.

Note 3. When accessing the 32-bit register (SPDR), access is 2 cycles more than the value in Table 3.2. When accessing an 8-bit or 16-bit register (SPDR_HA), the access cycles are as shown in Table 3.2.

Note 4. The access cycles differ depending on the frequency ratio between ICLK, PCLKA, and PCLKD, as shown in Table 3.3.

Note 5. The access cycles depend on the QSPI bus cycles.

Table 3.3 shows register access cycles for GPT modules.

Table 3.3 Access cycles for GPT modules

Frequency ratio between ICLK and PCLK	Number of access cycles		Cycle unit
	Read	Write	
ICLK > PCLKD = PCLKA	5 to 6	3 to 4	PCLKA
ICLK > PCLKD > PCLKA	3 to 4	2 to 3	PCLKA
PCLKD = ICLK = PCLKA	6	4	PCLKA
PCLKD = ICLK > PCLKA	2 to 3	1 to 2	PCLKA
PCLKD > ICLK = PCLKA	4	3	PCLKA
PCLKD > ICLK > PCLKA	2 to 3	1 to 2	PCLKA

3.3 Register Descriptions

This section provides information associated with registers described in this manual.

Table 3.4 shows a list of registers including address offsets, address sizes, access rights, and reset values.

Table 3.4 Register description (1 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
MMPU	-	-	-	MMPUCTLA	Bus Master MPU Control Register A	0x000	16	read/ write	0x0000	0xFFFF
				MMPUPTA	Group A Protection of Register	0x102	16	read/ write	0x0000	0xFFFF
	16	0x010	0-15	MMPUACA%s	Group A Region %s Access Control Register	0x200	16	read/ write	0x0000	0xFFFF
	16	0x010	0-15	MMPUSA%s	Group A Region %s Start Address Register	0x204	32	read/ write	0x00000000	0x00000003
	16	0x010	0-15	MMPUEA%s	Group A Region %s End Address Register	0x208	32	read/ write	0x00000003	0x00000003
SMPU	-	-	-	SMPUCTL	Slave MPU Control Register	0x00	16	read/ write	0x0000	0xFFFF
				SMPUMBIU	Access Control Register for MBIU	0x10	16	read/ write	0x0000	0xFFFF
				SMPUFBIU	Access Control Register for FBIU	0x14	16	read/ write	0x0000	0xFFFF
	2	0x4	0,1	SMPUSRAM%s	Access Control Register for SRAM%s	0x18	16	read/ write	0x0000	0xFFFF
	3	0x4	0,2,6	SMPUP%sBIU	Access Control Register for P%sBIU	0x20	16	read/ write	0x0000	0xFFFF
	-	-	-	SMPUEXBIU	Access Control Register for EXBIU	0x30	16	read/ write	0x0000	0xFFFF
				SMPUEXBIU2	Access Control Register for EXBIU2	0x34	16	read/ write	0x0000	0xFFFF
SPMON	-	-	-	MSPMPUOAD	Stack Pointer Monitor Operation After Detection Register	0x00	16	read/ write	0x0000	0xFFFF
				MSPMPUCTL	Stack Pointer Monitor Access Control Register	0x04	16	read/ write	0x0000	0xFEFF
				MSPMPUPT	Stack Pointer Monitor Protection Register	0x06	16	read/ write	0x0000	0xFFFF
				MSPMPUSA	Main Stack Pointer (MSP) Monitor Start Address Register	0x08	32	read/ write	0x00000000	0x00000003
				MSPMPUEA	Main Stack Pointer (MSP) Monitor End Address Register	0x0C	32	read/ write	0x00000003	0x00000003
				PSPMPUOAD	Stack Pointer Monitor Operation After Detection Register	0x10	16	read/ write	0x0000	0xFFFF
SPMON	-	-	-	PSPMPUCTL	Stack Pointer Monitor Access Control Register	0x14	16	read/ write	0x0000	0xFEFF
				PSPMPUPT	Stack Pointer Monitor Protection Register	0x16	16	read/ write	0x0000	0xFFFF
				PSPMPUSA	Process Stack Pointer (PSP) Monitor Start Address Register	0x18	32	read/ write	0x00000000	0x00000003
				PSPMPUEA	Process Stack Pointer (PSP) Monitor End Address Register	0x1C	32	read/ write	0x00000003	0x00000003

Table 3.4 Register description (2 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
MMF	-	-	-	MMSFR	MemMirror Special Function Register	0x00	32	read/ write	0x00000000	0xFFFFFFFF
				MMEN	MemMirror Enable Register	0x04	32	read/ write	0x00000000	0xFFFFFFFF
SRAM	-	-	-	PARIOAD	SRAM Parity Error Operation After Detection Register	0x00	8	read/ write	0x00	0xFF
				SRAMPRCR	SRAM Protection Register	0x04	8	read/ write	0x00	0xFF
				ECCMODE	ECC Operating Mode Control Register	0xC0	8	read/ write	0x00	0xFF
				ECC2STS	ECC 2-Bit Error Status Register	0xC1	8	read/ write	0x00	0xFF
				ECC1STSEN	ECC 1-Bit Error Information Update Enable Register	0xC2	8	read/ write	0x00	0xFF
				ECC1STS	ECC 1-Bit Error Status Register	0xC3	8	read/ write	0x00	0xFF
				ECCPRCR	ECC Protection Register	0xC4	8	read/ write	0x00	0xFF
				ECCPRCR2	ECC Protection Register 2	0xD0	8	read/ write	0x00	0xFF
				ECCETST	ECC Test Control Register	0xD4	8	read/ write	0x00	0xFF
				ECCOAD	SRAM ECC Error Operation After Detection Register	0xD8	8	read/ write	0x00	0xFF
BUS	4	0x10	0-3	CS%sMOD	CS%s Mode Register	0x0002	16	read/ write	0x0000	0xFFFF
	4	0x10	0-3	CS%sWCR1	CS%s Wait Control Register 1	0x0004	32	read/ write	0x07070707	0xFFFFFFFF
	4	0x10	0-3	CS%sWCR2	CS%s Wait Control Register 2	0x0008	32	read/ write	0x00000007	0xFFFFFFFF
	-	-	-	CS0CR	CS0 Control Register	0x0802	16	read/ write	0x0021	0xFFFF
	4	0x10	0-3	CS%sREC	CS%s Recovery Cycle Register	0x080A	16	read/ write	0x0000	0xFFFF
	3	0x10	1-3	CS%sCR	CS%s Control Register	0x0812	16	read/ write	0x0000	0xFFFF
BUS	-	-	-	CSRECEN	CS Recovery Cycle Insertion Enable Register	0x0880	16	read/ write	0x3E3E	0xFFFF
	4	0x4	M4I,M4D,SYS,DMA	BUSMCNT%s	Master Bus Control Register	0x1000	16	read/ write	0x0000	0xFFFF
	-	-	-	BUSSCNTFLI	Slave Bus Control Register FLI	0x1100	16	read/ write	0x0000	0xFFFF
				BUSSCNTMBIU	Slave Bus Control Register MBIU	0x1108	16	read/ write	0x0000	0xFFFF
	2	0x4	RAM0,RAM1	BUSSCNT%s	Slave Bus Control Register %s	0x110C	16	read/ write	0x0000	0xFFFF
	4	0x4	P0B,P2B,P3B,P4B	BUSSCNT%s	Slave Bus Control Register %s	0x1114	16	read/ write	0x0000	0xFFFF
	-	-	-	BUSSCNTP6B	Slave Bus Control Register P6B	0x1128	16	read/ write	0x0000	0xFFFF
	3	0x4	FBU,EXT,EXT2	BUSSCNT%s	Slave Bus Control Register %s	0x1130	16	read/ write	0x0000	0xFFFF

Table 3.4 Register description (3 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
BUS	4	0x10	1-4	BUS%sERRADD	Bus Error Address Register %s	0x1800	32	read-only	0x00000000	0x00000000
	4	0x10	1-4	BUS%sERRSTAT	Bus Error Status Register %s	0x1804	8	read-only	0x00	0xFE
DMAC0-3	-	-	-	DMSAR	DMA Source Address Register	0x00	32	read/write	0x00000000	0xFFFFFFFF
				DMDAR	DMA Destination Address Register	0x04	32	read/write	0x00000000	0xFFFFFFFF
				DMCRA	DMA Transfer Count Register	0x08	32	read/write	0x00000000	0xFFFFFFFF
				DMCRB	DMA Block Transfer Count Register	0x0C	16	read/write	0x0000	0xFFFF
				DMTMD	DMA Transfer Mode Register	0x10	16	read/write	0x0000	0xFFFF
				DMINT	DMA Interrupt Setting Register	0x13	8	read/write	0x00	0xFF
				DMAMD	DMA Address Mode Register	0x14	16	read/write	0x0000	0xFFFF
				DMOFR	DMA Offset Register	0x18	32	read/write	0x00000000	0xFFFFFFFF
				DMCNT	DMA Transfer Enable Register	0x1C	8	read/write	0x00	0xFF
				DMREQ	DMA Software Start Register	0x1D	8	read/write	0x00	0xFF
				DMSTS	DMA Status Register	0x1E	8	read/write	0x00	0xFF
DMA	-	-	-	DMAST	DMAC Module Activation Register	0x00	8	read/write	0x00	0xFF
DTC	-	-	-	DTCCR	DTC Control Register	0x00	8	read/write	0x08	0xFF
DTC	-	-	-	DTCVBR	DTC Vector Base Register	0x04	32	read/write	0x00000000	0xFFFFFFFF
				DTCST	DTC Module Start Register	0x0C	8	read/write	0x00	0xFF
				DTCSTS	DTC Status Register	0x0E	16	read-only	0x0000	0xFFFF
ICU	16	0x1	0-15	IRQCR%s	IRQ Control Register %s	0x000	8	read/write	0x00	0xFF
	-	-	-	NMICR	NMI Pin Interrupt Control Register	0x100	8	read/write	0x00	0xFF
				NMIER	Non-Maskable Interrupt Enable Register	0x120	16	read/write	0x0000	0xFFFF
				NMICLR	Non-Maskable Interrupt Status Clear Register	0x130	16	read/write	0x0000	0xFFFF
				NMISR	Non-Maskable Interrupt Status Register	0x140	16	read-only	0x0000	0xFFFF
				WUPEN	Wake Up Interrupt Enable Register	0x1A0	32	read/write	0x00000000	0xFFFFFFFF
				SELSR0	SYS Event Link Setting Register	0x200	16	read/write	0x0000	0xFFFF
	4	0x4	0-3	DELSR%s	DMAC Event Link Setting Register %s	0x280	16	read/write	0x0000	0xFFFF
	64	0x4	0-63	IELSR%s	ICU Event Link Setting Register %s	0x300	32	read/write	0x00000000	0xFFFFFFFF

Table 3.4 Register description (4 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
DBG	-	-	-	DBGSTR	Debug Status Register	0x000	32	read-only	0x00000000	0xFFFFFFFF
				DBGSTOPCR	Debug Stop Control Register	0x010	32	read/write	0x00000003	0xFFFFFFFF
				TRACECTR	Trace Control Register	0x020	32	read/write	0x00000000	0xFFFFFFFF
FCACHE	-	-	-	FCACHEE	Flash Cache Enable Register	0x100	16	read/write	0x0000	0xFFFF
				FCACHEIV	Flash Cache Invalidate Register	0x104	16	read/write	0x0000	0xFFFF
SYSTEM	-	-	-	SBYCR	Standby Control Register	0x00C	16	read/write	0x4000	0xFFFF
				MSTPCRA	Module Stop Control Register A	0x01C	32	read/write	0xFFBFFFBC	0xFFFFFFFF
				SCKDIVCR	System Clock Division Control Register	0x020	32	read/write	0x44044444	0xFFFFFFFF
				SCKSCR	System Clock Source Control Register	0x026	8	read/write	0x01	0xFF
				PLLCR	PLL Control Register	0x02A	8	read/write	0x01	0xFF
				PLLCCR2	PLL Clock Control Register2	0x02B	8	read/write	0x07	0xFF
				BCKCR	External Bus Clock Control Register	0x030	8	read/write	0x00	0xFF
				MEMWAIT	Memory Wait Cycle Control Register	0x031	8	read/write	0x00	0xFF
				MOSCCR	Main Clock Oscillator Control Register	0x032	8	read/write	0x01	0xFF
				HOCOCR	High-Speed On-Chip Oscillator Control Register	0x036	8	read/write	0x00	0xFE
				MOCOCR	Middle-Speed On-Chip Oscillator Control Register	0x038	8	read/write	0x00	0xFF
				OSCSF	Oscillation Stabilization Flag Register	0x03C	8	read-only	0x00	0xFE
				CKOCR	Clock Out Control Register	0x03E	8	read/write	0x00	0xFF
				TRCKCR	Trace Clock Control Register	0x03F	8	read/write	0x01	0xFF
				OSTDCR	Oscillation Stop Detection Control Register	0x040	8	read/write	0x00	0xFF
				OSTDSR	Oscillation Stop Detection Status Register	0x041	8	read/write	0x00	0xFF
				SLCDSCKCR	Segment LCD Source Clock Control Register	0x050	8	read/write	0x00	0xFF
				EBCKOCR	External Bus Clock Output Control Register	0x052	8	read/write	0x00	0xFF
				MOCOUTCR	MOCO User Trimming Control Register	0x061	8	read/write	0x00	0xFF
				HOCOUTCR	HOCO User Trimming Control Register	0x062	8	read/write	0x00	0xFF
				SNZCR	Snooze Control Register	0x092	8	read/write	0x00	0xFF

Table 3.4 Register description (5 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SYSTEM	-	-	-	SNZEDCR	Snooze End Control Register	0x094	8	read/ write	0x00	0xFF
				SNZREQCR	Snooze Request Control Register	0x098	32	read/ write	0x00000000	0xFFFFFFFF
				FLSTOP	Flash Operation Control Register	0x09E	8	read/ write	0x00	0xFF
				PSMCR	Power Save Memory Control Register	0x09F	8	read/ write	0x00	0xFF
				OPCCR	Operating Power Control Register	0x0A0	8	read/ write	0x02	0xFF
				MOSCWTCR	Main Clock Oscillator Wait Control Register	0x0A2	8	read/ write	0x05	0xFF
				HOCOWTCR	High-Speed On-Chip Oscillator Wait Control Register	0x0A5	8	read/ write	0x05	0xFF
				SOPCCR	Sub Operating Power Control Register	0x0AA	8	read/ write	0x00	0xFF
				RSTSR1	Reset Status Register 1	0x0C0	16	read/ write	0x0000	0xE0F8
	2	0x2	1,2	LVD%sCR1	Voltage Monitor %s Circuit Control Register 1	0x0E0	8	read/ write	0x01	0xFF
	2	0x2	1,2	LVD%sSR	Voltage Monitor %s Circuit Status Register	0x0E1	8	read/ write	0x02	0xFF
	-	-	-	PRCR	Protect Register	0x3FE	16	read/ write	0x0000	0xFFFF
				SYOCDCR	System Control OCD Control Register	0x40E	8	read/ write	0x00	0xFF
				RSTSR0	Reset Status Register 0	0x410	8	read/ write	0x00	0xF0
				RSTSR2	Reset Status Register 2	0x411	8	read/ write	0x00	0xFE
				MOMCR	Main Clock Oscillator Mode Oscillation Control Register	0x413	8	read/ write	0x00	0xFF
				LVCMPCR	Voltage Monitor Circuit Control Register	0x417	8	read/ write	0x00	0xFF
				LVDLVLR	Voltage Detection Level Select Register	0x418	8	read/ write	0x07	0xFF
	2	0x1	1,2	LVD%sCR0	Voltage Monitor %s Circuit Control Register 0	0x41A	8	read/ write	0x80	0xF7
	-	-	-	VBTCR1	VBATT Control Register1	0x41F	8	read/ write	0x00	0xFF
				SOSCCR	Sub-Clock Oscillator Control Register	0x480	8	read/ write	0x01	0xFF
				SOMCR	Sub Clock Oscillator Mode Control Register	0x481	8	read/ write	0x00	0xFF
				LOCOCR	Low-Speed On-Chip Oscillator Control Register	0x490	8	read/ write	0x00	0xFF
				LOCOUTCR	LOCO User Trimming Control Register	0x492	8	read/ write	0x00	0xFF
				VBTCR2	VBATT Control Register2	0x4B0	8	read/ write	0x00	0xFF
				VBTSR	VBATT Status Register	0x4B1	8	read/ write	0x01	0xEC

Table 3.4 Register description (6 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SYSTEM	-	-	-	VBTCMPCR	VBATT Comparator Control Register	0x4B2	8	read/ write	0x00	0xFF
				VBTLVDICR	VBATT Pin Low Voltage Detect Interrupt Control Register	0x4B4	8	read/ write	0x00	0xFF
				VBTWCTLR	VBATT Wakeup function Control Register	0x4B6	8	read/ write	0x00	0xFF
				VBTWCH0OTSR	VBATT Wakeup I/O 0 Output Trigger Select Register	0x4B8	8	read/ write	0x00	0xFF
				VBTWCH1OTSR	VBATT Wakeup I/O 1 Output Trigger Select Register	0x4B9	8	read/ write	0x00	0xFF
				VBTWCH2OTSR	VBATT Wakeup I/O 2 Output Trigger Select Register	0x4BA	8	read/ write	0x00	0xFF
				VBTICTLR	VBATT Input Control Register	0x4BB	8	read/ write	0x00	0xFF
				VBTOCTLR	VBATT Output Control Register	0x4BC	8	read/ write	0x00	0xFF
				VBTWTER	VBATT Wakeup Trigger source Enable Register	0x4BD	8	read/ write	0x00	0xFF
				VBTWEGR	VBATT Wakeup Trigger source Edge Register	0x4BE	8	read/ write	0x00	0xFF
				VBTWFR	VBATT Wakeup trigger source Flag Register	0x4BF	8	read/ write	0x00	0xFF
	512	0x1	0-511	VBTBKR[%s]	VBATT Backup Register [%s]	0x500	8	read/ write	0x00	0x00
PORT0,5-9	-	-	-	PCNTR1	Port Control Register 1	0x00	32	read/ write	0x00000000	0xFFFFFFFF
				PODR	Output Data Register	0x00	16	read/ write	0x0000	0xFFFF
				PDR	Data Direction Register	0x02	16	read/ write	0x0000	0xFFFF
				PCNTR2	Port Control Register 2	0x04	32	read-only	0x00000000	0xFFFF0000
				PIDR	Input Data Register	0x06	16	read-only	0x0000	0xFFFF
				PCNTR3	Port Control Register 3	0x08	32	write-only	0x00000000	0xFFFFFFFF
				PORR	Output Reset Register	0x08	16	write-only	0x0000	0xFFFF
				POSR	Output Set Register	0x0A	16	write-only	0x0000	0xFFFF
PORT1-4	-	-	-	PCNTR1	Port Control Register 1	0x00	32	read/ write	0x00000000	0xFFFFFFFF
				PODR	Output Data Register	0x00	16	read/ write	0x0000	0xFFFF
				PDR	Data Direction Register	0x02	16	read/ write	0x0000	0xFFFF
				PCNTR2	Port Control Register 2	0x04	32	read-only	0x00000000	0xFFFF0000
				EIDR	Event Input Data Register	0x04	16	read-only	0x0000	0x0000

Table 3.4 Register description (7 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
PORT1-4	-	-	-	PIDR	Input Data Register	0x06	16	read-only	0x0000	0xFFFF
				PCNTR3	Port Control Register 3	0x08	32	write-only	0x00000000	0xFFFFFFFF
				PORR	Output Reset Register	0x08	16	write-only	0x0000	0xFFFF
				POSR	Output Set Register	0x0A	16	write-only	0x0000	0xFFFF
				PCNTR4	Port Control Register 4	0x0C	32	read/write	0x00000000	0xFFFFFFFF
				EORR	Event Output Reset Register	0x0C	16	read/write	0x0000	0xFFFF
				EOSR	Event Output Set Register	0x0E	16	read/write	0x0000	0xFFFF
PFS	-	-	-	P000PFS	P000 Pin Function Control Register	0x000	32	read/write	0x00000000	0xFFFFFFFD
				P000PFS_HA	P000 Pin Function Control Register	0x002	16	read/write	0x0000	0xFFFD
				P000PFS_BY	P000 Pin Function Control Register	0x003	8	read/write	0x00	0xFD
	9	0x4	1-9	P00%sPFS	P00%s Pin Function Control Register	0x004	32	read/write	0x00000000	0xFFFFFFFD
	9	0x4	1-9	P00%sPFS_HA	P00%s Pin Function Control Register	0x006	16	read/write	0x0000	0xFFFD
	9	0x4	1-9	P00%sPFS_BY	P00%s Pin Function Control Register	0x007	8	read/write	0x00	0xFD
	6	0x4	10-15	P0%sPFS	P0%s Pin Function Control Register	0x028	32	read/write	0x00000000	0xFFFFFFFD
	6	0x4	10-15	P0%sPFS_HA	P0%s Pin Function Control Register	0x02A	16	read/write	0x0000	0xFFFD
	6	0x4	10-15	P0%sPFS_BY	P0%s Pin Function Control Register	0x02B	8	read/write	0x00	0xFD
	-	-	-	P100PFS	P100 Pin Function Control Register	0x040	32	read/write	0x00000000	0xFFFFFFFD
				P100PFS_HA	P100 Pin Function Control Register	0x042	16	read/write	0x0000	0xFFFD
				P100PFS_BY	P100 Pin Function Control Register	0x043	8	read/write	0x00	0xFD
	7	0x4	1-7	P10%sPFS	P10%s Pin Function Control Register	0x044	32	read/write	0x00000000	0xFFFFFFFD
	7	0x4	1-7	P10%sPFS_HA	P10%s Pin Function Control Register	0x046	16	read/write	0x0000	0xFFFD
	7	0x4	1-7	P10%sPFS_BY	P10%s Pin Function Control Register	0x047	8	read/write	0x00	0xFD
	-	-	-	P108PFS	P108 Pin Function Control Register	0x060	32	read/write	0x00010010	0xFFFFFFFD
				P108PFS_HA	P108 Pin Function Control Register	0x062	16	read/write	0x0010	0xFFFD
				P108PFS_BY	P108 Pin Function Control Register	0x063	8	read/write	0x10	0xFD
				P109PFS	P109 Pin Function Control Register	0x064	32	read/write	0x00010000	0xFFFFFFFD
				P109PFS_HA	P109 Pin Function Control Register	0x066	16	read/write	0x0000	0xFFFD
				P109PFS_BY	P109 Pin Function Control Register	0x067	8	read/write	0x00	0xFD
				P110PFS	P110 Pin Function Control Register	0x068	32	read/write	0x00010010	0xFFFFFFFD
				P110PFS_HA	P110 Pin Function Control Register	0x06A	16	read/write	0x0010	0xFFFD

Table 3.4 Register description (8 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
PFS	-	-	-	P110PFS_BY	P110 Pin Function Control Register	0x06B	8	read/ write	0x10	0xFD
	5	0x4	11-15	P1%sPFS	P1%s Pin Function Control Register	0x06C	32	read/ write	0x00000000	0xFFFFFFFD
	5	0x4	11-15	P1%sPFS_HA	P1%s Pin Function Control Register	0x06E	16	read/ write	0x0000	0xFFFD
	5	0x4	11-15	P1%sPFS_BY	P1%s Pin Function Control Register	0x06F	8	read/ write	0x00	0xFD
	-	-	-	P200PFS	P200 Pin Function Control Register	0x080	32	read/ write	0x00000000	0xFFFFFFFD
				P200PFS_HA	P200 Pin Function Control Register	0x082	16	read/ write	0x0000	0xFFFD
				P200PFS_BY	P200 Pin Function Control Register	0x083	8	read/ write	0x00	0xFD
				P201PFS	P201 Pin Function Control Register	0x084	32	read/ write	0x00000010	0xFFFFFFFD
				P201PFS_HA	P201 Pin Function Control Register	0x086	16	read/ write	0x0010	0xFFFD
				P201PFS_BY	P201 Pin Function Control Register	0x087	8	read/ write	0x10	0xFD
	5	0x4	2-6	P20%sPFS	P20%s Pin Function Control Register	0x088	32	read/ write	0x00000000	0xFFFFFFFD
	5	0x4	2-6	P20%sPFS_HA	P20%s Pin Function Control Register	0x08A	16	read/ write	0x0000	0xFFFD
	5	0x4	2-6	P20%sPFS_BY	P20%s Pin Function Control Register	0x08B	8	read/ write	0x00	0xFD
	4	0x4	12-15	P2%sPFS	P2%s Pin Function Control Register	0x0B0	32	read/ write	0x00000000	0xFFFFFFFD
	4	0x4	12-15	P2%sPFS_HA	P2%s Pin Function Control Register	0x0B2	16	read/ write	0x0000	0xFFFD
	4	0x4	12-15	P2%sPFS_BY	P2%s Pin Function Control Register	0x0B3	8	read/ write	0x00	0xFD
	-	-	-	P300PFS	P300 Pin Function Control Register	0x0C0	32	read/ write	0x00010010	0xFFFFFFFD
				P300PFS_HA	P300 Pin Function Control Register	0x0C2	16	read/ write	0x0010	0xFFFD
				P300PFS_BY	P300 Pin Function Control Register	0x0C3	8	read/ write	0x10	0xFD
	9	0x4	1-9	P30%sPFS	P30%s Pin Function Control Register	0x0C4	32	read/ write	0x00000000	0xFFFFFFFD
	9	0x4	1-9	P30%sPFS_HA	P30%s Pin Function Control Register	0x0C6	16	read/ write	0x0000	0xFFFD
	9	0x4	1-9	P30%sPFS_BY	P30%s Pin Function Control Register	0x0C7	8	read/ write	0x00	0xFD
	6	0x4	10-15	P3%sPFS	P3%s Pin Function Control Register	0x0E8	32	read/ write	0x00000000	0xFFFFFFFD
	6	0x4	10-15	P3%sPFS_HA	P3%s Pin Function Control Register	0x0EA	16	read/ write	0x0000	0xFFFD
	6	0x4	10-15	P3%sPFS_BY	P3%s Pin Function Control Register	0x0EB	8	read/ write	0x00	0xFD
	10	0x4	0-9	P40%sPFS	P40%s Pin Function Control Register	0x100	32	read/ write	0x00000000	0xFFFFFFFD
	10	0x4	0-9	P40%sPFS_HA	P40%s Pin Function Control Register	0x102	16	read/ write	0x0000	0xFFFD
	10	0x4	0-9	P40%sPFS_BY	P40%s Pin Function Control Register	0x103	8	read/ write	0x00	0xFD
	6	0x4	10-15	P4%sPFS	P4%s Pin Function Control Register	0x128	32	read/ write	0x00000000	0xFFFFFFFD
	6	0x4	10-15	P4%sPFS_HA	P4%s Pin Function Control Register	0x12A	16	read/ write	0x0000	0xFFFD

Table 3.4 Register description (9 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
PFS	6	0x4	10-15	P4%sPFS_BY	P4%s Pin Function Control Register	0x12B	8	read/ write	0x00	0xFD
	8	0x4	0-7	P50%sPFS	P50%s Pin Function Control Register	0x140	32	read/ write	0x00000000	0xFFFFFFFD
	8	0x4	0-7	P50%sPFS_HA	P50%s Pin Function Control Register	0x142	16	read/ write	0x0000	0xFFFD
	8	0x4	0-7	P50%sPFS_BY	P50%s Pin Function Control Register	0x143	8	read/ write	0x00	0xFD
	2	0x4	11-12	P5%sPFS	P5%s Pin Function Control Register	0x16C	32	read/ write	0x00000000	0xFFFFFFFD
	2	0x4	11-12	P5%sPFS_HA	P5%s Pin Function Control Register	0x16E	16	read/ write	0x0000	0xFFFD
	2	0x4	11-12	P5%sPFS_BY	P5%s Pin Function Control Register	0x16F	8	read/ write	0x00	0xFD
	7	0x4	0-6	P60%sPFS	P60%s Pin Function Control Register	0x180	32	read/ write	0x00000000	0xFFFFFFFD
	7	0x4	0-6	P60%sPFS_HA	P60%s Pin Function Control Register	0x182	16	read/ write	0x0000	0xFFFD
	7	0x4	0-6	P60%sPFS_BY	P60%s Pin Function Control Register	0x183	8	read/ write	0x00	0xFD
	2	0x4	8-9	P60%sPFS	P60%s Pin Function Control Register	0x1A0	32	read/ write	0x00000000	0xFFFFFFFD
	2	0x4	8-9	P60%sPFS_HA	P60%s Pin Function Control Register	0x1A2	16	read/ write	0x0000	0xFFFD
	2	0x4	8-9	P60%sPFS_BY	P60%s Pin Function Control Register	0x1A3	8	read/ write	0x00	0xFD
	5	0x4	10-14	P6%sPFS	P6%s Pin Function Control Register	0x1A8	32	read/ write	0x00000000	0xFFFFFFFD
	5	0x4	10-14	P6%sPFS_HA	P6%s Pin Function Control Register	0x1AA	16	read/ write	0x0000	0xFFFD
	5	0x4	10-14	P6%sPFS_BY	P6%s Pin Function Control Register	0x1AB	8	read/ write	0x00	0xFD
	6	0x4	0-5	P70%sPFS	P70%s Pin Function Control Register	0x1C0	32	read/ write	0x00000000	0xFFFFFFFD
	6	0x4	0-5	P70%sPFS_HA	P70%s Pin Function Control Register	0x1C2	16	read/ write	0x0000	0xFFFD
	6	0x4	0-5	P70%sPFS_BY	P70%s Pin Function Control Register	0x1C3	8	read/ write	0x00	0xFD
	2	0x4	8-9	P70%sPFS	P70%s Pin Function Control Register	0x1E0	32	read/ write	0x00000000	0xFFFFFFFD
	2	0x4	8-9	P70%sPFS_HA	P70%s Pin Function Control Register	0x1E2	16	read/ write	0x0000	0xFFFD
	2	0x4	8-9	P70%sPFS_BY	P70%s Pin Function Control Register	0x1E3	8	read/ write	0x00	0xFD
	4	0x4	10-13	P7%sPFS	P7%s Pin Function Control Register	0x1E8	32	read/ write	0x00000000	0xFFFFFFFD
	4	0x4	10-13	P7%sPFS_HA	P7%s Pin Function Control Register	0x1EA	16	read/ write	0x0000	0xFFFD
	4	0x4	10-13	P7%sPFS_BY	P7%s Pin Function Control Register	0x1EB	8	read/ write	0x00	0xFD
	10	0x4	0-9	P80%sPFS	P80%s Pin Function Control Register	0x200	32	read/ write	0x00000000	0xFFFFFFFD
	10	0x4	0-9	P80%sPFS_HA	P80%s Pin Function Control Register	0x202	16	read/ write	0x0000	0xFFFD
	10	0x4	0-9	P80%sPFS_BY	P80%s Pin Function Control Register	0x203	8	read/ write	0x00	0xFD
	3	0x4	0-2	P90%sPFS	P90%s Pin Function Control Register	0x240	32	read/ write	0x00000000	0xFFFFFFFD
	3	0x4	0-2	P90%sPFS_HA	P90%s Pin Function Control Register	0x242	16	read/ write	0x0000	0xFFFD

Table 3.4 Register description (10 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
PFS	3	0x4	0-2	P90%sPFS_BY	P90%s Pin Function Control Register	0x243	8	read/ write	0x00	0xFD
PMISC	-	-	-	PWPR	Write-Protect Register	0x03	8	read/ write	0x80	0xFF
ELC	-	-	-	ELCR	Event Link Controller Register	0x00	8	read/ write	0x00	0xFF
	2	0x2	0,1	ELSEGR%s	Event Link Software Event Generation Register %s	0x02	8	read/ write	0x80	0xFF
	10	0x4	0-9	ELSR%s	Event Link Setting Register %s	0x10	16	read/ write	0x0000	0xFFFF
	7	0x4	12-18	ELSR%s	Event Link Setting Register %s	0x40	16	read/ write	0x0000	0xFFFF
POEG	4	0x100	A,B,C,D	POEGG%s	POEG Group %s Setting Register	0x00	32	read/ write	0x00000000	0xFFFFFFFF
RTC	-	-	-	R64CNT	64-Hz Counter	0x00	8	read-only	0x00	0x80
				RSECCNT	Second Counter	0x02	8	read/ write	0x00	0x00
				BCNT0	Binary Counter 0	0x02	8	read/ write	0x00	0x00
				RMINCNT	Minute Counter	0x04	8	read/ write	0x00	0x00
				BCNT1	Binary Counter 1	0x04	8	read/ write	0x00	0x00
				RHRCNT	Hour Counter	0x06	8	read/ write	0x00	0x00
				BCNT2	Binary Counter 2	0x06	8	read/ write	0x00	0x00
RTC	-	-	-	RWKCNT	Day-of-Week Counter	0x08	8	read/ write	0x00	0x00
				BCNT3	Binary Counter 3	0x08	8	read/ write	0x00	0x00
				RDAYCNT	Day Counter	0x0A	8	read/ write	0x00	0xC0
				RMONCNT	Month Counter	0x0C	8	read/ write	0x00	0xE0
				RYRCNT	Year Counter	0x0E	16	read/ write	0x0000	0xFF00
				RSECAR	Second Alarm Register	0x10	8	read/ write	0x00	0x00
				BCNT0AR	Binary Counter 0 Alarm Register	0x10	8	read/ write	0x00	0x00
				RMINAR	Minute Alarm Register	0x12	8	read/ write	0x00	0x00
				BCNT1AR	Binary Counter 1 Alarm Register	0x12	8	read/ write	0x00	0x00
				RHRAR	Hour Alarm Register	0x14	8	read/ write	0x00	0x00
				BCNT2AR	Binary Counter 2 Alarm Register	0x14	8	read/ write	0x00	0x00
				RWKAR	Day-of-Week Alarm Register	0x16	8	read/ write	0x00	0x00
				BCNT3AR	Binary Counter 3 Alarm Register	0x16	8	read/ write	0x00	0x00
				RDAYAR	Date Alarm Register	0x18	8	read/ write	0x00	0x00
				BCNT0AER	Binary Counter 0 Alarm Enable Register	0x18	8	read/ write	0x00	0x00

Table 3.4 Register description (11 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
RTC	-	-	-	RMONAR	Month Alarm Register	0x1A	8	read/ write	0x00	0x00
				BCNT1AER	Binary Counter 1 Alarm Enable Register	0x1A	8	read/ write	0x00	0x00
				RYRAR	Year Alarm Register	0x1C	16	read/ write	0x0000	0xFF00
				BCNT2AER	Binary Counter 2 Alarm Enable Register	0x1C	16	read/ write	0x0000	0xFF00
				RYRAREN	Year Alarm Enable Register	0x1E	8	read/ write	0x00	0x00
				BCNT3AER	Binary Counter 3 Alarm Enable Register	0x1E	8	read/ write	0x00	0x00
				RCR1	RTC Control Register 1	0x22	8	read/ write	0x00	0x0A
				RCR2	RTC Control Register 2	0x24	8	read/ write	0x00	0x0E
				RCR4	RTC Control Register 4	0x28	8	read/ write	0x00	0xFE
				RFRH	Frequency Register H	0x2A	16	read/ write	0x0000	0xFFFE
				RFRL	Frequency Register L	0x2C	16	read/ write	0x0000	0x0000
				RADJ	Time Error Adjustment Register	0x2E	8	read/ write	0x00	0x00
	3	0x2	0-2	RTCCR%s	Time Capture Control Register %s	0x40	8	read/ write	0x00	0x00
	3	0x10	0-2	RSECCP%s	Second Capture Register %s	0x52	8	read-only	0x00	0x00
	3	0x10	0-2	BCNT0CP%s	BCNT0 Capture Register %s	0x52	8	read-only	0x00	0x00
	3	0x10	0-2	RMINCP%s	Minute Capture Register %s	0x54	8	read-only	0x00	0x00
	3	0x10	0-2	BCNT1CP%s	BCNT1 Capture Register %s	0x54	8	read-only	0x00	0x00
	3	0x10	0-2	RHRCP%s	Hour Capture Register %s	0x56	8	read-only	0x00	0x00
	3	0x10	0-2	BCNT2CP%s	BCNT2 Capture Register %s	0x56	8	read-only	0x00	0x00
	3	0x10	0-2	RDAYCP%s	Date Capture Register %s	0x5A	8	read-only	0x00	0x00
	3	0x10	0-2	BCNT3CP%s	BCNT3 Capture Register %s	0x5A	8	read-only	0x00	0x00
	3	0x10	0-2	RMONCP%s	Month Capture Register %s	0x5C	8	read-only	0x00	0x00
WDT	-	-	-	WDTRR	WDT Refresh Register	0x00	8	read/ write	0xFF	0xFF
				WDTCR	WDT Control Register	0x02	16	read/ write	0x33F3	0xFFFF
				WDTSR	WDT Status Register	0x04	16	read/ write	0x0000	0xFFFF
				WDTRCR	WDT Reset Control Register	0x06	8	read/ write	0x80	0xFF
				WDTCSTPR	WDT Count Stop Control Register	0x08	8	read/ write	0x80	0xFF

Table 3.4 Register description (12 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
IWDT	-	-	-	IWDTRR	IWDT Refresh Register	0x00	8	read/ write	0xFF	0xFF
				IWDTSR	IWDT Status Register	0x04	16	read/ write	0x0000	0xFFFF
CAC	-	-	-	CACR0	CAC Control Register 0	0x00	8	read/ write	0x00	0xFF
				CACR1	CAC Control Register 1	0x01	8	read/ write	0x00	0xFF
				CACR2	CAC Control Register 2	0x02	8	read/ write	0x00	0xFF
				CAICR	CAC Interrupt Control Register	0x03	8	read/ write	0x00	0xFF
				CASTR	CAC Status Register	0x04	8	read-only	0x00	0xFF
				CAULVR	CAC Upper-Limit Value Setting Register	0x06	16	read/ write	0x0000	0xFFFF
CAC	-	-	-	CALLVR	CAC Lower-Limit Value Setting Register	0x08	16	read/ write	0x0000	0xFFFF
				CACNTBR	CAC Counter Buffer Register	0x0A	16	read-only	0x0000	0xFFFF
MSTP	-	-	-	MSTPCRB	Module Stop Control Register B	0x00	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
				MSTPCRC	Module Stop Control Register C	0x04	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
				MSTPCRD	Module Stop Control Register D	0x08	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
SSI0,1	-	-	-	SSICR	Control Register	0x00	32	read/ write	0x00000000	0xFFFFFFFF
				SSISR	Status Register	0x04	32	read/ write	0x02000013	0x3E00007F
				SSIFCR	FIFO Control Register	0x10	32	read/ write	0x00000000	0xFFFFFFFF
				SSIFSR	FIFO Status Register	0x14	32	read/ write	0x00010000	0xFFFFFFFF
				SSIFTDR	Transmit FIFO Data Register	0x18	32	write-only	0x00000000	0x00000000
				SSIFRDR	Receive FIFO Data Register	0x1C	32	read-only	0x00000000	0x00000000
				SSITDMR	TDM Mode Register	0x20	32	read/ write	0x00000000	0xFFFFFFFF
CAN0	32	0x10	0-31	MB%s_ID	Mailbox Register	0x200	32	read/ write	0x00000000	0x00000000
	32	0x10	0-31	MB%s_DL	Mailbox Register	0x204	16	read/ write	0x0000	0x0000
	32	0x10	0-31	MB%s_D0	Mailbox Register	0x206	8	read/ write	0x00	0x00
	32	0x10	0-31	MB%s_D1	Mailbox Register	0x207	8	read/ write	0x00	0x00
	32	0x10	0-31	MB%s_D2	Mailbox Register	0x208	8	read/ write	0x00	0x00
	32	0x10	0-31	MB%s_D3	Mailbox Register	0x209	8	read/ write	0x00	0x00
	32	0x10	0-31	MB%s_D4	Mailbox Register	0x20A	8	read/ write	0x00	0x00
	32	0x10	0-31	MB%s_D5	Mailbox Register	0x20B	8	read/ write	0x00	0x00
	32	0x10	0-31	MB%s_D6	Mailbox Register	0x20C	8	read/ write	0x00	0x00

Table 3.4 Register description (13 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
CAN0	32	0x10	0-31	MB%s_D7	Mailbox Register	0x20D	8	read/ write	0x00	0x00
	32	0x10	0-31	MB%s_TS	Mailbox Register	0x20E	16	read/ write	0x0000	0x0000
	8	0x4	0-7	MKR[%s]	Mask Register	0x400	32	read/ write	0x00000000	0x00000000
	2	0x4	0,1	FIDCR%s	FIFO Received ID Compare Registers	0x420	32	read/ write	0x00000000	0x00000000
	-	-	-	MKIVLR	Mask Invalid Register	0x428	32	read/ write	0x00000000	0x00000000
				MIER	Mailbox Interrupt Enable Register	0x42C	32	read/ write	0x00000000	0x00000000
				MIER_FIFO	Mailbox Interrupt Enable Register for FIFO Mailbox Mode	0x42C	32	read/ write	0x00000000	0x00000000
	32	0x1	0-31	MCTL_TX[%s]	Message Control Register for Transmit	0x820	8	read/ write	0x00	0xFF
	32	0x1	0-31	MCTL_RX[%s]	Message Control Register for Receive	0x820	8	read/ write	0x00	0xFF
	-	-	-	CTLR	Control Register	0x840	16	read/ write	0x0500	0xFFFF
				STR	Status Register	0x842	16	read-only	0x0500	0xFFFF
				BCR	Bit Configuration Register	0x844	32	read/ write	0x00000000	0xFFFFFFFF
				RFCR	Receive FIFO Control Register	0x848	8	read/ write	0x80	0xFF
				RFPCR	Receive FIFO Pointer Control Register	0x849	8	write-only	0x00	0x00
				TFCR	Transmit FIFO Control Register	0x84A	8	read/ write	0x80	0xFF
				TFPCR	Transmit FIFO Pointer Control Register	0x84B	8	write-only	0x00	0x00
				EIER	Error Interrupt Enable Register	0x84C	8	read/ write	0x00	0xFF
				EIFR	Error Interrupt Factor Judge Register	0x84D	8	read/ write	0x00	0xFF
				RECR	Receive Error Count Register	0x84E	8	read-only	0x00	0xFF
				TECR	Transmit Error Count Register	0x84F	8	read-only	0x00	0xFF
				ECSR	Error Code Store Register	0x850	8	read/ write	0x00	0xFF
				CSSR	Channel Search Support Register	0x851	8	read/ write	0x00	0x00
				MSSR	Mailbox Search Status Register	0x852	8	read-only	0x80	0xFF
				MSMR	Mailbox Search Mode Register	0x853	8	read/ write	0x00	0xFF
				TSR	Time Stamp Register	0x854	16	read-only	0x0000	0xFFFF
				AFSR	Acceptance Filter Support Register	0x856	16	read/ write	0x0000	0x0000
				TCR	Test Control Register	0x858	8	read/ write	0x00	0xFF
IIC0	-	-	-	ICCR1	I²C Bus Control Register 1	0x00	8	read/ write	0x1F	0xFF
				ICCR2	I²C Bus Control Register 2	0x01	8	read/ write	0x00	0xFF

Table 3.4 Register description (14 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
IIC0	-	-	-	ICMR1	I2C Bus Mode Register 1	0x02	8	read/ write	0x08	0xFF
				ICMR2	I2C Bus Mode Register 2	0x03	8	read/ write	0x06	0xFF
				ICMR3	I2C Bus Mode Register 3	0x04	8	read/ write	0x00	0xFF
				ICFER	I2C Bus Function Enable Register	0x05	8	read/ write	0x72	0xFF
				ICSER	I2C Bus Status Enable Register	0x06	8	read/ write	0x09	0xFF
				ICIER	I2C Bus Interrupt Enable Register	0x07	8	read/ write	0x00	0xFF
				ICSR1	I2C Bus Status Register 1	0x08	8	read/ write	0x00	0xFF
				ICSR2	I2C Bus Status Register 2	0x09	8	read/ write	0x00	0xFF
	3	0x2	0-2	SARL%s	Slave Address Register L%s	0x0A	8	read/ write	0x00	0xFF
	3	0x2	0-2	SARU%s	Slave Address Register U%s	0x0B	8	read/ write	0x00	0xFF
	-	-	-	ICBRL	I2C Bus Bit Rate Low-Level Register	0x10	8	read/ write	0xFF	0xFF
				ICBRH	I2C Bus Bit Rate High-Level Register	0x11	8	read/ write	0xFF	0xFF
				ICDRT	I2C Bus Transmit Data Register	0x12	8	read/ write	0xFF	0xFF
				ICDRR	I2C Bus Receive Data Register	0x13	8	read-only	0x00	0xFF
				ICWUR	I2C Bus Wake Up Unit Register	0x16	8	read/ write	0x10	0xFF
				ICWUR2	Reserved	0x17	8	read-only	0xFF	0xFF
IIC1,2	-	-	-	ICCR1	I2C Bus Control Register 1	0x00	8	read/ write	0x1F	0xFF
				ICCR2	I2C Bus Control Register 2	0x01	8	read/ write	0x00	0xFF
				ICMR1	I2C Bus Mode Register 1	0x02	8	read/ write	0x08	0xFF
				ICMR2	I2C Bus Mode Register 2	0x03	8	read/ write	0x06	0xFF
				ICMR3	I2C Bus Mode Register 3	0x04	8	read/ write	0x00	0xFF
				ICFER	I2C Bus Function Enable Register	0x05	8	read/ write	0x72	0xFF
				ICSER	I2C Bus Status Enable Register	0x06	8	read/ write	0x09	0xFF
				ICIER	I2C Bus Interrupt Enable Register	0x07	8	read/ write	0x00	0xFF
				ICSR1	I2C Bus Status Register 1	0x08	8	read/ write	0x00	0xFF
				ICSR2	I2C Bus Status Register 2	0x09	8	read/ write	0x00	0xFF

Table 3.4 Register description (15 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
IIC1,2	3	0x2	0-2	SARL%s	Slave Address Register L%s	0x0A	8	read/ write	0x00	0xFF
	3	0x2	0-2	SARU%s	Slave Address Register U%s	0x0B	8	read/ write	0x00	0xFF
	-	-	-	ICBRL	I2C Bus Bit Rate Low-Level Register	0x10	8	read/ write	0xFF	0xFF
				ICBRH	I2C Bus Bit Rate High-Level Register	0x11	8	read/ write	0xFF	0xFF
				ICDRT	I2C Bus Transmit Data Register	0x12	8	read/ write	0xFF	0xFF
				ICDRR	I2C Bus Receive Data Register	0x13	8	read-only	0x00	0xFF
DOC	-	-	-	DOCR	DOC Control Register	0x00	8	read/ write	0x00	0xFF
				DODIR	DOC Data Input Register	0x02	16	read/ write	0x0000	0xFFFF
				DODSR	DOC Data Setting Register	0x04	16	read/ write	0x0000	0xFFFF
ADC140	-	-	-	ADCSR	A/D Control Register	0x000	16	read/ write	0x0000	0xFFFF
				ADANSA0	A/D Channel Select Register A0	0x004	16	read/ write	0x0000	0xFFFF
				ADANSA1	A/D Channel Select Register A1	0x006	16	read/ write	0x0000	0xFFFF
				ADADS0	A/D-Converted Value Addition/Average Channel Select Register 0	0x008	16	read/ write	0x0000	0xFFFF
				ADADS1	A/D-Converted Value Addition/Average Channel Select Register 1	0x00A	16	read/ write	0x0000	0xFFFF
				ADADC	A/D-Converted Value Addition/Average Count Select Register	0x00C	8	read/ write	0x00	0xFF
				ADCER	A/D Control Extended Register	0x00E	16	read/ write	0x0000	0xFFFF
				ADSTRGR	A/D Conversion Start Trigger Select Register	0x010	16	read/ write	0x0000	0xFFFF
				ADEXICR	A/D Conversion Extended Input Control Register	0x012	16	read/ write	0x0000	0xFFFF
				ADANSB0	A/D Channel Select Register B0	0x014	16	read/ write	0x0000	0xFFFF
				ADANSB1	A/D Channel Select Register B1	0x016	16	read/ write	0x0000	0xFFFF
				ADDBLDR	A/D Data Duplication Register	0x018	16	read-only	0x0000	0xFFFF
				ADTSDR	A/D Temperature Sensor Data Register	0x01A	16	read-only	0x0000	0xFFFF

Table 3.4 Register description (16 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
ADC140	-	-	-	ADOCDR	A/D Internal Reference Voltage Data Register	0x01C	16	read-only	0x0000	0xFFFF
				ADRD	A/D Self-Diagnosis Data Register	0x01E	16	read-only	0x0000	0xFFFF
	28	0x2	0-27	ADDR%s	A/D Data Register %s	0x020	16	read-only	0x0000	0xFFFF
	-	-	-	ADDISCR	A/D Disconnection Detection Control Register	0x07A	8	read/write	0x00	0xFF
				ADGSPCR	A/D Group Scan Priority Control Register	0x080	16	read/write	0x0000	0xFFFF
				ADDBLDRA	A/D Data Duplexing Register A	0x084	16	read-only	0x0000	0xFFFF
				ADDBLDRB	A/D Data Duplexing Register B	0x086	16	read-only	0x0000	0xFFFF
				ADHVREFCNT	A/D High-Potential/ Low-Potential Reference Voltage Control Register	0x08A	8	read/write	0x00	0xFF
				ADWINMON	A/D Compare Function Window A/B Status Monitor Register	0x08C	8	read-only	0x00	0xFF
				ADCMPCR	A/D Compare Function Control Register	0x090	16	read/write	0x0000	0xFFFF
				ADCMPANSER	A/D Compare Function Window A Extended Input Select Register	0x092	8	read/write	0x00	0xFF
				ADCMPLER	A/D Compare Function Window A Extended Input Comparison Condition Setting Register	0x093	8	read/write	0x00	0xFF
				ADCMPANSR0	A/D Compare Function Window A Channel Select Register 0	0x094	16	read/write	0x0000	0xFFFF
				ADCMPANSR1	A/D Compare Function Window A Channel Select Register 1	0x096	16	read/write	0x0000	0xFFFF
				ADCMPLR0	A/D Compare Function Window A Comparison Condition Setting Register 0	0x098	16	read/write	0x0000	0xFFFF
				ADCMPLR1	A/D Compare Function Window A Comparison Condition Setting Register 1	0x09A	16	read/write	0x0000	0xFFFF

Table 3.4 Register description (17 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
ADC140	-	-	-	ADCMPDR0	A/D Compare Function Window A Lower-Side Level Setting Register	0x09C	16	read/ write	0x0000	0xFFFF
				ADCMPDR1	A/D Compare Function Window A Upper-Side Level Setting Register	0x09E	16	read/ write	0x0000	0xFFFF
				ADCMPSR0	A/D Compare Function Window A Channel Status Register 0	0x0A0	16	read/ write	0x0000	0xFFFF
				ADCMPSR1	A/D Compare Function Window A Channel Status Register 1	0x0A2	16	read/ write	0x0000	0xFFFF
				ADCMPSER	A/D Compare Function Window A Extended Input Channel Status Register	0x0A4	8	read/ write	0x00	0xFF
				ADCMPBNSR	A/D Compare Function Window B Channel Selection Register	0x0A6	8	read/ write	0x00	0xFF
				ADWINLLB	A/D Compare Function Window B Lower-Side Level Setting Register	0x0A8	16	read/ write	0x0000	0xFFFF
				ADWINULB	A/D Compare Function Window B Upper-Side Level Setting Register	0x0AA	16	read/ write	0x0000	0xFFFF
				ADCMPBSR	A/D Compare Function Window B Status Register	0x0AC	8	read/ write	0x00	0xFF
				ADSSTRL	A/D Sampling State Register L	0x0DD	8	read/ write	0x0D	0xFF
				ADSSTRT	A/D Sampling State Register T	0x0DE	8	read/ write	0x0D	0xFF
				ADSSTRO	A/D Sampling State Register O	0x0DF	8	read/ write	0x0D	0xFF
	16	0x1	0-15	ADSSTR%s	A/D Sampling State Register %s	0x0E0	8	read/ write	0x0D	0xFF
DAC12	2	0x2	0,1	DADR%s	D/A Data Register %s	0x00	16	read/ write	0x0000	0xFFFF
	-	-	-	DACR	D/A Control Register	0x04	8	read/ write	0x1F	0xFF
				DADPR	DADR0 Format Select Register	0x05	8	read/ write	0x00	0xFF
				DAADSCR	D/A-A/D Synchronous Start Control Register	0x06	8	read/ write	0x00	0xFF
				DAVREFCR	D/A VREF Control Register	0x07	8	read/ write	0x00	0xFF

Table 3.4 Register description (18 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SDHI0	-	-	-	SD_CMD	Command Type Register	0x000	32	read/ write	0x00000000	0xFFFFFFFF
				SD_ARG	SD Command Argument Register	0x008	32	read/ write	0x00000000	0xFFFFFFFF
				SD_ARG1	SD Command Argument Register 1	0x00C	32	read/ write	0x00000000	0xFFFFFFFF
				SD_STOP	Data Stop Register	0x010	32	read/ write	0x00000000	0xFFFFFFFF
				SD_SECCNT	Block Count Register	0x014	32	read/ write	0x00000000	0xFFFFFFFF
				SD_RSP10	SD Card Response Register 10	0x018	32	read-only	0x00000000	0xFFFFFFFF
				SD_RSP1	SD Card Response Register 1	0x01C	32	read-only	0x00000000	0xFFFFFFFF
				SD_RSP32	SD Card Response Register 32	0x020	32	read-only	0x00000000	0xFFFFFFFF
				SD_RSP3	SD Card Response Register 3	0x024	32	read-only	0x00000000	0xFFFFFFFF
				SD_RSP54	SD Card Response Register 54	0x028	32	read-only	0x00000000	0xFFFFFFFF
				SD_RSP5	SD Card Response Register 5	0x02C	32	read-only	0x00000000	0xFFFFFFFF
				SD_RSP76	SD Card Response Register 76	0x030	32	read-only	0x00000000	0xFFFFFFFF
				SD_RSP7	SD Card Response Register 7	0x034	32	read-only	0x00000000	0xFFFFFFFF
				SD_INFO1	SD Card Interrupt Flag Register 1	0x038	32	read/ write	0x00000010	0xFFFFFB7F
				SD_INFO2	SD Card Interrupt Flag Register 2	0x03C	32	read/ write	0x00002000	0xFFFFFF7F
				SD_INFO1_MASK	SD_INFO1 Interrupt Mask Register	0x040	32	read/ write	0x0000031D	0xFFFFFFFF
				SD_INFO2_MASK	SD_INFO2 Interrupt Mask Register	0x044	32	read/ write	0x00008B7F	0xFFFFFFFF
				SD_CLK_CTRL	SD Clock Control Register	0x048	32	read/ write	0x00000020	0xFFFFFFFF
				SD_SIZE	Transfer Data Length Register	0x04C	32	read/ write	0x00000200	0xFFFFFFFF
				SD_OPTION	SD Card Access Control Option Register	0x050	32	read/ write	0x000040EE	0xFFFFFFFF
				SD_ERR_STS1	SD Error Status Register 1	0x058	32	read-only	0x00002000	0xFFFFFFFF
				SD_ERR_STS2	SD Error Status Register 2	0x05C	32	read-only	0x00000000	0xFFFFFFFF
				SD_BUF0	SD Buffer Read/Write Register	0x060	32	read/ write	0x00000000	0x00000000
				SDIO_MODE	SDIO Mode Control Register	0x068	32	read/ write	0x00000000	0xFFFFFFFF
				SDIO_INFO1	SDIO Interrupt Flag Register 1	0x06C	32	read/ write	0x00000000	0xFFFFFFF9

Table 3.4 Register description (19 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SDHI0	-	-	-	SDIO_INFO1_MASK	SDIO_INFO1 Interrupt Mask Register	0x070	32	read/ write	0x0000C007	0xFFFFFFFF
				SD_DMAEN	DMA Mode Enable Register	0x1B0	32	read/ write	0x00001010	0xFFFFFFFF
				SOFT_RST	Software Reset Register	0x1C0	32	read/ write	0x00000006	0xFFFFFFFF
				SDIF_MODE	SD Interface Mode Setting Register	0x1CC	32	read/ write	0x00000007	0xFFFFFFFF
				EXT_SWAP	Swap Control Register	0x1E0	32	read/ write	0x00000000	0xFFFFFFFF
SCI0-4,9	-	-	-	SMR	Serial Mode Register (SCMR.SMIF = 0)	0x00	8	read/ write	0x00	0xFF
				SMR_SMCI	Serial Mode Register (SCMR.SMIF = 1)	0x00	8	read/ write	0x00	0xFF
				BRR	Bit Rate Register	0x01	8	read/ write	0xFF	0xFF
				SCR	Serial Control Register (SCMR.SMIF = 0)	0x02	8	read/ write	0x00	0xFF
				SCR_SMCI	Serial Control Register (SCMR.SMIF = 1)	0x02	8	read/ write	0x00	0xFF
				TDR	Transmit Data Register	0x03	8	read/ write	0xFF	0xFF
				SSR	Serial Status Register (SCMR.SMIF = 0 and FCR.FM)	0x04	8	read/ write	0x84	0xFF
				SSR_FIFO	Serial Status Register(SCMR.SMIF = 0 and FCR.FM=1)	0x04	8	read/ write	0x80	0xFD
				SSR_SMCI	Serial Status Register(SCMR.SMIF = 1)	0x04	8	read/ write	0x84	0xFF
				RDR	Receive Data Register	0x05	8	read-only	0x00	0xFF
				SCMR	Smart Card Mode Register	0x06	8	read/ write	0xF2	0xFF
				SEMR	Serial Extended Mode Register	0x07	8	read/ write	0x00	0xFF
				SNFR	Noise Filter Setting Register	0x08	8	read/ write	0x00	0xFF
				SIMR1	I^2C Mode Register 1	0x09	8	read/ write	0x00	0xFF
				SIMR2	I^2C Mode Register 2	0x0A	8	read/ write	0x00	0xFF
				SIMR3	I^2C Mode Register 3	0x0B	8	read/ write	0x00	0xFF
				SISR	I^2C Status Register	0x0C	8	read-only	0x00	0xCB
				SPMR	SPI Mode Register	0x0D	8	read/ write	0x00	0xFF

Table 3.4 Register description (20 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SCI0-4,9	-	-	-	TDRHL	Transmit 9-bit Data Register	0x0E	16	read/ write	0xFFFF	0xFFFF
				FTDRHL	Transmit FIFO Data Register HL	0x0E	16	write-only	0xFFFF	0xFFFF
				FTDRH	Transmit FIFO Data Register H	0x0E	8	write-only	0xFF	0xFF
				FTDRL	Transmit FIFO Data Register L	0x0F	8	write-only	0xFF	0xFF
				RDRHL	Receive 9-bit Data Register	0x10	16	read-only	0x0000	0xFFFF
				FRDRHL	Receive FIFO Data Register HL	0x10	16	read-only	0x0000	0xFFFF
				FRDRH	Receive FIFO Data Register H	0x10	8	read-only	0x00	0xFF
				FRDRL	Receive FIFO Data Register L	0x11	8	read-only	0x00	0xFF
				MDDR	Modulation Duty Register	0x12	8	read/ write	0xFF	0xFF
				DCCR	Data Compare Match Control Register	0x13	8	read/ write	0x40	0xFF
				FCR	FIFO Control Register	0x14	16	read/ write	0xF800	0xFFFF
				FDR	FIFO Data Count Register	0x16	16	read-only	0x0000	0xFFFF
				LSR	Line Status Register	0x18	16	read-only	0x0000	0xFFFF
				CDR	Compare Match Data Register	0x1A	16	read/ write	0x0000	0xFFFF
				SPTR	Serial Port Register	0x1C	8	read/ write	0x03	0xFF
IRDA	-	-	-	IRCR	IrDA Control Register	0x00	8	read/ write	0x00	0xFF
SPI0,1	-	-	-	SPCR	SPI Control Register	0x00	8	read/ write	0x00	0xFF
				SSLP	SPI Slave Select Polarity Register	0x01	8	read/ write	0x00	0xFF
				SPPCR	SPI Pin Control Register	0x02	8	read/ write	0x00	0xFF
				SPSR	SPI Status Register	0x03	8	read/ write	0x20	0xFF
				SPDR	SPI Data Register	0x04	32	read/ write	0x00000000	0xFFFFFFFF
				SPDR_HA	SPI Data Register (halfword access)	0x04	16	read/ write	0x0000	0xFFFF
				SPSCR	SPI Sequence Control Register	0x08	8	read/ write	0x00	0xFF
				SPSSR	SPI Sequence Status Register	0x09	8	read-only	0x00	0xFF
				SPBR	SPI Bit Rate Register	0x0A	8	read/ write	0xFF	0xFF
				SPDCR	SPI Data Control Register	0x0B	8	read/ write	0x00	0xFF
				SPCKD	SPI Clock Delay Register	0x0C	8	read/ write	0x00	0xFF
				SSLND	SPI Slave Select Negation Delay Register	0x0D	8	read/ write	0x00	0xFF
				SPND	SPI Next-Access Delay Register	0x0E	8	read/ write	0x00	0xFF

Table 3.4 Register description (21 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
SPI0,1	-	-	-	SPCR2	SPI Control Register 2	0x0F	8	read/ write	0x00	0xFF
	8	0x2	0-7	SPCMD%s	SPI Command Register %s	0x10	16	read/ write	0x070D	0xFFFF
CRC	-	-	-	CRCCR0	CRC Control Register0	0x00	8	read/ write	0x00	0xFF
				CRCCR1	CRC Control Register1	0x01	8	read/ write	0x00	0xFF
				CRCDIR	CRC Data Input Register	0x04	32	read/ write	0x00000000	0xFFFFFFFF
				CRCDIR_BY	CRC Data Input Register (byte access)	0x04	8	read/ write	0x00	0xFF
				CRCDOR	CRC Data Output Register	0x08	32	read/ write	0x00000000	0xFFFFFFFF
				CRCDOR_HA	CRC Data Output Register (halfword access)	0x08	16	read/ write	0x0000	0xFFFF
				CRCDOR_BY	CRC Data Output Register (byte access)	0x08	8	read/ write	0x00	0xFF
				CRCSAR	Snoop Address Register	0x0C	16	read/ write	0x0000	0xFFFF
GPT320-9	-	-	-	GTWP	General PWM Timer Write-Protection Register	0x00	32	read/ write	0x00000000	0xFFFFFFFF
				GTSTR	General PWM Timer Software Start Register	0x04	32	read/ write	0x00000000	0xFFFFFFFF
				GTSTP	General PWM Timer Software Stop Register	0x08	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
				GTCLR	General PWM Timer Software Clear Register	0x0C	32	write-only	0x00000000	0xFFFFFFFF
				GTSSR	General PWM Timer Start Source Select Register	0x10	32	read/ write	0x00000000	0xFFFFFFFF
				GTPSR	General PWM Timer Stop Source Select Register	0x14	32	read/ write	0x00000000	0xFFFFFFFF
				GTCSR	General PWM Timer Clear Source Select Register	0x18	32	read/ write	0x00000000	0xFFFFFFFF
				GTUPSR	General PWM Timer Up Count Source Select Register	0x1C	32	read/ write	0x00000000	0xFFFFFFFF
				GTDNSR	General PWM Timer Down Count Source Select Register	0x20	32	read/ write	0x00000000	0xFFFFFFFF
				GTICASR	General PWM Timer Input Capture Source Select Register A	0x24	32	read/ write	0x00000000	0xFFFFFFFF
				GTICBSR	General PWM Timer Input Capture Source Select Register B	0x28	32	read/ write	0x00000000	0xFFFFFFFF
				GTCR	General PWM Timer Control Register	0x2C	32	read/ write	0x00000000	0xFFFFFFFF
				GTUDDTYC	General PWM Timer Count Direction and Duty Setting Register	0x30	32	read/ write	0x00000001	0xFFFFFFFF

Table 3.4 Register description (22 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
GPT320-9	-	-	-	GTIOR	General PWM Timer I/O Control Register	0x34	32	read/ write	0x00000000	0xFFFFFFFF
				GTINTAD	General PWM Timer Interrupt Output Setting Register	0x38	32	read/ write	0x00000000	0xFFFFFFFF
				GTST	General PWM Timer Status Register	0x3C	32	read/ write	0x00008000	0xFFFFFFFF
				GTBER	General PWM Timer Buffer Enable Register	0x40	32	read/ write	0x00000000	0xFFFFFFFF
				GTCNT	General PWM Timer Counter	0x48	32	read/ write	0x00000000	0xFFFFFFFF
				GTCCRA	General PWM Timer Compare Capture Register A	0x4C	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
				GTCCRB	General PWM Timer Compare Capture Register B	0x50	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
				GTCCRC	General PWM Timer Compare Capture Register C	0x54	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
				GTCCRE	General PWM Timer Compare Capture Register E	0x58	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
				GTCCRD	General PWM Timer Compare Capture Register D	0x5C	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
				GTCCRF	General PWM Timer Compare Capture Register F	0x60	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
				GTPR	General PWM Timer Cycle Setting Register	0x64	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
				GTPBR	General PWM Timer Cycle Setting Buffer Register	0x68	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
				GTDTCR	General PWM Timer Dead Time Control Register	0x88	32	read/ write	0x00000000	0xFFFFFFFF
				GTDVU	General PWM Timer Dead Time Value Register U	0x8C	32	read/ write	0xFFFFFFFF	0xFFFFFFFF
GPT_OPS	-	-	-	OPSCR	Output Phase Switching Control Register	0x00	32	read/ write	0x00000000	0xFFFFFFFF
KINT	-	-	-	KRCTL	KEY Return Control Register	0x00	8	read/ write	0x00	0xFF
				KRF	KEY Return Flag Register	0x04	8	read/ write	0x00	0xFF
				KRM	KEY Return Mode Register	0x08	8	read/ write	0x00	0xFF

Table 3.4 Register description (23 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
CTSU	-	-	-	CTSUCR0	CTSU Control Register 0	0x00	8	read/ write	0x00	0xFF
				CTSUCR1	CTSU Control Register 1	0x01	8	read/ write	0x00	0xFF
				CTSUSDPRS	CTSU Synchronous Noise Reduction Setting Register	0x02	8	read/ write	0x00	0xFF
				CTSUSST	CTSU Sensor Stabilization Wait Control Register	0x03	8	read/ write	0x00	0xFF
				CTSUMCH0	CTSU Measurement Channel Register 0	0x04	8	read/ write	0x3F	0xFF
				CTSUMCH1	CTSU Measurement Channel Register 1	0x05	8	read/ write	0x3F	0xFF
				CTSUCHAC0	CTSU Channel Enable Control Register 0	0x06	8	read/ write	0x00	0xFF
				CTSUCHAC1	CTSU Channel Enable Control Register 1	0x07	8	read/ write	0x00	0xFF
				CTSUCHAC2	CTSU Channel Enable Control Register 2	0x08	8	read/ write	0x00	0xFF
				CTSUCHAC3	CTSU Channel Enable Control Register 3	0x09	8	read/ write	0x00	0xFF
				CTSUCHAC4	CTSU Channel Enable Control Register 4	0x0A	8	read/ write	0x00	0xFF
				CTSUCHTRC0	CTSU Channel Transmit/Receive Control Register 0	0x0B	8	read/ write	0x00	0xFF
				CTSUCHTRC1	CTSU Channel Transmit/Receive Control Register 1	0x0C	8	read/ write	0x00	0xFF
				CTSUCHTRC2	CTSU Channel Transmit/Receive Control Register 3	0x0D	8	read/ write	0x00	0xFF
				CTSUCHTRC3	CTSU Channel Transmit/Receive Control Register 3	0x0E	8	read/ write	0x00	0xFF
				CTSUCHTRC4	CTSU Channel Transmit/Receive Control Register 4	0x0F	8	read/ write	0x00	0xFF
				CTSUDCLKC	CTSU High-Pass Noise Reduction Control Register	0x10	8	read/ write	0x00	0xFF
				CTSUST	CTSU Status Register	0x11	8	read/ write	0x00	0xFF
				CTSUSSC	CTSU High-Pass Noise Reduction Spectrum Diffusion Control Register	0x12	16	read/ write	0x0000	0xFFFF
				CTSUSO0	CTSU Sensor Offset Register 0	0x14	16	read/ write	0x0000	0xFFFF
				CTSUSO1	CTSU Sensor Offset Register 1	0x16	16	read/ write	0x0000	0xFFFF
				CTSUSC	CTSU Sensor Counter	0x18	16	read-only	0x0000	0xFFFF
				CTSURC	CTSU Reference Counter	0x1A	16	read-only	0x0000	0xFFFF

Table 3.4 Register description (24 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
CTSU	-	-	-	CTSUERRS	CTSU Error Status Register	0x1C	16	read-only	0x0000	0xFFFF
SLCDC	-	-	-	LCDM0	LCD Mode Register 0	0x000	8	read/write	0x00	0xFF
				LCDM1	LCD Mode Register 1	0x001	8	read/write	0x00	0xFF
				LCDC0	LCD Clock Control Register 0	0x002	8	read/write	0x00	0xFF
				VLCD	LCD Boost Level Control Register	0x003	8	read/write	0x04	0xFF
	52	0x1	0-51	SEG%s	LCD Display Data Register %s	0x100	8	read/write	0x00	0xFF
AGT0,1	-	-	-	AGT	AGT Counter Register	0x00	16	read/write	0xFFFF	0xFFFF
				AGTCMA	AGT Compare Match A Register	0x02	16	read/write	0xFFFF	0xFFFF
				AGTCMB	AGT Compare Match B Register	0x04	16	read/write	0xFFFF	0xFFFF
				AGTCR	AGT Control Register	0x08	8	read/write	0x00	0xFF
				AGTMR1	AGT Mode Register 1	0x09	8	read/write	0x00	0xFF
				AGTMR2	AGT Mode Register 2	0x0A	8	read/write	0x00	0xFF
				AGTIOC	AGT I/O Control Register	0x0C	8	read/write	0x00	0xFF
				AGTISR	AGT Event Pin Select Register	0x0D	8	read/write	0x00	0xFF
				AGTCMSR	AGT Compare Match Function Select Register	0x0E	8	read/write	0x00	0xFF
				AGTIOSEL	AGT Pin Select Register	0x0F	8	read/write	0x00	0xFF
ACMPHS0	-	-	-	CMPCTL	Comparator Control Register	0x000	8	read/write	0x00	0xFF
				CMPSEL0	Comparator Input Select Register	0x004	8	read/write	0x00	0xFF
				CMPSEL1	Comparator Reference Voltage Select Register	0x008	8	read/write	0x00	0xFF
				CMPMON	Comparator Output Monitor Register	0x00C	8	read-only	0x00	0xFF
				CPIOC	Comparator Output Control Register	0x010	8	read/write	0x00	0xFF
ACMPHS1	-	-	-	CMPCTL	Comparator Control Register	0x000	8	read/write	0x00	0xFF
				CMPSEL0	Comparator Input Select Register	0x004	8	read/write	0x00	0xFF
				CMPSEL1	Comparator Reference Voltage Select Register	0x008	8	read/write	0x00	0xFF
				CMPMON	Comparator Output Monitor Register	0x00C	8	read-only	0x00	0xFF
				CPIOC	Comparator Output Control Register	0x010	8	read/write	0x00	0xFF

Table 3.4 Register description (25 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
ACMPLP	-	-	-	COMPMDR	ACMPLP Mode Setting Register	0x00	8	read/ write	0x00	0xFF
				COMPFIR	ACMPLP Filter Control Register	0x01	8	read/ write	0x00	0xFF
				COMPOCR	ACMPLP Output Control Register	0x02	8	read/ write	0x00	0xFF
OPAMP	-	-	-	AMPMC	Operational Amplifier Mode Control Register	0x08	8	read/ write	0x00	0xFF
				AMPTRM	Operational Amplifier Trigger Mode Control Register	0x09	8	read/ write	0x00	0xFF
				AMPTRS	Operational Amplifier Activation Trigger Select Register	0x0A	8	read/ write	0x00	0xFF
				AMPC	Operational Amplifier Control Register	0x0B	8	read/ write	0x00	0xFF
				AMPMON	Operational Amplifier Monitor Register	0x0C	8	read-only	0x00	0xFF
USBFS	-	-	-	SYSCFG	System Configuration Control Register	0x000	16	read/ write	0x0000	0xFFFF
				SYSSTS0	System Configuration Status Register 0	0x004	16	read-only	0x0000	0x0000
				DVSTCTR0	Device State Control Register 0	0x008	16	read/ write	0x0000	0xFFFF
				CFIFO	CFIFO Port Register	0x014	16	read/ write	0x0000	0xFFFF
				CFIFOL	CFIFO Port Register L	0x014	8	read/ write	0x00	0xFF
				D0FIFO	D0FIFO Port Register	0x018	16	read/ write	0x0000	0xFFFF
				D0FIFOL	D0FIFO Port Register L	0x018	8	read/ write	0x00	0xFF
				D1FIFO	D1FIFO Port Register	0x01C	16	read/ write	0x0000	0xFFFF
				D1FIFOL	D1FIFO Port Register L	0x01C	8	read/ write	0x00	0xFF
				CFIFOSEL	CFIFO Port Select Register	0x020	16	read/ write	0x0000	0xFFFF
				CFIFOCTR	CFIFO Port Control Register	0x022	16	read/ write	0x0000	0xFFFF
				D0FIFOSEL	D0FIFO Port Select Register	0x028	16	read/ write	0x0000	0xFFFF
				D0FIFOCTR	D0FIFO Port Control Register	0x02A	16	read/ write	0x0000	0xFFFF
				D1FIFOSEL	D1FIFO Port Select Register	0x02C	16	read/ write	0x0000	0xFFFF
				D1FIFOCTR	D1FIFO Port Control Register	0x02E	16	read/ write	0x0000	0xFFFF
				INTENB0	Interrupt Enable Register 0	0x030	16	read/ write	0x0000	0xFFFF
				INTENB1	Interrupt Enable Register 1	0x032	16	read/ write	0x0000	0xFFFF
				BRDYENB	BRDY Interrupt Enable Register	0x036	16	read/ write	0x0000	0xFFFF
				NRDYENB	NRDY Interrupt Enable Register	0x038	16	read/ write	0x0000	0xFFFF
				BEMPENB	BEMP Interrupt Enable Register	0x03A	16	read/ write	0x0000	0xFFFF

Table 3.4 Register description (26 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
USBFS	-	-	-	SOFCFG	SOF Output Configuration Register	0x03C	16	read/ write	0x0000	0xFFFF
				INTSTS0	Interrupt Status Register 0	0x040	16	read/ write	0x0000	0xFF7F
				INTSTS1	Interrupt Status Register 1	0x042	16	read/ write	0x0000	0xFFFF
				BRDYSTS	BRDY Interrupt Status Register	0x046	16	read/ write	0x0000	0xFFFF
				NRDYSTS	NRDY Interrupt Status Register	0x048	16	read/ write	0x0000	0xFFFF
				BEMPSTS	BEMP Interrupt Status Register	0x04A	16	read/ write	0x0000	0xFFFF
				FRMNUM	Frame Number Register	0x04C	16	read/ write	0x0000	0xFFFF
				USBREQ	USB Request Type Register	0x054	16	read/ write	0x0000	0xFFFF
				USBVAL	USB Request Value Register	0x056	16	read/ write	0x0000	0xFFFF
				USBINDX	USB Request Index Register	0x058	16	read/ write	0x0000	0xFFFF
				USBLENG	USB Request Length Register	0x05A	16	read/ write	0x0000	0xFFFF
				DCPCFG	DCP Configuration Register	0x05C	16	read/ write	0x0000	0xFFFF
				DCPMAXP	DCP Maximum Packet Size Register	0x05E	16	read/ write	0x0040	0xFFFF
				DCPCTR	DCP Control Register	0x060	16	read/ write	0x0040	0xFFFF
				PIPESEL	Pipe Window Select Register	0x064	16	read/ write	0x0000	0xFFFF
				PIPECFG	Pipe Configuration Register	0x068	16	read/ write	0x0000	0xFFFF
				PIPEMAXP	Pipe Maximum Packet Size Register	0x06C	16	read/ write	0x0000	0xFFBF
				PIPEPERI	Pipe Cycle Control Register	0x06E	16	read/ write	0x0000	0xFFFF
	5	0x002	1-5	PIPE%sCTR	Pipe %s Control Register	0x070	16	read/ write	0x0000	0xFFFF
	4	0x002	6-9	PIPE%sCTR	Pipe %s Control Register	0x07A	16	read/ write	0x0000	0xFFFF
	5	0x004	1-5	PIPE%sTRE	Pipe %s Transaction Counter Enable Register	0x090	16	read/ write	0x0000	0xFFFF
	5	0x004	1-5	PIPE%sTRN	Pipe %s Transaction Counter Register	0x092	16	read/ write	0x0000	0xFFFF
	-	-	-	USBBCCTRL0	BC Control Register 0	0x0B0	16	read/ write	0x0000	0xFFFF
				USBMC	USB Module Control Register	0x0CC	16	read/ write	0x0002	0xFFFF
	6	0x002	0-5	DEVADD%s	Device Address %s Configuration Register	0x0D0	16	read/ write	0x0000	0xFFFF
TSN	-	-	-	TSCDRH	Temperature Sensor Calibration Data Register H	0x229	8	read-only	0x00	0x00
				TSCDRL	Temperature Sensor Calibration Data Register L	0x228	8	read-only	0x00	0x00

Table 3.4 Register description (27 of 27)

Peripheral	Dim	Dim incr.	Dim index	Register name	Description	Address offset	Size	Access	Reset value	Reset mask
QSPI	-	-	-	SFMSMD	Transfer Mode Control Register	0x000	32	read/ write	0x00000000	0xFFFFFFFF
				SFMSSC	Chip Selection Control Register	0x004	32	read/ write	0x00000037	0xFFFFFFFF
				SFMSKC	Clock Control Register	0x008	32	read/ write	0x00000008	0xFFFFFFFF
				SFMSST	Status Register	0x00C	32	read-only	0x00000080	0xFFFFFFFF
				SFMCOM	Communication Port Register	0x010	32	read/ write	0x00000000	0xFFFFFF00
				SFMCMD	Communication Mode Control Register	0x014	32	read/ write	0x00000000	0xFFFFFFFF
				SFMCST	Communication Status Register	0x018	32	read/ write	0x00000000	0xFFFFFFFF
				SFMSIC	Instruction Code Register	0x020	32	read/ write	0x00000000	0xFFFFFFFF
QSPI	-	-	-	SFMSAC	Address Mode Control Register	0x024	32	read/ write	0x00000002	0xFFFFFFFF
				SFMSDC	Dummy Cycle Control Register	0x028	32	read/ write	0x0000FF00	0xFFFFFFFF
				SFMSPC	SPI Protocol Control Register	0x030	32	read/ write	0x00000010	0xFFFFFFFF
				SFMPMD	Port Control Register	0x034	32	read/ write	0x00000000	0xFFFFFFFF
				SFMCNT1	External QSPI Address Register 1	0x804	32	read/ write	0x00000000	0xFFFFFFFF

Peripheral name = Name of peripheral
Dim = Number of elements in an array of registers
Dim inc = Address increment between two neighboring registers of a register array in the address map
Dim index = Substring that replaces the %s placeholder in the register name
Register name = Name of register
Description = Register description
Address offset = Address of the register relative to the base address defined by the peripheral of the register
Size = Bit width of the register
Access = Register access rights:
Read-only: Read access is permitted. Write operations have undefined results.
Write-only: Write access is permitted. Read operations have undefined results.
Read-write: Both read and write accesses are permitted. Writes affect the state of the register and reads return a value related to the register.
Reset value = Default reset value of a register
Reset mask = Identifies which register bits have a defined reset value

Revision History	S3A7 Microcontroller Group User's Manual

Rev.	Date	Chapter	Summary
1.00	Feb 23, 2016	-	Synergy S3A7 User's Manual
		section 1, Overview	Updated channel number of CTSU in Table 1.14, Function comparison
			Updated pin name of CTSU in section 1.5, Pin Functions
			Updated pin name of CTSU in section 1.7, Pin Lists
		section 2, CPU	Updated the component name for address E00F F000h and E00F F004h in Table 2.7, CoreSight ROM Table
		section 7, Option-Setting Memory	Updated the Note in section 7.2.2, Option Function Select Register 1 (OFS1)
			Updated bit [14] FSPR in section 7.2.4, Access Window Setting Control Register (AWSC) for the restriction on the FSPR bit in the AWSC register
		section 9, Clock Generation Circuit	Updated Table 9.1, Clock Generation Circuit specifications for the clock sources for the restriction on the PLL output frequency division ratio
		section 11, Low Power Modes	Added Note 12. to Table 11.2, Operating conditions of each low power mode
			Added Note 3. to Table 11.3, Interrupt sources to transition to Normal mode from Snooze mode and Software Standby mode
			Updated description in section 11.2.9, Snooze End Control Register (SNZEDCR)
			Added section 11.9.12, Conditions of CTSU in Snooze Mode
		section 12, Battery Backup Function	Updated the access permission for bit VBTRVLD in section 12.2.3, VBATT Status Register (VBTSR)
			Updated bit [0] VWEN in section 12.2.7, VBATT Wakeup Control Register (VBTWCTLR)
			Updated description in section 12.2.12, VBATT Output Control Register (VBTOCTLR)
			Updated bit names for bits [0], [1], and [2] in section 12.2.15, VBATT Wakeup Trigger Source Flag Register (VBTWFR)
		section 14, Interrupt Controller Unit (ICU)	Added Note 3. and updated Table 14.4, Event table
		section 15, Buses	Changed address output pins from A23 to A16 in Table 15.4, External pin configurations, and throughout the document
			Updated section 15.3.6, CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 3)
			Updated section 15.7, Notes on using Flash Cache
		section 18, Data Transfer Controller (DTC)	Updated address range in section 18.2.8, DTC Vector Base Register (DTCVBR)
		section 19, Event Link Controller (ELC)	Updated section 19.4.3, Module Stop Function Setting for the restriction on ELCON bit setting
		section 20, I/O Ports	Added description for P402, P403, and P404 and updated description for VBTICTLR in section 20.5.5, I/O Buffer Specification
			Updated Table 20.4, Register settings for input/output pin function (PORT0) for CTSU in P002 to P004 and P007
			Updated Table 20.4, Register settings for input/output pin function (PORT0) for the ASEL bit in P014 and P015
			Updated Table 20.5, Register settings for input/output pin function (PORT1) (1) for AGT in P100 and P101
			Updated Table 20.7, Register settings for input/output pin function (PORT2) (1) for AGT/CTSU in P204
			Updated Table 20.8, Register settings for input/output pin function (PORT2) (2) for AGT in P212
			Updated Table 20.10, Register settings for input/output pin function (PORT4) (1) for 64-pin product in P406
			Updated Table 20.10, Register settings for input/output pin function (PORT4) (1) for DSCR bit in P400
		section 21, Key Interrupt Function (KINT)	Updated section 21.4, Usage Note

Rev.	Date	Chapter	Summary
1.00	Feb 23, 2016	section 25, Realtime Clock (RTC)	Updated section 25.2.20, Frequency Register (RFRH/RFRL)
		section 28, USB 2.0 Full-Speed Module (USBFS)	Added Table 28.6, Endian operation in 8-bit access in section 28.2.4, CFIFO Port Register (CFIFO/CFIFOL) D0FIFO Port Register (D0FIFO/D0FIFOL) D1FIFO Port Register (D1FIFO/D1FIFOL)
			Updated Figure 28.3, Example of power supply connection when the USB LDO regulator is used (BC used) and Figure 28.4, Example of power supply connection when the USB LDO regulator is used (BC not used)
		section 29, Serial Communications Interface (SCI)	Updated the initial value and access permission for bit [1] in section 29.2.14, Serial Status Register for Non-Smart Card Interface and FIFO Mode (SSR_FIFO) (SCMR.SMIF = 0 and FCR.FM = 1)
			Updated the initial value and access permission for bit [6] in section 29.2.31, Serial Port Register (SPTR)
		section 34, Quad Serial Peripheral Interface (QSPI)	Updated Table 34.3, Relationship between SFMDV[4:0] bits, cycle multiplier, and serial clock frequencies
		section 35, Cyclic Redundancy Check (CRC) Calculator	Updated description for CRCSA [13:0] in section 35.2.5, Snoop Address Register (CRCSAR)
		section 39, 14-Bit A/D Converter (ADC14)	Added section 39.8.14, Notes on Canceling Software Standby Mode for the restriction on canceling Software Standby mode to use A/D conversion
		section 41, Temperature Sensor (TSN)	Deleted the TBD from section 41.3.1, Preparation for Using Temperature Sensor
		section 45, Capacitive Touch Sensing Unit (CTSU)	Updated channel number in Table 45.1, CTSU specifications
			Updated pin name in Table 45.2, CTSU pin configuration
			Updated descriptions in section 45.2.5, CTSU Measurement Channel Register 0 (CTSUMCH0)
			Updated descriptions in section 45.2.6, CTSU Measurement Channel Register 1 (CTSUMCH1)
			Updated descriptions and added Note 1.in section 45.2.9, CTSU Channel Enable Control Register 2 (CTSUCHAC2)
			Updated descriptions and added Note 1.in section 45.2.10, CTSU Channel Enable Control Register 3 (CTSUCHAC3)
			Updated descriptions and added Note 1.in section 45.2.14, CTSU Channel Transmit/Receive Control Register 2 (CTSUCHTRC2)
			Updated descriptions and added Note 1.in section 45.2.15, CTSU Channel Transmit/Receive Control Register 3 (CTSUCHTRC3)
			Updated bit [15] CTSUICOMP in section 45.2.24, CTSU Error Status Register (CTSUERRS)
			Updated section 45.4.5, TSCAP Pin
		section 52, Electrical Characteristics	Added section 52.17, Joint Test Action Group (JTAG) and section 52.17.1, Serial Wire Debug (SWD) in section 52, Electrical Characteristics
			Updated input voltage in Table 52.1, Absolute maximum ratings
			Added section 52.2.5, I/O Pin Output Characteristics of Low Drive Capacity
			Updated Table 52.6, I/O I_{OH}, I_{OL} in section 52.2.3, I/O I_{OH}, I_{OL} to change from normal drive to low drive
			Changed Note 6 to Note 5. in Table 52.11, Operating and standby current (1)
			Updated the conditions in Table 52.13, Operating and standby current (3)
			Updated Note 2. in Table 52.17, Operation frequency value in High-speed operating mode
			Updated Note 2. in Table 52.18, Operation frequency value in Middle-speed mode
			Removed the 2nd note from Table 52.19, Operation frequency value in Low-speed mode
			Updated Note 2. in Table 52.20, Operation frequency value in Low-voltage mode
			Updated Table 52.22, Clock timing
			Updated the condition of the I/O Ports in Table 52.35, I/O Ports, POEG, GPT, AGT, KINT, and ADC14 trigger timing
			Removed the 2nd note from Table 52.37, SCI timing (1)
			Updated the conditions in Table 52.38, SCI timing (2)

Rev.	Date	Chapter	Summary
1.00	Feb 23, 2016	section 52, Electrical Characteristics	Updated Figure 52.60, SPI timing (master, CPHA = 0) (bit rate: PCLKA division ratio is set to 1/2)
			Added the conditions in Table 52.42, IIC timing
			Updated Figure 52.69, SSI data transmit/receive timing (SSICR.SCKP = 0)
			Updated the Quantization error in the following tables: • Table 52.48, A/D conversion characteristics (1) in High-speed A/D conversion mode • Table 52.49, A/D conversion characteristics (2) in High-speed A/D conversion mode • Table 52.50, A/D conversion characteristics (3) in High-speed A/D conversion mode • Table 52.51, A/D conversion characteristics (4) in Low-power A/D conversion mode • Table 52.52, A/D conversion characteristics (5) in Low-power A/D conversion mode.
			Updated Table 52.55, 14-Bit A/D converter channel classification
			Updated Table 52.64, Battery Backup Function Characteristics
			Deleted VLCD = 0Dh to 13h in Table 52.71, Internal voltage boosting method LCD characteristics
			Updated the response time in Table 52.73, ACMPHS characteristics
			Added the temperature in Table 52.78, Code flash characteristics (3)
			Added the temperature in Table 52.81, Data flash characteristics (3)
		All	Deleted # from pin names
1.10	Jun 30, 2016	Features	Updated Connectivity feature for the external address space description
		section 1, Overview	Updated Table 1.3, System as follows: • Moved Interrupt Controller Unit (ICU) feature from Table 1.4, Interrupt control, to the row following CAC, and removed the Table 1.4, Interrupt control • Moved Key Interrupt Function (KINT) feature from Table 1.10, Human machine interfaces, to the row following ICU • In Memory Protection Unit (MPU) feature, changed the memory protection unit number from two to four.
			Updated Figure 1.1, Block diagram to move ICU and KINT from the Interrupt Control block (now removed) and Human Machine Interfaces blocks respectively, to System.
			Added introduction to section 1.3, Part Numbering and updated Figure 1.2, Part numbering scheme
			Added Table 1.13, Product list
			Updated Table 1.14, Function comparison as follows: • Removed the Interrupt control row • Moved ICU from Interrupt control to System • Moved KINT from HMI to System.
			Updated the Table 1.5, Pin Functions as follows: • Moved the interrupt function to the row below CAC • Moved the KINT function to the row above the on-chip debug function.
			Updated the Table 1.7, Pin Lists as follows: • Moved the interrupt column to precede the I/O ports column • Removed CTX1_A for D4 in the USBFS, CAN column • Removed CRX1_A for B1 in the USBFS, CAN column • Added /SCK1_A to N1 in the SCI column.
		section 2, CPU	Added introductory sentence to section 2, MCU Implementation Options
			Updated section 2.6.4.1, Debug Status Register (DBGSTR)
			Updated section 2.6.4.2, Debug Stop Control Register (DBGSTOPCR)
			Updated section 2.6.5.3, MCU Control Register (MCUCTRL)
			Added section 2.8, Flash Patch and Break Unit
			Updated section 2.11.3.4, Connecting Sequence and JTAG/SWD Authentication
		section 4, Address Space	Updated Figure 4.1, Memory map
			Updated Figure 4.2, Association between external address spaces and CS areas
		section 6, Resets	Updated Software reset entry in Table 6.1, Reset names and sources
			Updated Figure 6.2, Example of operations during voltage monitor 1 and voltage monitor 2 resets
			Updated Figure 6.3, Example of cold/warm start determination operation
		section 7, Option-Setting Memory	Updated Table 7.2, Specifications for ID code protection

Rev.	Date	Chapter	Summary
1.10	Jun 30, 2016	section 8, Low Voltage Detection (LVD)	Updated Table 8.1, LVD specifications
			Updated section 8.4, Reset from Voltage Monitor 0
		section 9, Clock Generation Circuit	Updated the Note following Table 9.2, Clock Generation Circuit Specifications (internal clock)
		section 9, Clock Generation Circuit	Updated CKSEL[2:0] bits (Clock Source Select) to remove reference to USBFS clock (UCLK), and added a Note
			Updated PLLSTP bit (PLL Stop Control)
			Updated MEMWAIT bit (Memory Wait Cycle Select) to add a Note
			Updated Figure 9.2, When setting the ICLK > 32 MHz
			Updated Figure 9.3, When setting the ICLK ≤ 32 MHz from ICLK > 32 MHz
			Updated HCSTP bit (HOCO Stop)
			Updated HOCOSF flag (HOCO Clock Oscillation Stabilization Flag)
			Updated OSTDF flag (Oscillation Stop Detection Flag)
			Updated section 9.2.16, High-Speed On-Chip Oscillator Wait Control Register (HOCOWTCR)
			Updated CKOEN bit (Clock Out enable)
			Updated the Note in section 9.2.25, Trace Clock Control Register (TRCKCR)
			Updated section 9.5.1, Oscillation Stop Detection and Operation after Detection
		section 11, Low Power Modes	Updated section 11.5.1, Setting Operating Power Control Mode
			Updated section 11.6.2, Canceling Sleep Mode
			Updated section 11.8.4, Snooze Operation Example
			Updated section 11.9.6, Timing of WFI Instruction
		section 12, Battery Backup Function	Changed "Tamper" pin to "Time capture" pin
			Updated Table 12.2, Operating states in VBATT mode
			Updated Figure 12.3.5, VBATT Wakeup Control Function Usage
			Updated Figure 12.6, The timing chart of VBATT wakeup function
			Updated step 3 in section 12.4, Usage Note
		section 14, Interrupt Controller Unit (ICU)	Removed section USBFSWUPEN bit (USBFS Interrupt Software Standby Returns Enable)
		section 15, Buses	Updated Table 15.4, External pin configurations
		section 20, I/O Ports	Updated section 20.2.1, Port Control Register 1 (PCNTR1/PODR/PDR)
			Updated section 20.2.2, Port Control Register 2 (PCNTR2/EIDR/PIDR)
			Updated section 20.2.3, Port Control Register 3 (PCNTR3/PORR/POSR)
			Updated section 20.2.4, Port Control Register 4 (PCNTR4/EORR/EOSR)
			Updated section 20.2.5, Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) (m = 0 to 9; n = 00 to 15)
			Updated the sentence "All pins operate as general I/O input ports after reset" to "All pins except P108, P109, P110, and P300 operate as general I/O input ports after reset"
			Updated the ELC_PORTx references to add x = 0, 1, 2, 3, or 4 in section 20.3.1, General I/O Ports and section 20.5.3, Port Output Data Register (PODR) Summary
			Updated section 20.5.5, I/O Buffer Specification
			Updated Table 20.7, Register settings for input/output pin function (PORT2) (1)
		section 21, Key Interrupt Function (KINT)	Updated KEY to KINT in section 21.2.1, Key Return Control Register (KRCTL) to section 21.2.3, Key Return Mode Register (KRM)
		section 22, Port Output Enable for GPT (POEG)	Updated Figure 22.1, POEG block diagram
			Updated section 22.2.1, POEG Group n Setting Register (POEGGn) (n = A to D)
			Updated section 22.3, Output-Disable Control Operation
			Updated Table 22.3, Interrupt sources and conditions
		section 23., General PWM Timer (GPT)	Added a Note to the External trigger pin entry in Table 23.2, GPT Functions
			Updated Figure 23.1, GPT block diagram
			Updated Figure 23.2, Correspondence between GPT channels and module names
			Updated Figure 23.10, Example for setting an event count operation in down-counting using hardware sources
			Updated section 23.3.3.2, Saw-Wave One-Shot Pulse Mode
			Updated section 23.3.3.5, Triangle-Wave PWM Mode 3 (64-bit transfer at trough)
			Updated Figure 23.52, Example for setting count start/stop operation by hardware source

Rev.	Date	Chapter	Summary
1.10	Jun 30, 2016	section 23., General PWM Timer (GPT)	Updated Figure 23.55, Example for setting count clearing operation by a hardware source
			Updated Figure 23.60, Example for setting simultaneous start by hardware source
			Updated step 3 in section 23.9.2, Settings of GTCCRn during Compare Match Operation (n = A to F) to remove the description regarding the restrictions for GTCCRE and GTCCRF registers
		section 24., Asynchronous General Purpose Timer (AGT)	Updated the Note in section 24.1, Overview to replace RTC with AGT
			Removed section 24.4.2, Access to Flags (TEDGF, TUNDF, TCMAF, and TCMBF Bits in AGTCR Register) following section 24.4.1, Count Operation Start and Stop Control
		section 25, Realtime Clock (RTC)	Updated section 25.2.1, 64-Hz Counter to remove R64CNT from the section heading.
		section 26, Watchdog Timer (WDT)	Updated section 26.2.2, WDT Control Register (WDTCR)
			Updated section 26.2.4, WDT Reset Control Register (WDTRCR)
			Updated section 26.2.5, WDT Count Stop Control Register (WDTCSTPR)
		section 28, USB 2.0 Full-Speed Module (USBFS)	Updated Note 2 in section 28.3.15.1, Processing in Device Controller Mode
		section 29, Serial Communications Interface (SCI)	Updated section 29.2.13, Serial Status Register (SSR) for Non-Smart Card Interface and Non-FIFO Mode (SCMR.SMIF = 0 and FCR.FM = 0)
			Updated section 29.2.14, Serial Status Register for Non-Smart Card Interface and FIFO Mode (SSR_FIFO) (SCMR.SMIF = 0 and FCR.FM = 1)
			Updated section 29.2.15, Serial Status Register for Smart Card Interface Mode (SSR_SMCI) (SCMR.SMIF = 1)
			Updated IICSTIF in section 29.2.23, I²C Mode Register 3 (SIMR3)
			Updated FM in section 29.2.26, FIFO Control Register (FCR)
			Updated section 29.2.30, Data Compare Match Control Register (DCCR)
			Updated step 3 in the procedure following Figure 29.19, Data format stored to FRDRH and FRDRL with FIFO selected to remove the sentence "When the FRDRL register..."
			Updated section 29.4.2, Multi-Processor Serial Data Reception
			Updated Figure 29.30, Example flow of serial reception in multi-processor mode with FIFO selected to add "Continued in..." at the bottom of the graphic
			Updated section 29.5.6, Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode including adding Note 2 in Figure 29.44, Example flow of simultaneous serial transmission and reception in clock synchronous mode with FIFO selected
			Updated section 29.11, Event Linking to move step 6, Address non-match event output to section 29.12, Address non-match event output (SCI0_DCUF)
			Updated section 29.14.2, SCI Operations during Low Power State
			Updated step 1 in section 29.14.6, Restrictions on Clock Synchronous Transmission in Clock Synchronous and Simple SPI Modes
			Updated Figure 29.78, Restrictions on the use of external clock in clock synchronous transmission to change t ≥ 5 cycles of PCLK to t ≥ 1 PCLK cycle + data output delay time for the slave (tDO) + setup time for the master (tSU)
		section 30, IrDA Interface	Updated section 30.3.2, Transmission to add the sentence "When the serial data is 1, no pulses are output."
			Updated section 30.3.3, Reception
		section 31, I²C Bus Interface (IIC)	Updated the Note following Table 31.8, Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 1 and NFE = 1
			Updated step 2 in Figure 31.35, Example operation of command recovery and EEP response modes when wakeup is triggered by a wakeup interrupt on a match of the slave address
			Updated Table 31.11, Register states when issuing each condition to remove the duplicate entry for ICWUR and the corresponding Note 1
		section 34, Quad Serial Peripheral Interface (QSPI)	Updated section 34.2.4, Status Register (SFMSST)
		section 35, Cyclic Redundancy Check (CRC) Calculator	Updated the Note 1 following section Table 35.1, CRC calculator specifications

Rev.	Date	Chapter	Summary
1.10	Jun 30, 2016	section 36, Serial Sound Interface (SSI)	Updated Table 36.1, SSI specifications
			Updated Figure 36.1, SSI0 block diagram
			Updated Table 36.2, SSI1 block diagram
			Updated PDTA and CKS bits in section 36.2.1, Control Register (SSICR)
		section 37, SD/MMC Host Interface (SDHI)	Updated STP bit in section 37.2.4, Data Stop Register (SD_STOP)
			Updated section 37.2.11, SD Card Interrupt Flag Register 2 (SD_INFO2)
			Updated section 37.2.20, SDIO Mode Control Register (SDIO_MODE)
			Updated section 37.2.21, SDIO Interrupt Flag Register (SDIO_INFO1)
		section 38, Boundary Scan	Added the Note to section 38.2.3, Bypass Register (JTBPR)
		section 39, 14-Bit A/D Converter (ADC14)	Updated Figure 39.22, Setting example when using event output of the compare function to change ADCSR0 to ADCSR
			Updated Figure 39.31, Procedures for clearing the ADCSR.ADST bit by software to change ADCSR.PGS to ADGSPCR.PGS
			Added section 39.8.10, Notes on Board Design
			Removed section 39.8.12, Caution When Using an External Bus
			Updated section 39.8.12, Port Setting when Using the 14-bit A/D Converter Input
		section 40, 12-Bit D/A Converter (DAC12)	Updated section 40.2.4, D/A A/D Synchronous Start Control Register (DAADSCR) to remove the sentence "Unit 1 must be selected as the target unit of the ADC14 before setting the DAADST bit to 1."
			Updated section 40.3.1, Minimizing Interference between D/A and A/D Conversion
			Updated section Figure 40.3, Example conversion when DAC12 is synchronized with ADC14 for step numbering
			Updated section Figure 40.4, Example when the DAC12 cannot capture the synchronous D/A conversion enable input signal from the ADC14 for step numbering
		section 44, Low Power Analog Comparator (ACMPLP)	Updated PCLK to PCLKB in section 44.2.2, ACMPLP Filter Control Register (COMPFIR)
			Updated Table 44.3, Procedure for setting the ACMPLP associated registers (i = 0, 1)
		section 45, Capacitive Touch Sensing Unit (CTSU)	Updated Table 45.1, CTSU specifications
			Updated Figure 45.3, CTSU block diagram
			Updated Table 45.2.2, CTSU Control Register 1 (CTSUCR1)
			Updated Figure 45.10, CTSU stopping flow
			Updated section 45.4.2, Software Trigger including Figure 45.21, Notes on restarting measurement
		section 47., SRAM	Updated section 47.3.3, ECC Error Generation to remove the sentence "When the debugger is connected, reset and NMI interrupt do not occur. However, EEC1 bit errors are corrected."
			Updated section 47.3.5, Parity Calculation Function to remove a Note, "When a debugger is connected, reset and NMI interrupt do not occur."
			Removed the following sentence after section Table 47.2, SRAM error sources: "Reset and non-maskable interrupt do not occur when a debugger is connected. Other ECC functions are not affected by the debugger."
		section 48, Flash Memory	Updated Table 48.6, Specifications for ID code protection
			Updated section 48.4, Operation to add step 2. and Note.
		section 52, Electrical Characteristics	Added a sentence regarding pin function after Figure 52.1, Input or output timing measurement conditions
			Updated Table 52.1, Absolute maximum ratings, added Note 4 following this table. Also updated Note 1 to move some text out of this note to Caution.
			Updated Table 52.7, I/O V_{OH}, V_{OL} (1) to Table 52.9, I/O V_{OH}, V_{OL} (3)
			Updated Table 52.11, Operating and standby current (1)
			Updated Table 52.15, Rise and fall gradient characteristics and the Note 2 following the table
			Updated Figure 52.27, EXTAL external clock input timing
			Updated Table 52.23, Reset timing
			Updated the Note 1 following Table 52.24, Timing of recovery from Low power modes (1), Table 52.25, Timing of Recovery from Low power modes (2), Table 52.26, Timing of recovery from Low power modes (3), and Table 52.27, Timing of recovery from Low power modes (4)
			Updated the Conditions preceding Table 52.37, SCI timing (1)
			Updated the Conditions preceding Table 52.38, SCI timing (2)

Rev.	Date	Chapter	Summary
1.10	Jun 30, 2016	section 52, Electrical Characteristics	Updated the Conditions preceding Table 52.39, SCI timing (3)
			Updated Table 52.42, IIC timing for t_{Sr} and t_{Sf}
			Removed the Note 2 following Table 52.42, IIC timing
			Updated Figure 52.67, I2C bus interface input/output timing to change SDA3 and SCL3 to SDA2 and SCL2 respectively
			Updated the Conditions preceding Table 52.46, USB characteristics
			Updated the Conditions preceding Table 52.57, D/A conversion characteristics (1)
			Removed the Conditions preceding Table 52.62, Power-on reset circuit and voltage detection circuit characteristics (1)
			Updated the Table 52.63, Power-on reset circuit and voltage detection circuit characteristics (2) and removed the Conditions preceding the table
			Updated the Conditions preceding Table 52.75, OPAMP characteristics
			Updated the Note 1 following Table 52.76, Code flash characteristics (1)
			Updated Table 52.77, Code flash characteristics (2)
			Updated Table 52.78, Code flash characteristics (3)
		Appendix 1. Port States in Each Processing Mode	Updated Table 1.1, Port states in each processing state
		Appendix 2. Package Dimensions	Updated Figure 2.2, LQFP 144-pin
1.20	Aug 26, 2016	section 2, CPU	Updated the start address for SCS in Table 2.6, Cortex-M4 peripheral address map from E000 ED00h to E000 0000h
		section 5, Memory Mirror Function (MMF)	Updated section 5.3.2, Setting Example to change MME.EN to MMEN.EN
		section 6, Resets	Updated Table 6.1, Reset names and sources to add a row for VBATT selected voltage power on reset
			Updated the Note following Table 6.1, Reset names and sources to add $V_{DETBATT}$ to the voltages being monitored
		section 7, Option-Setting Memory	Updated the description for bit FSPR to remove references to FISR register that is not documented
		section 8, Low Voltage Detection (LVD)	Updated section 8.2.3, Voltage Monitor 2 Circuit Control Register 1 (LVD2CR1) to swap Note and Note1
			Updated the RN bit (Voltage Monitor 1 Reset Negate Select) description to change LOCOCR.LCSTP to MOCOCR.MCSTP in section 8.2.7, Voltage Monitor 1 Circuit Control Register 0 (LVD1CR0)
			Updated the RN bit (Voltage Monitor 2 Reset Negate Select) description to change LOCOCR.LCSTP to MOCOCR.MCSTP in section 8.2.8, Voltage Monitor 2 Circuit Control Register 0 (LVD2CR0)
			Updated Table 8.4, Procedures for setting bits related to the voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitor operates to add a reference to Note 1 to step 4
		section 9, Clock Generation Circuit specifications for the clock sources	Updated Table 9.2, Clock Generation Circuit Specifications (internal clock) for the PCLKB peripheral modules in the Clock supply column
			Updated section 9.2.21, External Bus Clock Output Control Register (EBCKOCR) to change BCLK to EBCLK in the register table
			Updated section 9.7.2, Peripheral Module Clock (PCLKA, PCLKB, PCLKC, PCLKD) to change PLOIDIV[1:0] to PLODIV[1:0]
			Updated section 9.7.4, External Bus Clock (BCLK)
		section 11, Low Power Modes	Updated Table 11.2, Operating conditions of each low power mode
			Updated section 11.5.2, Operating range to remove "The maximum operating frequency during flash programming/erasure is 16 MHz when the operating voltage is 2.4 V or larger and smaller than 2.7 V"
			Updated section 11.9.1, Register Access step (4) to change DMA to DMAC
			Updated section 11.9.6, Timing of WFI Instruction
		section 14, Interrupt Controller Unit (ICU)	Updated section 14.2.6, ICU Event Link Setting Register n (IELSRn) (n= 0 to 63)
			Updated section 14.2.7, DMAC Event Link Setting Register n (DELSRn)
			Updated Figure 14.2, Interrupt path of the ICU and CPU: NVIC to change ICU to CPU

Rev.	Date	Chapter	Summary
1.20	Aug 26, 2016	section 15, Buses	Updated section 15.2.6, Restrictions to remove (1) Prohibition of access to address space that spans areas
			Updated values after reset in section 15.3.9, Bus Error Address Register (BUSnERRADD) (n = 1 to 4) from 00000000h to xxxxxxxxh, and the Note following the register table
			Updated section 15.3.10, Bus Error Status Register (BUSnERRSTAT) (n = 1 to 4) to change the ACCSTAT bit value after reset from 0 to x, and updated the Note following the register table
			Updated the description for ERRSTAT bit (Bus Error Status)
		section 16., Memory Protection Unit (MPU)	Updated the description following the register table in section 16.6.1.1, Security MPU Program Counter Start Address Register (SECMPUPCSn) (n = 0, 1)
		section 17., DMA Controller (DMAC)	Updated Module-Stop Function description in section 17.7, Low Power Consumption Function to correct MSTPA28 to MSTPA22
		section 18, Data Transfer Controller (DTC)	Updated Figure 18.1, DTC block diagram to correct the register name for MRA
		section 19, Event Link Controller (ELC)	Updated Table 19.1, ELC specifications
			Updated Figure 19.1, ELC block diagram (n = 0 to 9, 12 to 18)
			Updated the description for SEG bit (Software Event Generation)
			Updated Table 19.4, Module operations when event occurs
		section 22, Port Output Enable for GPT (POEG)	Updated section 22.3, Output-Disable Control Operation
		section 23, General PWM Timer (GPT)	Updated the bit name and description for bits SSELCA to SSELCH in section 23.2.5, General PWM Timer Start Source Select Register (GTSSR)
			Updated the bit name and description for bits PSELCA to PSELCH in section 23.2.6, General PWM Timer Stop Source Select Register (GTPSR)
			Updated the bit name and description for bits CSELCA to CSELCH in section 23.2.7, General PWM Timer Clear Source Select Register (GTCSR)
			Updated the bit name and description for bits USELCA to USELCH in section 23.2.8, General PWM Timer Up Count Source Select Register (GTUPSR)
			Updated the bit name and description for bits DSELCA to DSELCH in section 23.2.9, General PWM Timer Down Count Source Select Register (GTDNSR)
			Updated the bit name and description for bits ASELCA to ASELCH in section 23.2.10, General PWM Timer Input Capture Source Select Register A(GTICASR)
			Updated the bit name and description for bits BSELCA to BSELCH in section 23.2.11, General PWM Timer Input Capture Source Select Register B(GTICBSR)
			Updated MD[2:0] in section 23.2.12, General PWM Timer Control Register (GTCR)
			Updated the description for section 23.2.23, General PWM Timer Dead Time Value Register U (GTDVU) to change GTDVm to GTDVU
			Updated Figure 23.5, Example of periodic count operation in down-counting by the count clock
			Updated section (5) Event count operation in down-counting using hardware sources in section 23.3.1.1, Counter Operation
			Updated section 23.3.2.2, Buffer Operation for GTCCRA and GTCCRB
			Updated Figure 23.28, Example for setting GTCCRA and GTCCRB buffer operation (for input capture)
			Updated section 23.3.3.2, Saw-Wave One-Shot Pulse Mode
			Updated Figure 23.6, Example for setting periodic count operation in down-counting by count clock
			Updated section 23.3.7.1, Hardware Start Operation for Figure 23.47 and Figure 23.48
			Updated section 23.3.7.2, Hardware Stop Operation including Figure 23.49 and Figure 23.50
			Updated section 23.3.7.3, Hardware Clear Operation including Figure 23.53 and Figure 23.54
			Updated Figure 23.55, Example for setting count clearing operation by a hardware source
			Updated Figure 23.59, Example of a simultaneous start, stop and clear by a hardware source with the same count cycle (GTPR register value)

Rev.	Date	Chapter	Summary
1.20	Aug 26, 2016	section 23, General PWM Timer (GPT)	Updated Figure 23.60, Example for setting simultaneous start by hardware source
			Updated Figure 23.66, Example of 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting
			Updated Table 23.9, Conditions of up-counting/down-counting in phase counting mode 2 (B)
			Updated Table 23.10, Conditions of up-counting/down-counting in phase counting mode 2 (C)
			Updated Table 23.16, Conditions of up-counting/down-counting in phase counting mode 5 (B)
			Updated Table 23.19, Output selection control method (positive phase)
			Updated section 23.9.2, Settings of GTCCRn during Compare Match Operation (n = A to F)
		section 24, Asynchronous General Purpose Timer (AGT)	Updated section 24.2.6, AGT Mode Register 2 (AGTMR2) for Note 1 and the LPM bit (Low Power Mode) description
		section 26, Watchdog Timer (WDT)	Updated UNDFF flag (Underflow Flag) description in section 26.2.3, WDT Status Register (WDTSR)
			Updated REFEF flag (Refresh Error Flag) description in section 26.2.3, WDT Status Register (WDTSR)
		section 27, Independent Watchdog Timer (IWDT)	Updated UNDFF flag (Underflow Flag) description in section 27.2.2, IWDT Status Register (IWDTSR)
			Updated REFEF flag (Refresh Error Flag) description in section 27.2.2, IWDT Status Register (IWDTSR)
		section 28, USB 2.0 Full-Speed Module (USBFS)	Updated Table 28.1, USBFS specifications to remove the sentence "HOCO clock that can be used as USB clock" from Others
			Updated section 28.2.13, Interrupt Status Register 0 (INTSTS0) for Note 5 following the register table
			Updated section 28.2.14, Interrupt Status Register 1 (INTSTS1)
			Updated BSTS bit (Buffer Status) description
			Updated section 28.3.3.16, Portable Device Detection Interrupt
			Updated section 28.3.15.1, Processing in Device Controller Mode
			Updated section 28.3.15.2, Processing when Host Controller is Selected
			Updated section Figure 28.25, Process flow for operating as charging downstream port (steps 1 to 4)
			Updated section Figure 28.26, Process flow for operating as charging downstream port (steps a to b)
		Figure 29., Serial Communications Interface (SCI)	Updated Table 29.2.13, Serial Status Register (SSR) for Non-Smart Card Interface and Non-FIFO Mode (SCMR.SMIF = 0 and FCR.FM = 0)
			Updated section 29.2.24, I²C Status Register (SISR) to change the R/W mode of bits [1], [3], and [7:6] from R/W to R
			Updated the title for Figure 29.23, Example flow of multi-processor serial transmission with non-FIFO selected
			Updated the title for Figure 29.27, Example flow of multi-processor serial reception with non-FIFO selected (1)
			Updated the title for Figure 29.28, Example flow of multi-processor serial reception with non-FIFO selected (2)
			Updated the title for Figure 29.37, Example flow of serial transmission in clock synchronous mode with non-FIFO selected
			Updated the title for Figure 29.41, Example flow of serial reception in clock synchronous mode with non-FIFO selected
			Updated Figure 29.42, Example flow of serial reception in clock synchronous mode with FIFO selected
			Updated the title for Figure 29.43, Example flow of simultaneous serial transmission and reception in clock synchronous mode with non-FIFO selected
			Updated Figure 29.66, Example flow of master transmission in simple IIC mode with transmission interrupts and reception interrupts
			Figure 29.68, Example flow of master reception in simple IIC mode with transmission interrupts and reception interrupts
		section 33, Serial Peripheral Interface (SPI)	Updated Table 33.3, Relationship between SPBR settings, BRDV[1:0] settings, and bit rates

Rev.	Date	Chapter	Summary
1.20	Aug 26, 2016	section 36, Serial Sound Interface (SSI)	Updated section 36.2.2, Status Register (SSISR)
			Updated section 36.2.4, FIFO Status Register (SSIFSR)
		section 37, SD/MMC Host Interface (SDHI)	Updated the value after reset for bits SDD3RMM and SDD3INM from 0 to 1 in section 37.2.12, SD INFO1 Interrupt Mask Register (SD_INFO1_MASK)
			Updated section 37.2.13, SD INFO2 Interrupt Mask Register (SD_INFO2_MASK) for bit descriptions for bit [10], and [14:12]
			Changed the value after reset for bit [5] from 1 to 0 in section 37.2.15, Transfer Data Length Register (SD_SIZE)
			Updated the descriptions for IOIRQ flag (SDIO Interrupt Status Flag), EXPUB52 flag (EXPUB52 Status Flag), and EXWT flag (EXWT Status Flag)
			Updated section 37.2.21, SDIO Interrupt Flag Register (SDIO_INFO1)
			Updated section 37.2.25, SD Interface Mode Setting Register (SDIF_MODE)
			Updated Figure 37.15, Example of IO_RW_DIRECT command (CMD52) operation
		Figure 39., 14-Bit A/D Converter (ADC14)	Updated Figure 39.1, 14-bit ADC block diagram
			Updated section 39.2.19, A/D Compare Function Window A Channel Select Register 1 (ADCMPANSR1)
			Updated section 39.2.28, A/D Compare Function Window B Channel Select Register (ADCMPBNSR)
			Updated section 39.8.8, ADHSC Bit Rewriting Procedure
		section 41, Temperature Sensor (TSN)	Updated section 41.2.1, Temperature Sensor Calibration Data Register H (TSCDRH)
			Updated section 41.2.2, Temperature Sensor Calibration Data Register L (TSCDRL)
		section 48., Flash Memory	Removed section 48.3.2.3, Flash Wait Cycle Register (FLWT)
			Removed section Additional Programming Disabled and added section 48.14.3, Constraints on Additional Writes
			Removed section 48.14.10, Flash Interface Clock (FCLK) during Programming and Erasure
		section 49., Segment LCD Controller/Driver (SLCDC)	Updated section Table 49.10, Relationship between LCD Display Data Register contents and segment/common outputs to add Note 2
		section 52, Electrical Characteristics	Updated Figure 52.17, Voltage dependency in High-speed operating mode (reference data)
			Updated Figure 52.18, Voltage dependency in Middle-speed mode (reference data)
			Updated Figure 52.19, Voltage dependency in Low-speed mode (reference data)
			Updated Figure 52.20, Voltage dependency in Low-voltage mode (reference data)
			Updated Figure 52.21, Voltage dependency in Subosc-speed mode (reference data)
			Updated Table 52.14, Operating and standby current (4) to add Note 6
			Split Figure 52.31 into two parts as Figure 52.30, Main clock oscillation start timing and Figure 52.31, PLL clock oscillation start timing (PLL is operated after main clock oscillation has settled)
			Updated Table 52.23, Reset timing for the Wait time after internal reset cancellation, and removed Reset period
			Removed Figure 51.34, Reset input timing (2)
			Updated Table 52.25, Timing of Recovery from Low power modes (2)
			Updated Table 52.26, Timing of recovery from Low power modes (3)
			Updated Table 52.27, Timing of recovery from Low power modes (4)
			Updated Table 52.29, Timing of recovery from Low power modes (6)
			Updated the conditions for Table 52.31, Bus timing (1)
			Updated the conditions for Table 52.32, Bus timing (2)
			Updated the conditions for Table 52.33, Bus timing (3)
			Updated the conditions for Table 52.34, Bus timing (4)
			Added Note 2 following Table 52.35, I/O Ports, POEG, GPT, AGT, KINT, and ADC14 trigger timing
			Updated Table 52.38, SCI timing (2) for the master data rise and fall time

Rev.	Date	Chapter	Summary
1.20	Aug 26, 2016	section 52, Electrical Characteristics	Updated Table 52.39, SCI timing (3) for Note 1 and to add a condition for Simple IIC (Fast mode)
			Updated Table 52.40, SPI timing to update Slave access time and Slave output release time information, and to add Note 4
			Updated Table 52.41, QSPI timing to add Note 4
			Updated Table 52.42, IIC timing to add a condition for IIC (Fast mode)
			Updated Table 52.56, A/D internal reference voltage characteristics to add Sampling time
			Updated Figure 52.81, Oscillation stop detection timing
			Updated Table 52.73, ACMPHS characteristics to add Internal reference voltage
			Updated Table 52.74, ACMPLP characteristics to add Internal reference voltage
		section 1, Port States in Each Processing Mode	Updated the Note following Table 1.1, Port states in each processing state
		section 3, I/O Registers	Added appendix I/O Registers
1.30	Feb 7, 2018	-	Updated section , Features for General Purpose I/O Ports, and Memory sections
		section 1, Overview	Updated Table 1.1, Arm core
			Updated Table 1.9, Analog to change ADC to ADC14 in Temperature Sensor row
			Updated Table 1.11, Data processing
			Updated Figure 1.1, Block diagram
			Updated Figure 1.2, Part numbering scheme
			Updated Table 1.14, Function comparison
			Updated section 1.5, Pin Functions
		section 2, CPU	Updated section 2.1.1, CPU
			Updated Table 2.1, Implementation options
			Updated section 2.5.1, Debug Mode Definition
			Updated section 2.6.3.2, CoreSight Component Registers, section 2.6.4.4, DBGREG CoreSight component registers, section 2.6.5.4, OCDREG CoreSight component registers to change CoreSight Registers to CoreSight Component Registers
			Updated section 2.6.5.3, MCU Control Register (MCUCTRL) for the note
			Updated section 2.9, SysTick System Timer to add note 1
			Updated section 2.11, OCD Emulator Connection to add section 2.11.1, DBGEN
			Updated section 2.11.3.4, Connecting Sequence and JTAG/SWD Authentication
		section 3, Operating Modes	Updated Figure 3.1, Mode-setting pin level and operating mode
		section 5, Memory Mirror Function (MMF)	Updated section 5.3.2, Setting Example to move the paragraph "The application code on the code flash..." to the top of the section, and removed the heading "Setting up MMSFR.MEMMIRADDR"
		section 6, Resets	Updated the legend following Table 6.2, Reset detect flags initialized by each reset source
			Updated Table 6.3, Module-related registers initialized by each reset source and the note 2 following the table
			Updated Table 6.4, States of SOSC when a reset occurs
			Updated Figure 6.1, Example of operations during power-on and voltage monitor 0 resets
		section 7, Option-Setting Memory	Updated section 7.2.1, Option Function Select Register 0 (OFS0)
			Updated section 7.2.2, Option Function Select Register 1 (OFS1)
			Updated section 7.2.4, Access Window Setting Control Register (AWSC)
			Updated section 7.2.5, Access Window Setting Register (AWS)
			Updated Table 7.2, Specifications for ID code protection
		section 8, Low Voltage Detection (LVD)	Updated section 8.2.7, Voltage Monitor 1 Circuit Control Register 0 (LVD1CR0)
			Updated section 8.2.8, Voltage Monitor 2 Circuit Control Register 0 (LVD2CR0)
			Updated Figure 8.4, Example operation of voltage monitor 0 reset
			Removed note 3 following Table 8.6, Procedures for setting bits related to voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitor operates

Rev.	Date	Chapter	Summary
1.30	Feb 7, 2018	section 9, Clock Generation Circuit	Updated Table 9.1, Clock Generation Circuit specifications for the clock sources
			Updated Table 9.2, Clock Generation Circuit Specifications (internal clock)
			Updated Figure 9.1, Clock generation circuit block diagram
			Updated section 9.2.3, PLL Clock Control Register 2 (PLLCCR2)
			Updated section 9.2.7, Main Clock Oscillator Control Register (MOSCCR)
			Updated section 9.2.12, Oscillation Stabilization Flag Register (OSCSF)
			Updated section 9.2.15, Main Clock Oscillator Wait Control Register (MOSCWTCR)
			Updated section 9.2.20, Clock Out Control Register (CKOCR)
			Updated section 9.5.2, Oscillation Stop Detection Interrupts
		section 10, Clock Frequency Accuracy Measurement Circuit (CAC)	Updated section 10.1, Overview
			Updated Figure 10.1, CAC block diagram
			Updated section 10.2.3, CAC Control Register 2 (CACR2)
		section 11, Low Power Modes	Updated Table 11.2, Operating conditions of each low power mode
			Updated section 11.2.4, Module Stop Control Register C (MSTPCRC) to add note 2
			Updated section 11.2.8, Snooze Control Register (SNZCR)
			Updated section 11.2.9, Snooze End Control Register (SNZEDCR)
			Updated section 11.2.10, Snooze Request Control Register (SNZREQCR)
			Updated section 11.7.1, Transition to Software Standby Mode to move a paragraph from section 11.8.1, Transition to Snooze Mode
			Added section 11.9.15, Module-Stop Function for an Unused Circuit
		section 12, VBATT Status Register (VBTSR)	Updated section 12.2.3, VBATT Status Register (VBTSR) for note 1 and 2
		section 14, Interrupt Controller Unit (ICU)	Updated section 14.2.1, IRQ Control Register i (IRQCRi) (i = 0 to 15)
			Updated section 14.2.3, Non-Maskable Interrupt Enable Register (NMIER)
			Updated section 14.2.4, Non-Maskable Interrupt Status Clear Register (NMICLR)
			Updated section 14.2.7, DMAC Event Link Setting Register n (DELSRn)
		section 15, Buses	Updated Table 15.1, Bus specifications
			Updated section 15.2.1, Main Buses
			Updated section 15.2.5, Bus Settings
			Updated section 15.3.3, CS Recovery Cycle Insertion Enable Register (CSRECEN)
			Updated section 15.3.8, Slave Bus Control Register (BUSSCNT<slave>)
			Updated section 15.3.9, Bus Error Address Register (BUSnERRADD) (n = 1 to 4)
			Updated Table 15.11, Conditions leading to illegal address access errors
		section 16, Memory Protection Unit (MPU)	Updated section 16.2, CPU Stack Pointer Monitor
			Updated section 16.2.3.1, Main Stack Pointer (MSP) Monitor Start Address Register (MSPMPUSA)
			Updated section 16.2.3.6, Stack Pointer Monitor Access Control Register (MSPMPUCTL, PSPMPUCTL)
			Updated section 16.2.3.7, Stack Pointer Monitor Protection Register (MSPMPUPT, PSPMPUPT)
			Updated Figure 16.3, MPU bus master diagram
			Updated Figure 16.4, MPU bus master group A
			Updated section 16.4.1.1, Group A Region n Start Address Register (MMPUSAn) (n = 0 to 15)
			Updated section 16.4.1.2, Group A Region n End Address Register (MMPUEAn) (n = 0 to 15)
			Updated section 16.4.1.3, Group A Region n Access Control Register (MMPUACAn) (n = 0 to 15)
			Updated Figure 16.9, Block diagram of the bus slave MPU
			Updated the note in section 16.5.1, Register Descriptions, corrected reset value of b13
			Updated section 16.5.2.1, Memory Protection
			Updated section 16.6.1.1, Security MPU Program Counter Start Address Register (SECMPUPCSn) (n = 0, 1)
			Updated section 16.6.1.2, Security MPU Program Counter End Address Register (SECMPUPCEn) (n = 0, 1)

Rev.	Date	Chapter	Summary
1.30	Feb 7, 2018	section 16, Memory Protection Unit (MPU)	Updated section 16.6.1.3, Security MPU Region 0 Start Address Register (SECMPUS0)
			Updated section 16.6.1.4, Security MPU Region 0 End Address Register (SECMPUE0)
		section 17, DMA Controller (DMAC)	Updated section 17.2.7, DMA Address Mode Register (DMAMD)
		section 18, Data Transfer Controller (DTC)	Updated Figure 18.1, DTC block diagram
			Updated Figure 18.2, DTC vector table and transfer information
			Updated section 18.6.3, Chain Transfer When Counter = 0
		section 19, Event Link Controller (ELC)	Updated Figure 19.1, ELC block diagram (n = 0 to 9, 12 to 18)
			Updated section 19.2.3, Event Link Setting Register n (ELSRn) (n = 0 to 9, 12 to 18)
			Added section 19.4.4, ELC Delay Time
		section 20, I/O Ports	Updated section 20.2.1, Port Control Register 1 (PCNTR1/PODR/PDR)
			Updated section 20.2.2, Port Control Register 2 (PCNTR2/EIDR/PIDR)
			Updated section 20.2.3, Port Control Register 3 (PCNTR3/PORR/POSR)
			Updated section 20.2.4, Port Control Register 4 (PCNTR4/EORR/EOSR)
			Updated section 20.2.5, Port mn Pin Function Select Register (PmnPFS/PmnPFS_HA/PmnPFS_BY) (m = 0 to 9; n = 00 to 15)
			Updated section 20.3.1, General I/O Ports
			Updated Table 20.3, Handling of unused pins
			Updated section 20.5.1, Procedure for Specifying the Pin Functions
			Updated section 20.5.4, Notes on Using of Analog Functions
		section 21, Key Interrupt Function (KINT)	Updated Table 21.1, Assignment of key interrupt detection pins and added a paragraph following the table
			Updated Figure 21.1, Key interrupt function block diagram
			Updated section 21.2.1, Key Return Control Register (KRCTL)
			Updated section 21.2.2, Key Return Flag Register (KRF)
			Updated section 21.2.3, Key Return Mode Register (KRM)
		section 22, Port Output Enable for GPT (POEG)	Updated the introduction to section 22, Port Output Enable for GPT (POEG)
			Updated Figure 22.1, POEG block diagram
			Updated section 22.2.1, POEG Group n Setting Register (POEGGn) (n = A to D)
			Updated section 22.3, Output-Disable Control Operation
			Updated Figure 22.2, Example of digital filter operation
			Updated section 22.3.2, Output-Disable Request from GPT
			Updated section 22.3.3, Comparator Interrupt Detection
			Updated section 22.3.6, Release from Output Disable
			Updated Figure 22.4, Output timing of external trigger to GPT
		section 23, General PWM Timer (GPT)	Updated section 23.1, Overview
			Updated Figure 23.1, GPT block diagram
			Updated section 23.2.5, General PWM Timer Start Source Select Register (GTSSR)
			Updated section 23.2.6, General PWM Timer Stop Source Select Register (GTPSR)
			Updated section 23.2.7, General PWM Timer Clear Source Select Register (GTCSR)
			Updated section 23.2.8, General PWM Timer Up Count Source Select Register (GTUPSR)
			Updated section 23.2.9, General PWM Timer Down Count Source Select Register (GTDNSR)
			Updated section 23.2.10, General PWM Timer Input Capture Source Select Register A(GTICASR)
			Updated section 23.2.11, General PWM Timer Input Capture Source Select Register B(GTICBSR)
			Updated section 23.2.12, General PWM Timer Control Register (GTCR)
			Updated section 23.2.13, General PWM Timer Count Direction and Duty Setting Register (GTUDDTYC)
			Updated section 23.2.14, General PWM Timer I/O Control Register (GTIOR)
			Updated section 23.2.15, General PWM Timer Interrupt Output Setting Register (GTINTAD)
			Updated section 23.2.24, Output Phase Switching Control Register (OPSCR)

Rev.	Date	Chapter	Summary
1.30	Feb 7, 2018	section 23, General PWM Timer (GPT)	Updated Figure 23.25, Example for setting GTCCRA and GTCCRB buffer operation for output compare
			Updated section 23.3.2.2, Buffer Operation for GTCCRA and GTCCRB
			Updated Figure 23.30, Example for setting saw-wave PWM mode
			Updated Figure 23.32, Example for setting saw-wave one-shot pulse mode
			Updated Figure 23.34, Example for setting triangle-wave PWM mode 1
			Updated Figure 23.36, Example for setting triangle-wave PWM mode 2
			Updated Figure 23.38, Example for setting triangle-wave PWM mode 3
			Updated Table 23.6, Output values after releasing 0%/100% duty setting (m = A, B)
			Updated section 23.3.8.2, Synchronized Operation by Hardware
			Updated Figure 23.77, GPT_OPS control flow conceptual diagram
			Updated Figure 23.78, 6-phase level output operation example
			Updated Figure 23.79, 6-phase PWM output operation example (chopper control)
			Updated Figure 23.80, Group output disable control operation example
			Updated Table 23.19, Output selection control method (positive phase)
			Updated Table 23.20, Output selection control method (negative phase)
			Removed section 23.3.11.4, Rotation Direction Control
			Updated section 23.3.11.6, Event Link Controller (ELC) Output
			Updated section 23.3.11.7, GPT_OPS Start Operation Setting Flow
			Updated section 23.4.1, Interrupt Sources
			Updated section 23.7.3, GTIOC Pin Output Negate Control
			Updated section 23.8.1, Pin Settings after Reset
			Updated section 23.8.2, Pin Initialization Due to Error during Operation
			Updated section 23.9.2, Settings of GTCCRn during Compare Match Operation (n = A to F)
		section 24, Asynchronous General Purpose Timer (AGT)	Updated Figure 24.1, AGT block diagram
			Updated note 1 following Table 24.2, AGT I/O pins
			Updated section 24.2.4, AGT Control Register (AGTCR)
			Updated section 24.3.1, Reload Register and Counter Rewrite Operation
			Updated Figure 24.7, Operation example 1 in event counter mode
			Updated Figure 24.8, Operation example 2 in event counter mode
			Updated section 24.4.1, Count Operation Start and Stop Control
			Updated section 24.4.7, When Selecting AGT0 Underflow as the Count Source
		section 25, Realtime Clock (RTC)	Updated section 25.2.22, Time Capture Control Register y (RTCCRy) (y = 0 to 2)
			Updated section 25.2.26, Date Capture Register y (RDAYCPy) (y = 0 to 2)/ BCNT3 Capture Register y (BCNT3CPy) (y = 0 to 2)
			Updated section 25.2.27, Month Capture Register y (RMONCPy) (y = 0 to 2)
			Updated Figure 25.7, Using alarm function
			Updated section 25.3.8.1, Automatic Adjustment
		section 26, Watchdog Timer (WDT)	Updated Figure 26.3, Operation example in register start mode
			Updated section 26.5.1, ICU Event Link Setting Register n (IELSRn) setting
		section 27, Independent Watchdog Timer (IWDT)	Updated Figure 27.1, IWDT block diagram
			Updated section 27.3.2, Refresh Operation
		section 28, USB 2.0 Full-Speed Module (USBFS)	Updated Table 28.1, USBFS specifications
			Updated Table 28.2, USBFS pin configuration
			Updated section 28.2.1, System Configuration Control Register (SYSCFG)
			Updated section 28.2.3, Device State Control Register 0 (DVSTCTR0)
			Updated section 28.2.4, CFIFO Port Register (CFIFO/CFIFOL) D0FIFO Port Register (D0FIFO/D0FIFOL) D1FIFO Port Register (D1FIFO/D1FIFOL)
			Updated section 28.2.5, CFIFO Port Select Register (CFIFOSEL) D0FIFO Port Select Register (D0FIFOSEL) D1FIFO Port Select Register (D1FIFOSEL)
			Updated section 28.2.12, SOF Output Configuration Register (SOFCFG)
			Updated section 28.2.13, Interrupt Status Register 0 (INTSTS0)
			Updated section 28.2.28, Pipe Maximum Packet Size Register (PIPEMAXP)
			Updated section 28.2.29, Pipe Cycle Control Register (PIPEPERI)
			Updated Table 28.11, Data cleared by USBFS when ACLRM = 1
			Updated section 28.2.33, Device Address n Configuration Register (DEVADDn) (n = 0 to 5)

Rev.	Date	Chapter	Summary
1.30	Feb 7, 2018	section 28, USB 2.0 Full-Speed Module (USBFS)	Updated section 28.2.35, BC Control Register 0 (USBBCCTRL0)
			Updated section 28.3.1.3, Controlling the USB Data Bus Using Resistors
			Updated Figure 28.5, Example OTG connection in self-powered state
			Updated Figure 28.6, Example device connection in self-powered state
			Updated Figure 28.7, Example host connection
			Updated Figure 28.8, Example device connection in bus-powered state
			Added Figure 28.9, Example device connection in bus-powered state 2
			Updated Figure 28.10, Example of functional connection with Battery Charging Rev 1.2 supported
			Updated Figure 28.11, USBFS interrupt-related circuits
			Updated Table 28.14, USBFS interrupts
			Updated section 28.3.3.1, BRDY Interrupt
			Updated Table 28.15, Condition for clearing BRDY bit
			Updated Figure 28.13, Timing of NRDY interrupt generation in device controller mode
			Updated Figure 28.16, Control transfer stage transitions
			Updated section 28.3.4.7, Data PID Sequence Bit
			Updated section 28.3.6, FIFO Buffer Clearing
			Updated section 28.3.7, FIFO Port Functions
			Updated section 28.3.11.1, Interval Counter for Interrupt Transfers in Host Controller Mode
			Updated section 28.3.12.3, Interval Counter
			Updated Figure 28.21, Example data setup operation
			Updated section 28.4.2, Clearing the Interrupt Status Register on Exiting Software Standby Mode
		section 29, Serial Communications Interface (SCI)	Updated section 29.2.4, Receive FIFO Data Register H, L, HL (FRDRH, FRDRL, FRDRHL)
			Updated section 29.2.10, Serial Mode Register for Smart Card Interface Mode (SMR_SMCI) (SCMR.SMIF = 1)
			Updated section 29.2.11, Serial Control Register (SCR) for Non-Smart Card Interface Mode (SCMR.SMIF = 0)
			Updated section 29.2.12, Serial Control Register for Smart Card Interface Mode (SCR_SMCI) (SCMR.SMIF = 1)
			Updated section 29.2.13, Serial Status Register (SSR) for Non-Smart Card Interface and Non-FIFO Mode (SCMR.SMIF = 0 and FCR.FM = 0)
			Updated section 29.2.14, Serial Status Register for Non-Smart Card Interface and FIFO Mode (SSR_FIFO) (SCMR.SMIF = 0 and FCR.FM = 1)
			Updated section 29.2.15, Serial Status Register for Smart Card Interface Mode (SSR_SMCI) (SCMR.SMIF = 1)
			Updated section 29.2.17, Bit Rate Register (BRR)
			Updated Table 29.15, BRR settings for different bit rates in smart card interface mode, n = 0, S = 372
			Updated Table 29.17, BRR settings for different bit rates in simple IIC mode
			Updated Table 29.18, Minimum widths at high and low level for SCL at different bit rates in simple IIC mode
			Updated section 29.2.18, Modulation Duty Register (MDDR)
			Updated section 29.2.19, Serial Extended Mode Register (SEMR)
			Updated section 29.2.23, I^2C Mode Register 3 (SIMR3)
			Updated section 29.2.26, FIFO Control Register (FCR)
			Updated section 29.2.30, Data Compare Match Control Register (DCCR)
			Updated section 29.2.31, Serial Port Register (SPTR)
			Updated section 29.3.5, CTS and RTS Functions
			Updated Figure 29.5, Example of address match (1) non-multi-processor mode
			Updated Figure 29.6, Example of address match (2) multi-processor mode
			Updated Figure 29.12, Example flow of serial transmission in asynchronous mode with non-FIFO selected
			Updated Figure 29.14, Example flow of serial transmission in asynchronous mode with FIFO selected
			Updated Figure 29.21, Example flow of serial reception in asynchronous mode with FIFO selected (2)
			Updated section 29.4, Multi-Processor Communication Function

Rev.	Date	Chapter	Summary
1.30	Feb 7, 2018	section 29, Serial Communications Interface (SCI)	Updated Figure 29.23, Example flow of multi-processor serial transmission with non-FIFO selected
			Updated Figure 29.25, Example flow of serial transmission in multi-processor mode with FIFO selected
			Updated Figure 29.30, Example flow of serial reception in multi-processor mode with FIFO selected
			Updated section 29.5.1, Clock
			Updated section 29.5.2, CTS and RTS Functions
			Updated section 29.5.3, SCI Initialization in Clock Synchronous Mode
			Updated section 29.5.4, Serial Data Transmission in Clock Synchronous Mode
			Updated Figure 29.38, Example flow of serial transmission in clock synchronous mode with FIFO selected
			Updated section 29.5.6, Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode
			Updated Figure 29.45, Example connection with a smart card (IC card)
			Updated section 29.6.3, Block Transfer Mode
			Updated section 29.6.4, Receive Data Sampling Timing and Reception Margin
			Updated Figure 29.50, Example flow of SCI initialization in smart card interface mode
			Added Figure 29.51, Example of Timing Chart of Data Transmission (Smart Card Interface Mode) and the preceding paragraph
			Added Figure 29.52, Data retransfer operation in SCI transmission mode
			Added Figure 29.53, SSR.TEND flag generation timing during transmission
			Updated section 29.6.8, Clock Output Control
			Updated section 29.8, Operation in Simple SPI Mode
			Updated Table 29.24, States of pins by mode and input level on SSn Pin
			Updated Table 29.25, SCI interrupt sources with non-FIFO selected
			Updated Table 29.26, SCI interrupt sources with FIFO selected
			Updated section 29.10.4, Interrupts in Smart Card Interface Mode
			Updated section 29.12, Address non-match event output (SCI0_DCUF)
			Updated section 29.13, Noise Cancellation Function
			Added Figure 29.72, Digital noise filter circuit block diagram
			Updated section 29.14.2, SCI Operations during Low Power State
			Added Figure 29.74, Port pin states during transition to Software Standby mode with internal clock and asynchronous transmission
			Added Figure 29.75, Port pin states during transition to Software Standby mode with internal clock and clock synchronous transmission
			Added section 29.14.8, Notes on Starting Transfer
		section 31, I^2C Bus Interface (IIC)	Updated section 31.2.1, I^2C Bus Control Register 1 (ICCR1)
			Updated Table 31.3, IIC resets
			Updated section 31.2.2, I^2C Bus Control Register 2 (ICCR2)
			Updated section 31.2.16, I^2C Bus Bit Rate High-Level Register (ICBRH)
			Updated Table 31.6, Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 0 title
			Updated Table 31.7, Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 1 and NFE = 0 title
			Updated Table 31.8, Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 1 and NFE = 1 title
			Updated section 31.3.1, Communication Data Format
			Updated section 31.3.3, Master Transmit Operation
			Updated Figure 31.26, AASy flag set and clear timing with 7-bit and 10-bit address formats mixed
			Updated Figure 31.28, AASy/DID flag set/clear timing during reception of device ID
			Updated Figure 31.31, Example operation of normal wakeup modes 1 and 2 when wakeup is triggered by an interrupt other than IIC wakeup interrupt, for example, IRQn
			Updated Figure 31.36, Example operation of command recovery mode and EEP response mode when wakeup is triggered by an interrupt other than IIC wakeup interrupt, for example, the IRQn
			Updated Figure 31.37, Timing of command recovery mode and EEP response mode

Rev.	Date	Chapter	Summary
1.30	Feb 7, 2018	section 31, I2C Bus Interface (IIC)	Updated section 31.9.1, Function to Prevent Wrong Transmission of Transmit Data
			Updated Figure 31.45, Start or restart condition issue timing (ST and RS bits)
			Updated Figure 31.48, Timeout function (TMOE, TMOS, TMOH, and TMOL bits)
			Updated Figure 31.50, SMBus timeout measurement
			Updated Table 31.10, Interrupt sources
			Updated section 31.15, Register States when Issuing each Condition
		section 32, Controller Area Network (CAN) Module	Updated Table 32.1, CAN module specifications
			Updated section 32.2.4, FIFO Received ID Compare Registers 0 and 1 (FIDCR0 and FIDCR1)
			Updated section 32.2.6, Mailbox Register j (MBj_ID, MBj_DL, MBj_Dm, MBj_TS) (j = 0 to 31, m = 0 to 7)
			Updated section 32.2.13, Transmit FIFO Control Register (TFCR)
			Updated section 32.2.21, Error Interrupt Factor Judge Register (EIFR)
			Updated Figure 32.9, Transition between different modes of operation
			Updated section 32.6, Acceptance Filtering and Masking Functions
			Updated section 32.7.2, Transmission
		section 33, Serial Peripheral Interface (SPI)	Updated section 33.2.1, SPI Control Register (SPCR)
			Updated section 33.2.4, SPI Status Register (SPSR)
			Updated Figure 33.2, Configuration of SPDR/SPDR_HA
			Updated Table 33.5, Relationship between SPI modes and SPCR settings and description of each mode
			Updated section 33.3.3.3, Single Master and Multi-Slave with the MCU configured as a Master
			Updated section 33.3.4.1, When Parity is Disabled (SPCR2.SPPE = 0), step (4)
			Updated section 33.3.5.1, CPHA = 0
			Updated Figure 33.29, Clock stop waveform when serial transfer continues while receive buffer is full in master mode (CPHA = 1)
			Updated Figure 33.30, Clock stop waveform when serial transfer continues while receive buffer is full in master mode (CPHA = 0)
			Updated section 33.3.9.1, Initialization by Clearing the SPE Bit
			Updated section 33.3.10.1, Master Mode Operation, step (1)
			Updated Figure 33.37, Transmission flow in master mode transmission
			Updated Figure 33.38, Reception flow in master mode
			Updated Figure 33.39, Flowchart for error processing in master mode
			Updated section 33.3.11, Clock Synchronous Operation
			Updated Figure 33.47, Example of initialization flow in master mode for clock synchronous operation
			Updated section 33.3.11.2, Slave Mode Operation
			Updated section 33.3.12, Loopback Mode
			Updated Table 33.13, SPI interrupt sources
			Updated Table 33.15, Conditions for generation of transmission-complete event in slave mode
		section 34, Quad Serial Peripheral Interface (QSPI)	Updated Table 34.1, QSPI specifications
			Updated Figure 34.2, Default area setting and AHB space memory map
			Updated Figure 34.12, Hold time adjustment of output enable with SFMOEX bit
			Updated section 34.8.2, Releasing XIP Mode to remove Table 34.9, Release codes for automatic issuing of release from XIP mode
		section 35, Cyclic Redundancy Check (CRC) Calculator	Updated the title of Table 35.1, CRC calculator specifications
			Updated Figure 35.1, CRC calculator block diagram
			Updated Figure 35.4, LSB-first data reception
		section 36, Serial Sound Interface (SSI)	Updated Figure 36.1, SSI0 block diagram
			Updated Figure 36.2, SSI1 block diagram
			Updated section 36.2.2, Status Register (SSISR)
			Updated Figure 36.6, MSB-first and right-aligned format, transmitted and received in the order of padding bits and serial data
			Updated section 36.3.4, Operating States
			Updated section 36.5.1, Setting for the Module-Stop State
		section 37, SD/MMC Host Interface (SDHI)	Updated section 37.2.1, Command Type Register (SD_CMD) and removed table 37.3, Examples of SD_CMD register settings

Rev.	Date	Chapter	Summary
1.30	Feb 7, 2018	section 37, SD/MMC Host Interface (SDHI)	Updated section 37.2.10, SD Card Interrupt Flag Register 1 (SD_INFO1)
			Updated section 37.2.11, SD Card Interrupt Flag Register 2 (SD_INFO2)
			Updated section 37.2.14, SD Clock Control Register (SD_CLK_CTRL)
			Updated section 37.2.17, SD Error Status Register 1 (SD_ERR_STS1)
			Updated Figure 37.6, Reading from the SD_BUF0 register
			Updated Table 37.5, Interrupt sources
		section 39, 14-Bit A/D Converter (ADC14)	Updated Table 39.1, Specifications of 14-bit ADC
			Updated Table 39.2, 14-bit ADC functions
			Updated section 39.2.1, A/D Data Registers y (ADDRy), A/D Data Duplexing Register (ADDBLDR), A/D Data Duplexing Register A (ADDBLDRA), A/D Data Duplexing Register B (ADDBLDRB), A/D Temperature Sensor Data Register (ADTSDR), A/D Internal Reference Voltage Data Register (ADOCDR)
			Updated section 39.2.3, A/D Control Register (ADCSR)
			Updated section 39.2.9, A/D-Converted Value Addition/Average Channel Select Register 1 (ADADS1)
			Updated section 39.2.10, A/D-Converted Value Addition/Average Count Select Register (ADADC)
			Updated section 39.2.13, A/D Conversion Extended Input Control Register (ADEXICR)
			Updated section 39.2.14, A/D Sampling State Register n (ADSSTRn) (n = 00 to 15, L, T, O)
			Updated section 39.2.15, A/D Disconnection Detection Control Register (ADDISCR)
			Updated section 39.2.20, A/D Compare Function Window A Extended Input Select Register (ADCMPANSER)
			Updated section 39.2.24, A/D Compare Function Window A Lower-Side Level Setting Register (ADCMPDR0), A/D Compare Function Window A Upper-Side Level Setting Register (ADCMPDR1), A/D Compare Function Window B Lower-Side Level Setting Register (ADWINLLB), A/D Compare Function Window B Upper-Side Level Setting Register (ADWINULB)
			Updated section 39.2.28, A/D Compare Function Window B Channel Select Register (ADCMPBNSR)
			Updated section 39.3.1, Scanning Operation
			Updated section 39.3.2.2, Channel Selection and Self-Diagnosis
			Updated section 39.3.2.3, A/D Conversion of Temperature Sensor Output/Internal Reference Voltage
			Updated section 39.3.2.4, A/D Conversion in Double Trigger Mode
			Updated section 39.3.2.5, Extended Operations when Double Trigger Mode is Selected
			Updated Figure 39.10, Example of extended operation in double-trigger mode (1) with duplication selected for AN003, ELC_AD00/ELC_AD01 selected
			Updated section 39.3.4.3, Operation under Group A Priority Control
			Updated Figure 39.15, Flow of ADGSPCR.PGS bit setting
			Updated Figure 39.18, Example of operations under group A priority control (3) when ADGSPCR.GBRSCN = 1 and ADGSPCR.GBRP = 0
			Updated section 39.3.5.1, Compare Function Window A/B
			Updated Figure 39.22, Setting example when using event output of the compare function
			Updated section 39.3.5.2, Event Output of Compare Function
			Updated Figure 39.23, Event output operation example of compare function when AN000 to AN001 are compared
			Updated Table 39.10, Times for conversion during scanning (in numbers of cycles of ADCLK and PCLKB)
			Updated Figure 39.24, Scan conversion timing activated by software or synchronous trigger ELC input
			Updated Figure 39.25, Scan conversion timing activated by asynchronous trigger input (ADTRG0)
			Updated section 39.3.7, Usage Example of A/D Data Register Automatic Clearing Function
			Updated section 39.3.9, Disconnection Detection Assist Function
			Updated section 39.4.1, Interrupt Requests

Rev.	Date	Chapter	Summary
1.30	Feb 7, 2018	section 39, 14-Bit A/D Converter (ADC14)	Updated section 39.7, A/D Conversion Procedure when Selecting Internal Reference Voltage as High-Potential Reference Voltage
			Updated Figure 39.31, Procedures for clearing the ADCSR.ADST bit by software
			Updated section 39.8.7, Error in Absolute Accuracy when Disconnection Detection Assistance is in Use
			Updated section 39.8.9, Notes on Operating Modes and Status Bits
			Updated section 39.8.12, Port Setting when Using the 14-bit A/D Converter Input
			Updated Table 39.12, List of OPAMP, ACMPHS, and ACMPLP pin that should not be selected during A/D conversion
		section 40, 12-Bit D/A Converter (DAC12)	Updated Figure 40.4, Example when the DAC12 cannot capture the synchronous D/A conversion enable input signal from the ADC14
		section 41, Temperature Sensor (TSN)	Updated section 41.2.1, Temperature Sensor Calibration Data Register H (TSCDRH)
			Updated section 41.2.2, Temperature Sensor Calibration Data Register L (TSCDRL)
		section 42, Operational Amplifier (OPAMP)	Updated section 42.1, Overview
			Updated Figure 42.2, Operational amplifier state transitions
			Updated Figure 42.3, Operational amplifier control operation in software trigger mode is used for control when the reference current circuit and operational amplifier are activated/stopped by software trigger mode
			Updated Figure 42.4, Operational amplifier control operation when activation trigger mode is used for activation, when the reference current circuit and operational amplifier are activated by an activation trigger and stopped by setting the AMPC register
			Updated Figure 42.5, Operational amplifier control operation in activation and A/D trigger mode (1) when the reference current circuit and operational amplifier are activated by an activation trigger and stopped by an A/D conversion end (trigger)
			Updated Figure 42.6, Operational amplifier control operation in activation and A/D trigger mode (2) when the reference current circuit and operational amplifier are stopped by setting the AMPC register to be activated by an activation trigger and stopped by an A/D conversion end (trigger)
			Updated Figure 42.7, Procedure to start and stop OPAMP in software trigger mode
		section 43, High-Speed Analog Comparator (ACMPHS)	Updated section 43.2.1, Comparator Control Register (CMPCTL)
			Updated section 43.2.5, Comparator Output Control Register (CPIOC)
		section 44, Low Power Analog Comparator (ACMPLP)	Moved note 1 to follow Table 44.2, Comparator pin configuration
			Updated Table 44.3, Procedure for setting the ACMPLP associated registers (i = 0, 1)
			Updated section 44.6, ELC Event Output
			Updated section 44.8, Comparator Pin Output
		section 45, Capacitive Touch Sensing Unit (CTSU)	Updated Figure 45.3, CTSU block diagram
			Updated section 45.2.5, CTSU Measurement Channel Register 0 (CTSUMCH0)
			Updated section 45.2.8, CTSU Channel Enable Control Register 1 (CTSUCHAC1)
			Updated section 45.2.13, CTSU Channel Transmit/Receive Control Register 1 (CTSUCHTRC1)
			Updated section 45.2.14, CTSU Channel Transmit/Receive Control Register 2 (CTSUCHTRC2)
			Updated section 45.2.15, CTSU Channel Transmit/Receive Control Register 3 (CTSUCHTRC3)
			Updated section 45.2.16, CTSU Channel Transmit/Receive Control Register 4 (CTSUCHTRC4)
			Updated Figure 45.6, Discharging operation
			Updated Figure 45.9, CTSU initial setting flow
			Updated section 45.3.2.2, Status Counter
			Updated Figure 45.14, Software flow and example operation of self-capacitance multi-scan mode
			Updated Figure 45.16, Software flow and operation example of mutual capacitance full-scan mode
			Updated the sentence "The CTSU mutual capacitance status flag....." preceding Table 45.8, Touch pin states in mutual capacitance full-scan mode

Rev.	Date	Chapter	Summary
1.30	Feb 7, 2018	section 47, SRAM	Added section section 47.4.2, Store Buffer of SRAM
		section 48, Flash Memory	Updated Table 48.1, Specifications of code flash memory and data flash memory
			Updated section 48.2, Memory Structure
			Updated Table 48.2, Read and P/E addresses of the code flash memory
			Updated Table 48.3, Read and P/E addresses of the data flash memory
			Updated Figure 48.4, FCACHE block diagram
			Removed section 48.3.2.3. Flash Wait Cycle Register (FLWT)
			Updated Table 48.6, Specifications for ID code protection
			Updated Figure 48.10, System configuration in USB boot mode
			Updated Figure 48.11, Environment for writing programs to the flash memory
			Updated Figure 48.12, Schematic view of self-programming
			Updated section 48.14.5, Non-maskable Interrupt Disabled during Programming and Erasure
		section 52, Electrical Characteristics	Updated Table 52.2, Recommended operating conditions and the notes following the table
			Updated Table 52.3, DC Characteristics and added note 1 following the table
			Updated Table 52.4, I/O VIH, VIL (1)
			Updated Table 52.5, I/O VIH, VIL (2)
			Updated Table 52.7, I/O V_{OH}, V_{OL} (1) and added note 5 following the table
			Updated Table 52.8, I/O V_{OH}, V_{OL} (2) and added note 5 following the table
			Updated Table 52.9, I/O V_{OH}, V_{OL} (3) and added note 2 following the table
			Updated Figure 52.13, V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 2.7 V when middle drive output is selected (reference data)
			Updated Figure 52.15, V_{OH}/V_{OL} and I_{OH}/I_{OL} temperature characteristics at VCC = 5.5 V when middle drive output is selected (reference data)
			Updated Table 52.11, Operating and standby current (1) and updated note 6 following the table
			Updated Figure 52.19, Voltage dependency in Low-speed mode (reference data)
			Updated Table 52.13, Operating and standby current (3)
			Updated Table 52.14, Operating and standby current (4)
			Updated Table 52.17, Operation frequency value in High-speed operating mode to add note 5 following the table
			Updated Table 52.18, Operation frequency value in Middle-speed mode to add note 5 following the table
			Updated Table 52.19, Operation frequency value in Low-speed mode to add note 4 following the table
			Updated Table 52.20, Operation frequency value in Low-voltage mode to add note 5 following the table
			Updated Table 52.22, Clock timing
			Updated Table 52.23, Reset timing to remove note 3
			Updated Table 52.29, Timing of recovery from Low power modes (6)
			Added Figure 52.36, Recovery timing from Software Standby mode to Snooze mode
			Updated Figure 52.53, SCI simple SPI mode timing (master, CKPH = 1)
			Updated Table 52.39, SCI timing (3) and note 1 following the table
			Updated Table 52.40, SPI timing
			Updated Figure 52.58, SPI clock timing
			Updated Figure 52.59, SPI timing (master, CPHA = 0) (bit rate: PCLKA division ratio is set to any value other than 1/2)
			Updated Figure 52.60, SPI timing (master, CPHA = 0) (bit rate: PCLKA division ratio is set to 1/2)
			Updated Figure 52.61, SPI timing (master, CPHA = 1) (bit rate: PCLKA division ratio is set to any value other than 1/2)
			Updated Figure 52.62, SPI timing (master, CPHA = 1) (bit rate: PCLKA division ratio is set to 1/2)
			Updated Figure 52.63, SPI timing (slave, CPHA = 0)
			Updated Figure 52.64, SPI timing (slave, CPHA = 1)
			Updated Figure 52.72, SD/MMC host interface signal timing
			Updated Figure 52.77, AVCC0 to VREFH0 voltage range
			Updated Table 52.48, A/D conversion characteristics (1) in High-speed A/D conversion mode and added note 2 following the table

Rev.	Date	Chapter	Summary
1.30	Feb 7, 2018	section 52, Electrical Characteristics	Updated Table 52.49, A/D conversion characteristics (2) in High-speed A/D conversion mode and added note 2 following the table
			Updated Table 52.50, A/D conversion characteristics (3) in High-speed A/D conversion mode and added note 2 following the table
			Updated Table 52.51, A/D conversion characteristics (4) in Low-power A/D conversion mode and added note 2 following the table
			Updated Table 52.52, A/D conversion characteristics (5) in Low-power A/D conversion mode and added note 2 following the table
			Updated Table 52.53, A/D conversion characteristics (6) in Low-power A/D conversion mode and added note 2 following the table
			Updated Table 52.54, A/D conversion characteristics (7) in Low-power A/D conversion mode and added note 2 following the table
			Added Figure 52.78, Equivalent circuit for analog input
			Updated Figure 52.79, Illustration of 14-bit A/D converter characteristic terms
			Updated note 1 following the Table 52.62, Power-on reset circuit and voltage detection circuit characteristics (1)
			Updated Table 52.63, Power-on reset circuit and voltage detection circuit characteristics (2)
			Updated Figure 52.83, Power-on reset timing
			Added Table 52.65, VBATT-I/O characteristics
			Updated Table 52.74, ACMPLP characteristics
			Updated Table 52.75, OPAMP characteristics
			Updated Table 52.77, Code flash characteristics (2)
			Updated Table 52.81, Data flash characteristics (3)
			Updated Table 52.82, Boundary scan
			Updated Table 52.83, JTAG (Debug) characteristics (1)
			Updated Table 52.84, JTAG (Debug) characteristics (2)
		section 1, Port States in Each Processing Mode	Updated Table 1.1, Port states in each processing state, and added note 2 following the table
		section 3, I/O Registers	Updated Table 3.1, Peripheral base address
			Updated Table 3.2, Access cycles for non-GPT modules
			Updated Table 3.4, Register description
1.40	Oct 29, 2018	section 1, Preface	Updated the Preface to use the latest template
		section 1, Overview	Updated Table 1.3, System for the Clock Frequency Accuracy Measurement Circuit
			Updated Table 1.14, Function comparison for the HMI row
		section 2, CPU	Updated section 2.1.2, Debug to add a sentence to the Flash Patch and Breakpoint unit (FPB) item
		section 6, Module-related registers initialized by each reset source	Updated the note 2 following Table 6.3, Module-related registers initialized by each reset source
		section 9, Clock Generation Circuit Specifications (internal clock)	Updated Table 9.2, Clock Generation Circuit Specifications (internal clock) for the AGT clock specification value
			Updated section 9.2.6, Memory Wait Cycle Control Register (MEMWAIT)
		section 14, Return from Snooze mode	Updated section 14.6.3, Return from Snooze mode
		section 20, I/O Ports registers connection diagram	Updated Figure 20.1, I/O Ports registers connection diagram
			Updated section 20.2.1, Port Control Register 1 (PCNTR1/PODR/PDR)
			Updated section 20.2.2, Port Control Register 2 (PCNTR2/EIDR/PIDR)
			Updated section 20.2.3, Port Control Register 3 (PCNTR3/PORR/POSR)
			Updated section 20.2.4, Port Control Register 4 (PCNTR4/EORR/EOSR)
			Updated Figure 20.2, Event ports input data
			Updated Figure 20.4, Generation of event pulse
			Updated Table 20.3, Handling of unused pins
		section 22, POEG specifications	Updated Table 22.1, POEG specifications, section 22.3, Output-Disable Control Operation, section 22.5, External Trigger Output to GPT to update "GTETRGA to GTETRGD" to "GTETRGn"

Rev.	Date	Chapter	Summary
1.40	Oct 29, 2018	section 24, AGT I/O Control Register (AGTIOC)	Updated section 24.2.7, AGT I/O Control Register (AGTIOC), for the note 1 following the table
			Updated section 24.2.9, AGT Compare Match Function Select Register (AGTCMSR)
			Updated section 24.3.1, Reload Register and Counter Rewrite Operation
			Updated Figure 24.8, Operation example 2 in event counter mode
		section 26, Watchdog Timer (WDT)	Updated Figure 26.4, Operation example in auto-start mode
		section 28, USB 2.0 Full-Speed Module (USBFS)	Updated section 28.2.5, CFIFO Port Select Register (CFIFOSEL) D0FIFO Port Select Register (D0FIFOSEL) D1FIFO Port Select Register (D1FIFOSEL)
			Updated section 28.2.30, PIPEn Control Registers (PIPEnCTR) (n = 1 to 9)
		section 29, Smart Card Mode Register (SCMR)	Updated section 29.2.16, Smart Card Mode Register (SCMR)
			Updated section 29.2.24, I2C Status Register (SISR)s
		section 31, I2C Bus Interface (IIC)	Updated Table 31.6, Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 0 through Table 31.8, Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 1 and NFE = 1
		section 32, Controller Area Network (CAN) Module	Updated Table 32.1, CAN module specifications for the CAN clock source
			Updated section 32.2.2, Bit Configuration Register (BCR)
			Updated section 32.7.1, Reception
		section 33, Serial Peripheral Interface (SPI)	Updated section 33.2.4, SPI Status Register (SPSR)
		section 37, SD/MMC Host Interface (SDHI)	Updated section 37.2.11, SD Card Interrupt Flag Register 2 (SD_INFO2)
			Updated section 37.2.17, SD Error Status Register 1 (SD_ERR_STS1)
			Updated section 37.2.21, SDIO Interrupt Flag Register (SDIO_INFO1)
			Updated section 37.2.22, SDIO INFO1 Interrupt Mask Register (SDIO_INFO1_MASK)
		section 39, 14-Bit A/D Converter (ADC14)	Updated section 39.2.20, A/D Compare Function Window A Extended Input Select Register (ADCMPANSER)
		section 43, High-Speed Analog Comparator (ACMPHS)	Updated section 43.2.4, Comparator Output Monitor Register (CMPMON)
			Updated section 43.2.5, Comparator Output Control Register (CPIOC)
		section 45, CTSU Control Register 0 (CTSUCR0)	Updated section 45.2.1, CTSU Control Register 0 (CTSUCR0)
		section 48, Flash Memory	Updated Table 48.4, Flash cache overview
		section 50, Secure Cryptographic Engine (SCE5)	Added section 50, Secure Cryptographic Engine (SCE5)
		section 52, Electrical Characteristics	Updated section 52, Electrical Characteristics
		section 2, Package Dimensions	Replaced Figure 2.1, LGA 145-pin to use the latest graphic
		section 20, I/O Ports	Updated Figure 20.1, I/O Ports registers connection diagram
			Updated Figure 20.2, Event ports input data
			Updated Figure 20.4, Generation of event pulse

S3A7 Microcontroller Group User's Manual

Publication Date: Rev.1.40 Oct 29, 2018

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics Corporation
TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

© 2018 Renesas Electronics Corporation. All rights reserved.

Colophon 5.2

Renesas Synergy™ Platform
S3A7 Microcontroller Group

Renesas Electronics Corporation

R01UM0002EU0140